

**Penerapan Algoritma Genetik Untuk Vehicle Routing Problem
with Time Windows (VRPTW) Pada Kasus Optimasi Distribusi
Beras Bersubsidi**

SKRIPSI

**Untuk Memenuhi Sebagian Persyaratan Mencapai Gelar Sarjana Komputer
Pada Program Teknologi Informasi Dan Ilmu Komputer**



Oleh :

Farah Bahtera Putri

105090600111015

**PROGRAM STUDI ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2014**

LEMBAR PERSETUJUAN

**Penerapan Algoritma Genetik Untuk Vehicle Routing Problem
with Time Windows (VRPTW) Pada Kasus Optimasi Distribusi
Beras Bersubsidi**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

FARAH BAHTERA PUTRI

105090600111015

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Wayan Firdaus Mahmudy, S.Si., MT., Ph.D

Dian Eka Ratnawati, S.Si., M.Kom

NIP. 197209191997021001

NIP. 197306192002122001

LEMBAR PENGESAHAN

Penerapan Algoritma Genetik Untuk Vehicle Routing Problem with Time Windows (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

FARAH BAHTERA PUTRI

NIM. 105090600111015

Skripsi ini telah diuji dan dinyatakan lulus
tanggal 18 Desember 2014

Dosen Penguji I

Dosen Penguji II

Edy Santoso, S.Si, M.Kom.

NIP. 19740414 200312 1 004

Lailil Muflikhah, S.Kom, M.Sc.

NIP. 19741113 200501 2 001

Dosen Penguji III

Rekyan Regasari Mardi Putri, ST., MT.

NIP. 77041406120253

Mengetahui,
Ketua Program Studi Teknik Informatika / Ilmu Komputer

Drs. Marji, MT

NIP. 19670801 199203 1001

PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : Farah Bahtera Putri
NIM : 105090600111015
Jurusan/Program studi : Ilmu Komputer / Informatika
Fakultas/Program : Program Teknologi Informasi dan Ilmu
Komputer / Strata - 1

Menyatakan dengan sungguh-sungguh bahwa skripsi yang saya tulis ini benar-benar tulisan saya, dan bukan merupakan plagiat baik sebagian atau seluruhnya. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil plagiat, baik sebagian atau seluruhnya, maka saya bersedia menerima sanksi atas perbuatan tersebut sesuai dengan ketentuan yang berlaku.



Malang, 18 Desember 2014

Mahasiswa,

Farah Bahtera Putri
NIM 105090600111015

KATA PENGANTAR

Syukur dan alhamdulillah penulis panjatkan kehadiran Allah SWT atas segala Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul: “Penerapan Algoritma Genetik Untuk Vehicle Routing Problem with Time Windows (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi”.

Tujuan dari penulisan skripsi ini adalah syarat ujian skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTIIK), Program Studi Informatika/Illmu Komputer, Universitas Brawijaya Malang. Sehubungan dengan selesainya tugas akhir ini, penulis mengucapkan terima kasih kepada:

1. Wayan Firdaus Mahmudy, S.Si., MT., Ph.D selaku Dosen Pembimbing Skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Dian Eka Ratnawati, S.Si., M.Kom selaku Dosen Pembimbing kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, MT. selaku Ketua Program Studi Informatika/Illmu Komputer Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Bapak Budiono dan Ibu Sulistyawati selaku orangtua yang selalu memberikan semangat dan doa hingga terselesaikannya skripsi ini.
6. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
7. Teman-teman dan sahabat-sahabat Teknik Informatika angkatan 2010 khususnya Dwi Aries S, Meitasari Winardi, Shinta Ayu V, Dita Oktaria,

Fayruz Al-Baity, Monica Intan, Anjar Dwi O yang bersedia memberi bantuan dan semangat ketika penulis merasa kesulitan.

8. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 18 Desember 2014

Penulis



ABSTRAK

Farah Bahtera Putri. 2014 : Penerapan Algoritma Genetik Untuk Vehicle Routing Problem with Time Windows (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi.

Dosen Pembimbing : Wayan Fidaus Mahmudy, S.Si., MT., Ph.D dan Dian Eka Ratnawati, S.Si., M.Kom

Penyaluran Beras Bersubsidi Bagi Kelompok Masyarakat Berpendapatan Rendah (Raskin) merupakan program nasional dari Pemerintah Pusat dan Daerah untuk meningkatkan kesejahteraan dalam memenuhi kebutuhan pangan masyarakat berpendapatan rendah. Penyaluran Beras Bersubsidi telah dilakukan Perum Bulog kesetiap pelanggan dengan kebutuhan khusus. Kondisi ini menyebabkan distributor memiliki kesulitan saat melakukan pengiriman beras ke setiap tujuan dengan permintaan, waktu jarak tempuh, dan waktu pelayanan yang berbeda-beda. Solusi yang perlu diperhatikan dalam permasalahan ini adalah menghitung rute optimal dengan batas waktu pelayanan yang sudah ditentukan. *Vehicle Routing Problem with Time Windows* (VRPTW) dalam Algoritma Genetika mampu menghitung optimasi rute dengan kapasitas kendaraan yang terbatas. Pada penelitian ini terdapat 20 kromosom sebagai pelanggan dengan jarak, jumlah permintaan dan waktu pelayanan (*time frame*). Jumlah permintaan setiap kromosom dibagi sesuai kapasitas truk. Dari data tersebut akan dibentuk sebuah populasi dengan jumlah yang bervariasi. Ukuran populasi yang optimal dari hasil uji coba adalah 80 populasi. Dengan generasi optimal sebanyak 2500 generasi. Nilai *crossover rate* dan *mutation rate* didapat dari nilai *fitness* terbaik yaitu 0.021716518 dengan *crossover rate* 0.4 dan *mutation rate* 0.6. Hasil akhir adalah kromosom terbaik yang merupakan keberhasilan distribusi beras bersubsidi dengan waktu optimal dan jumlah pinalti terendah.

Kata Kunci : Algoritma genetika, optimasi rute, *Vehicle Routing Problem with Time Windows*

ABSTRACT

Farah Bahtera Putri, 2014 : *Application of Genetic Algorithm for Vehicle Routing Problem with Time Windows (VRPTW) In The Case of Subsidized Rice Distribution Optimization.*

Advisor : Wayan Fidaus Mahmudy, S.Si., MT., Ph.D and Dian Eka Ratnawati, S.Si., M.Kom

The distribution of Subsidized Rice is performed by *Perum Bulog* for Group of Low Income Communities (*Raskin*). This assistance is a kind of national program from Central Government and local government to improve welfare to fulfill of food needed for low-income communities. This condition causes the distributor having a problem during delivery of rice to every destination with demand, mileage time and differences of service time. The solutions need to be considered in this problem is to calculate the optimal route to the deadline specified service. Vehicle Routing Problem with Time Windows (VRPTW) in the Genetic Algorithm (GA) capable of calculating vehicle route optimization with a limited of capacity. The GA principle is based on the concepts of biology evolution that is reproduction, mutation, crossover, and selection. In this study, there are 20 chromosomes as a customer with the distance, demand and the service time (time frame). The number of requests each chromosome is divided according to the capacity of the truck. From these data would set up a population with varying amounts. The optimal of population size from the trial results is 80 populations and the optimal generation as much as 2500 generations. Value of the crossover rate and mutation rates obtained from the best fitness value is 0.021716518 with crossover rate 0.4 and mutation rate of 0.6. The final result is the best chromosome which is the success of the subsidized rice distribution with optimal time and the lowest number of penalties.

Keywords : *Genetic algorithm, route optimization, Vehicle Routing Problem with Time Windows*

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN.....	iii
KATA PENGANTAR	iii
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR SOURCE CODE	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	17
2.1 Pelaksanaan Distribusi Raskin	17
2.1.1 Sistem Distribusi Beras.....	17
2.1.2 Pemilihan Rute	18
2.2 Vehicle Routing Problem (VRP).....	18
2.2.1 Vehicle Routing Problem with Time Windows (VRPTW)	19
2.3 Algoritma Genetika	20
2.3.1 Struktur Algoritma Genetika.....	20
2.3.2 Representasi Kromosom	22
2.4 Fungsi Evaluasi (<i>Fitness</i>)	23
2.5 Seleksi	23

2.5.1	Elitis	24
2.5.2	<i>Binary Tournament</i>	24
2.6	Operator Genetika	25
2.6.1	Persilangan (<i>Crossover</i>)	26
2.6.2	Mutasi.....	27
BAB III METODE PENELITIAN DAN PERANCANGAN.....		28
3.1	Studi Pustaka	29
3.1.1	Pengumpulan dan Analisis Data	29
3.1.2	Analisa dan Perancangan Sistem	29
3.2	Proses Algoritma Genetika.....	31
3.2.1	Membuat Populasi Awal	31
3.2.2	Proses Crossover	32
3.2.3	Proses Mutasi	33
3.2.4	Hitung Nilai <i>Fitness</i>	34
3.2.5	Proses Seleksi.....	37
3.2	Perhitungan Manual	39
3.3.1	Membuat Populasi Awal	40
3.3.2	Menghitung Nilai <i>Fitness</i>	41
3.3.3	Crossover	44
3.3.4	Mutasi.....	45
3.3.5	Evaluasi	46
3.3.6	Seleksi	46
3.3.7	Memilih Kromosom Terbaik.....	49
3.4	Perancangan User Interface	49
3.5	Perancangan Uji Coba dan Evaluasi.....	51
3.5.1	Uji Coba Banyaknya Generasi	52
3.5.2	Uji Coba Banyaknya Populasi	52
3.5.3	Uji Coba Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	53
3.5.4	Uji Coba Perbandingan Seleksi Menggunakan Metode Elitis dan <i>Binary Tournament</i>	54

BAB IV IMPLEMENTASI	55
4.1 Lingkungan Implementasi	55
4.1.1 Lingkunan Perangkat Keras	55
4.1.2 Lingkungan Perangkat Lunak	55
4.2 Implementasi Program	56
4.2.1 Struktur Data	56
4.2.2 Membangkitkan Individu Awal	56
4.2.3 Pembagian Kapasitas Kendaraan	57
4.2.4 Perhitungan Nilai Fitness	58
4.2.5 Mengambil Data Jarak Pelanggan	59
4.2.6 Proses Crossover	59
4.2.7 Proses Mutasi	61
4.2.8 Proses Seleksi Elitis	61
4.2.9 Proses Seleksi Binary Tournament	63
4.2.10 Proses Pemilihan Kromosom Terbaik	63
4.3 Implementasi User Interface	64
BAB V PENGUJIAN DAN ANALISA	67
5.1 Pengujian dan Analisa Banyaknya Generasi	67
5.2 Pengujian dan Analisa Ukuran Populasi	69
5.3 Pengujian dan Analisa Kombinasi Crossover rate dan <i>Mutation rate</i> ...	70
5.4 Pengujian dan Analisa Perbandingan Seleksi Metode Elitis dan <i>Binary Tournament</i>	72
BAB VI KESIMPULAN DAN SARAN	74
6.1 Kesimpulan	74
6.2 Saran	75
DAFTAR PUSTAKA	76

DAFTAR GAMBAR

Gambar 2.1 Mekanisme Kerja Algoritma Genetik	22
Gambar 2.2 Mekanisme Seleksi Turnamen	25
Gambar 2. 3 Area Pemetaan PMX	26
Gambar 2. 4 Hasil Protochild PMX	26
Gambar 2. 5 Pemetaan PMX	27
Gambar 2. 6 Hasil Keturunan PMX	27
Gambar 2. 7 Contoh Mutasi	27
Gambar 3. 1 Diagram Alir Penelitian	28
Gambar 3. 2 Flowchart Algoritma Genetik Pada Sistem	30
Gambar 3. 3 Flowchart Membangkitkan Populasi Awal	32
Gambar 3. 4 Flowchart Proses Crossover	33
Gambar 3. 5 Flowchart Proses Mutasi	34
Gambar 3. 6 Flowchart Proses Pengisian Muatan	35
Gambar 3. 7 Proses Perhitungan Nilai Fitness	36
Gambar 3. 8 Flowchart Proses Seleksi Elitis	37
Gambar 3. 9 Flowchart Binary Tournament Selection	38
Gambar 3. 10 Proses Pembagian Gen dalam Truk	41
Gambar 3. 11 Hasil Crossover PMX	45
Gambar 3. 12 Hasil Mutasi	45
Gambar 3. 13 Halaman Utama Perancangan User Interface	49
Gambar 3. 14 Tampilan Data Jarak	50
Gambar 3. 15 Detail Kromosom Terbaik	51
Gambar 4. 1 Implementasi User Interface Halaman Utama	64
Gambar 4. 2 Implementasi User Interface Halaman Data	65
Gambar 4. 3 Implementasi User Interface Detail Hasil Proses GA	66
Gambar 5. 1 Grafik Hasil Uji Coba Banyak Generasi	68
Gambar 5. 2 Grafik Hasil Uji Coba Banyak Populasi	70
Gambar 5. 3 Grafik Hasil Uji Coba Kombinasi Mutation rate dan Crossover rate	71
Gambar 5. 4 Grafik Hasil Uji Coba Metode Seleksi	73

DAFTAR TABEL

Tabel 3. 1 Data Pelanggan	39
Tabel 3. 2 Data Jarak Pelanggan	40
Tabel 3. 3 Populasi Awal	40
Tabel 3. 4 Data Tujuan Truk 1	42
Tabel 3. 5 Data Tujuan Truk 2	42
Tabel 3. 6 Data Tujuan Truk 3	42
Tabel 3. 7 Data Tujuan Truk 4	43
Tabel 3. 8 Perhitungan Nilai Fitness	43
Tabel 3. 9 Hasil Fitness Setiap Individu	44
Tabel 3. 10 Hasil Fitness Populasi Baru	46
Tabel 3. 11 Proses Seleksi Elitis	47
Tabel 3. 12 Hasil Seleksi Elitis	47
Tabel 3. 13 Proses Pemilihan Individu	48
Tabel 3. 14 Individu Yang Terpilih Sebagai Winner	48
Tabel 3. 15 Hasil Kromosom Terbaik	49
Tabel 3. 16 Skenario Uji Coba Banyak Generasi	52
Tabel 3. 17 Skenario Uji Coba Banyaknya Populasi	53
Tabel 3. 18 Skenario Uji Coba Nilai Kombinasi Cr dan Mr	53
Tabel 3. 19 Skenario Percobaan Metode Seleksi	54
Tabel 5. 1 Hasil Uji Coba Banyak Generasi	67
Tabel 5. 2 Hasil Uji Coba Ukuran Populasi	69
Tabel 5. 3 Hasil Uji Coba Kombinasi Cr dan Mr	71
Tabel 5. 4 Hasil Uji Coba Metode Seleksi	72

DAFTAR SOURCE CODE

<i>Source code 4. 1 Struktur Data</i>	56
<i>Source code 4. 2 Membangkitkan Populasi Awal</i>	57
<i>Source code 4. 3 Pembagian Kapasitas Kendaraan</i>	58
<i>Source code 4. 4 Perhitungan Nilai Fitness</i>	59
<i>Source code 4. 5 Mengambil Data Jarak Pelanggan</i>	59
<i>Source code 4. 6 Proses Crossover</i>	61
<i>Source code 4. 7 Proses Mutasi</i>	61
<i>Source code 4. 8 Proses Seleksi Elitis.....</i>	62
<i>Source code 4. 9 Proses Seleksi Binary Tournament.....</i>	63
<i>Source code 4. 10 Proses Pemilihan Kromosom Terbaik.....</i>	64



BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyaluran Beras Bersubsidi Bagi Kelompok Masyarakat Berpendapatan Rendah (Raskin) merupakan program nasional dari Pemerintah Pusat hingga pemerintah Daerah untuk meningkatkan kesejahteraan dalam memenuhi kebutuhan pangan pada masyarakat berpendapatan rendah. Melalui program ini diharapkan dapat mengurangi beban pengeluaran Rumah Tangga Sasaran (RTS) dalam bentuk beras dengan jumlah yang sudah ditentukan. Keberhasilan Program Raskin diukur berdasarkan tingkat pencapaian indikator 6T, yaitu: tepat sasaran, tepat jumlah, tepat harga, tepat waktu, tepat kualitas, dan tepat administrasi. Penyaluran Beras Bersubsidi telah dilakukan sesuai peraturan pemerintah RI No.7 tahun 2003 menyatakan bahwa Perum Bulog yang bertanggung jawab atas tugas pokok dalam kegiatan pendistribusian beras miskin (Raskin) [BAF-13].

Perum Bulog melakukan penyaluran Raskin kesetiap pelanggan dengan jumlah permintaan yang berdeda-beda. Karena hal tersebut maka pihak distributor memiliki kesulitan saat melakukan pengiriman beras ke beberapa tujuan dengan permintaan, waktu jarak tempuh, dan pelayanan pelanggan yang berbeda-beda. Distributor memiliki jam kerja yang sudah ditentukan untuk melakukan pengiriman beras kepada setiap pelanggan. Seringkali menjadi masalah jika pengiriman beras membutuhkan waktu lebih lama hingga melewati jam kerja pengiriman karena dipengaruhi oleh beberapa faktor yaitu; waktu tempuh pada setiap titik tujuan dan kapasitas kendaraan. Maka solusi yang perlu diperhatikan adalah menghitung rute pada setiap tujuan yang dimulai dari pusat keberangkatan kendaraan (Gudang Distribusi). Perhitungan rute menyatakan kendaraan akan mengunjungi pelanggan tepat hanya satu kali sesuai dengan urutan rute terpendek hingga kembali lagi pada tempat semula yaitu gudang distribusi [SUN-11].

Perhitungan rute terpendek ini dapat diselesaikan dengan Algoritma Genetika yang merupakan teknik pencarian untuk menemukan hasil yang lebih

baik. Dalam proses perhitungan digunakan representasi kromosom terhadap suatu populasi yang akan menghasilkan keturunan hingga generasi terbaik. Generasi terbaik didapatkan dari pemilihan nilai *fitness* pada setiap populasi yang dilakukan secara berulang-ulang hingga kondisi terpenuhi [YOZ-13].

Dalam menyelesaikan masalah yang telah diuraikan diatas, untuk menghitung rute optimal dengan batas waktu pelayanan yang sudah ditentukan, maka yang dapat digunakan adalah metode Algoritma Genetika untuk *Vehicle Routing Problem with Time Windows* (VRPTW). VRP merupakan solusi permasalahan dalam optimasi penentuan rute dengan kapasitas kendaraan yang terbatas. *Time Windows* merupakan waktu pelayanan (*time frame*) buka-tutup yang sudah ditentukan oleh pelanggan, jika kendaraan datang sebelum *time windows* "open" maka akan dikenakan waktu tunggu pelayanan, dan jika waktu kendaraan datang melebihi waktu tutup (*time windows* "closed") pelayanan = 0 dan kendaraan akan mendapatkan pinalti keterlambatan.

Pada penelitian sebelumnya, algoritma genetika mampu menyelesaikan berbagai macam permasalahan yang berhubungan dengan pencarian rute, misalnya pencarian rute kunjungan wisata yang dibatasi oleh waktu buka tutup tempat wisata tersebut [WID-2010], pencarian rute untuk beberapa permasalahan selesman [MAH-08]. Hal ini membuktikan kemampuan algoritma genetika untuk permasalahan kombinatorial ini [MAH-13]. Karena itu, pada skripsi ini digunakan algoritma genetika untuk menemukan solusi optimal pada sistem pencarian rute terpendek. Dengan demikian judul yang diangkat dalam skripsi ini adalah **"Penerapan Algoritma Genetik Untuk Vehicle Routing Problem With Time Windows (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi"**.

1.2 Rumusan Masalah

Permasalahan yang dapat dirumuskan dari latar belakang diatas adalah :

1. Bagaimana menerapkan algoritma genetika untuk menyelesaikan permasalahan VRPTW pada distribusi beras bersubsidi.
2. Bagaimana pengaruh parameter genetika (*crossover rate*, *mutation rate*, jumlah populasi dan jumlah generasi) terhadap hasil nilai *fitness* yang didapatkan.

1.3 Batasan Masalah

Dari rumusan masalah diatas, maka batasan masalah yang digunakan dalam masalah optimasi distribusi beras bersubsidi dengan algoritma genetika adalah:

1. Kajian yang dilakukan berdasarkan data simulai yang disesuaikan dengan keadaan sebenarnya.
2. Terdapat kromosom sepanjang 20. Susunan gen dalam kromosom diasumsikan sebagai pelanggan.
3. Setiap pelanggan memiliki jumlah permintaan, waktu pelayanan, dan jarak tempuh yang berbeda-beda.
4. Distribusi dimulai pukul 07.00 dan dilakukan dalam sehari *one day service* dengan mengunjungi pelanggan tepat hanya satu kali.
5. Distributor mengunjungi pelanggan sesuai dengan waktu buka – tutup pelayanan yang sudah ditentukan oleh pelanggan.
6. Kedatangan kendaraan sebelum waktu buka pelanggan maka akan dikenakan waktu tunggu.
7. Keterlambatan kendaraan datang melewati batas waktu tutup, maka akan dikenakan waktu pinalti dan pelanggan terlewat.
8. Kondisi jalan diasumsikan lancar pada dua arah

1.4 Tujuan

Tujuan yang ingin dicapai dalam penelitian untuk tugas akhir ini adalah :

1. Menerapkan algoritma genetika dengan metode *Vehicle Routing Problem with Time Windows* (VRPTW) untuk optimasi distribusi beras bersubsidi keseluruh pelanggan.
2. Mengetahui dan mengukur pengaruh parameter genetika (*crossover rate*, *mutation rate*, dan jumlah populasi dan jumlah generasi) terhadap hasil dari optimasi distribusi beras bersubsidi.

1.5 Manfaat

Manfaat yang dapat diambil dari penulisan tugas akhir ini adalah didapatkan sebuah sistem berbasis algoritma genetika yang mampu menemukan optimasi rute terpendek dengan nilai pinalti terkecil sehingga dapat menghemat waktu untuk distribusi beras bersubsidi.

1.6 Sistematika Penulisan

Sistematika penulisan diuraikan untuk memberikan gambaran dalam tugas akhir ini. Mencakup enam bab dengan masing-masing bab diuraikan secara garis besar pembahasan :

BAB I : Pendahuluan

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan tugas akhir.

BAB II : Tinjauan Pustaka

Menguraikan teori dan referensi tentang algoritma genetic, VRPTW, serta metode-metode lain yang mendasari proses implementasi sisem pada tugas akhir ini.

BAB III : Metodologi Penelitian dan Perancangan

Membahas tentang metode Algoritma Genetika yang digunakan dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi sistem Optimasi Distribusi Beras Bersubsidi.

BAB IV : Implementasi

Menguraikan spesifikasi sistem dan proses implementasi pada sistem Optimasi Distribusi Beras Bersubsidi dengan Algoritma Genetika.

BAB V : Pengujian dan Analisis

Membahas hasil pengujian dan analisa hasil pengujian terhadap implementasi perangkat lunak Optimasi Distribusi Beras Bersubsidi.

BAB VI : Kesimpulan dan Saran

Menguraikan kesimpulan yang dihasilkan dari implementasi dan pengujian perangkat lunak yang dikembangkan pada sistem Optimasi Distribusi Beras Bersubsidi sesuai dengan metode VRPTW pada Algoritma Genetika, serta berisi saran-saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Pelaksanaan Distribusi Raskin

Indonesia tercatat sebagai negara dengan penduduk yang hidup dibawah garis kemiskinan. Pemerintah berupaya menanggulagi kemiskinan masyarakat dengan berbagai kebijakan yang layak untuk kehidupan masyarakat miskin. Berdasarkan Surat Kementerian Koordinator Kesejahteraan Rakyat Nomor B-2143/KMK/Dep.II/XI/2007 tanggal 30 November 2007 kebijakan pemerintah yang dilakukan dalam upaya penanggulangan masyarakat miskin adalah bantuan beras bersubsidi. Program bantuan tersebut merupakan kebijakan beras untuk rumah tangga miskin (RASKIN) yang mulai ditetapkan sejak tahun 2002 dengan tujuan memberikan penjelasan mengenai arti program RASKIN sebenarnya melalui pelaksanaan di lapangan. Program RASKIN merupakan program bantuan dimana beras didistribusikan kepada setiap kepala rumah tangga miskin (RTM) dengan ketentuan berdasarkan Surat Deputi Menko Kesra Bidang Perlindungan Sosial dan Perumahan Rakyat No. B.216/KMK/Dep.II/II/2008 tanggal 5 Februari 2008 menyatakan bahwa setiap RTM memperoleh 10 Kg hingga 15 Kg selama 10 bulan dengan harga Rp. 1.600,-/kg netto yang telah diakumulasi dari Rp. 4.616/harga beras sesuai dengan HPP harga pembelian oleh pemerintah harga dan Rp. 3.016,- dana subsidi oleh pemerintah/APBN. Untuk melakukan distribusi secara merata pada RTM maka pemerintah mengirimkan beras pada setiap kecamatan dengan permintaan beras yang sesuai dengan jumlah RTM yang sudah tercatat. Hal ini merupakan kebijakan pemerintah yang telah bertanggung jawab agar memberikan bantuan secara rutin dan konsisten terhadap pelaksanaan bantuan masyarakat miskin [KUS-08].

2.1.1 Sistem Distribusi Beras

Distribusi beras bersubsidi dilakukan dengan mengirimkan beras kesetiap kecamatan oleh Perum Bulog sesuai dengan data permintaan RTM. Sistem distribusi beras bersubsidi meliputi [PUT-08] :

- a. Perum Bulog menyediakan beberapa kendaraan khusus sesuai kapasitas yang dibutuhkan untuk melakukan proses distribusi.
- b. Proses distribusi berawal dari gudang Bulog sebagai keberangkatan kendaraan yang sudah berisi muatan beras sesuai permintaan pelanggan.
- c. Setiap kendaraan berangkat dengan jumlah muatan yang berbeda-beda karena pada setiap tujuan (kecamatan) memiliki data permintan RTM yang tidak sama.
- d. Pada saat kendaraan sampai, kendaraan memiliki waktu tunggu jika tidak sesuai dengan waktu buka pada setiap kecamatan.
- e. Setelah proses distribusi dilakukan, kendaraan akan kembali dan berakhir pada tempat semula yaitu gudang Bulog.

2.1.2 Pemilihan Rute

Pemilihan rute pada setiap proses pengantaran barang merupakan faktor penting yang perlu diperhatikan. Pada proses pengiriman barang diperlukan optimasi rute jarak tempuh sehingga akan meminimalkan waktu tempuh perjalanan dan meminimalkan biaya transportasi yang digunakan saat proses distribusi berlangsung. Proses distribusi dilakukan dalam sehari sesuai dengan jam kerja pelaksanaan pengiriman beras ke beberapa tujuan. Keberhasilan dalam suatu proses distribusi diukur dari optimasi waktu jarak tempuh, waktu pembongkaran barang, waktu tunggu buka pelayanan dengan kemungkinan akan mengakibatkan keterlambatan kendaraan kembali. Agar proses distribusi tetap berjalan lancar perlu adanya solusi untuk optimasi beberapa faktor yang dapat mempengaruhi penggunaan waktu dan biaya yang lebih. Pemilihan rute dengan perhitungan jarak tempuh merupakan faktor utama yang mempengaruhi penggunaan waktu dan biaya dalam proses distribusi. Dengan menghitung jarak tempuh kemudian memutuskan urutan tujuan sesuai rute optimum yang terpilih maka dapat mempengaruhi waktu minimum [PUT-08].

2.2 Vehicle Routing Problem (VRP)

Pada tahun 1959, Dantzing dan Ramser menemukan *Vehicle Routing Problem* VRP pertama kali. VRP merupakan program non-linear untuk pencarian sebuah

solusi dalam memecahkan suatu masalah. Bentuk permasalahan VRP adalah salah satu masalah optimasi kombinatorial yang paling banyak dipelajari dan memperhatikan gambaran rute optimal yang dapat dilalui kendaraan untuk melayani pelanggan. Setiap kendaraan memiliki kapasitas angkut dan setiap pelanggan memiliki permintaan (*demand*). Setiap pelanggan dikunjungi kendaraan tepat satu kali dengan total *demand* tiap rute tidak boleh melebihi kapasitas angkut kendaraan. Dalam VRP kendaraan harus berangkat dan kembali pada tempat distribusi yang sama, tempat tersebut biasa dikenal dengan istilah depot. Faktor yang sering muncul pada VRP adalah masalah kapasitas yang dikenal dengan nama *Capacitated Vehicle Routing Problem* (CVRP) serta permasalahan pada batasan waktu yang dikenal dengan nama *Vehicle Routing Problem with Time Windows* (VRPTW). Kedua permasalahan ini dapat digabungkan dengan prioritas utama yaitu semua permintaan terpenuhi ada batas waktu tertentu [TOT-02].

2.2.1 Vehicle Routing Problem with Time Windows (VRPTW)

Vehicle Routing Problem with Time Windows (VRPTW) merupakan bentuk metode perluasan dari VRP dalam kendala penentuan kapasitas sesuai kebutuhan yang terkait dengan kendala pengantaran pada masing-masing pelanggan yang memiliki interval waktu atau biasa disebut *time windows*. Pada VRPTW permasalahan yang dapat dipecahkan adalah masalah kapasitas angkut kendaraan dalam *time frame* dan kendaraan harus melayani setiap konsumen pada *time frame* tertentu. VRPTW dapat digambarkan sebagai masalah perancangan optimasi rute dengan pengeluaran biaya yang minimal dari suatu depot beberapa titik tujuan. Rute harus dirancang sesuai perhitungan jarak kesetiap tujuan agar setiap titik hanya dikunjungi sekali saja oleh kendaraan. Semua rute yang dilalui harus berangkat dan berakhir di depot yang sama dengan total *demand* semua titik pada rute tersebut tidak boleh melebihi kapasitas angkut kendaraan. VRPTW dibagi menjadi dua bagian yaitu *Vehicle Routing Problem with Hard Time Windows* (VRPHTW) dan *Vehicle Routing Problem with Soft Time Windows* (VRPSTW). Dimana pada VRPHTW konsumen hanya dapat dilayani selama selang waktu yang telah ditentukan, sedangkan pada VRPSTW konsumen dapat dilayani setiap saat dengan

asumsi apabila melewati batas waktu pelayanan yang sudah ditentukan akan dikenakan biaya tambahan atau pinalti [KAL-01].

2.3 Algoritma Genetika

Algoritma genetika adalah algoritma yang mengadaptasi istilah dalam proses genetik yang ada dalam suatu populasi makhluk hidup. Konsep yang mendasari timbulnya teori genetika yang dikemukakan oleh Charles Darwin menjelaskan bahwa pada proses genetika alami, setiap individu harus melakukan adaptasi terhadap lingkungan disekitarnya agar dapat bertahan hidup [SET-03].

Algoritma genetika merupakan cabang dari algoritma evolusi dengan metode *adaptive* yang pertama kali diperkenalkan oleh John Holland dari Universitas Michigan pada tahun 1975. Algoritma genetika merupakan metode pencarian solusi yang dapat digunakan untuk memecahkan pencarian nilai dalam masalah optimasi [GEN-00]. Pencarian dalam algoritma genetika berdasar pada mekanisme biologis dalam beragam evolusi yang berupa variasi kromosom pada setiap individu organisme. Dimana variasi kromosom akan mempengaruhi sistem reproduksi dan tingkat kemampuan hidup organisme sebagai generasi penerus terbaik [KUS-03].

Algoritma genetika memiliki 5 komponen penting, yaitu [MIC-99]:

1. Representasi genetik sebagai solusi dari sebuah masalah
2. Cara untuk membangkitkan populasi awal
3. Fungsi untuk mengevaluasi solusi dengan nilai *fitness* pada setiap individu
4. Beberapa operator genetika yang menghasilkan keturunan *offspring*
5. Nilai parameter yang mencakup ukuran populasi dan nilai probabilitas yang digunakan dalam operator genetika.

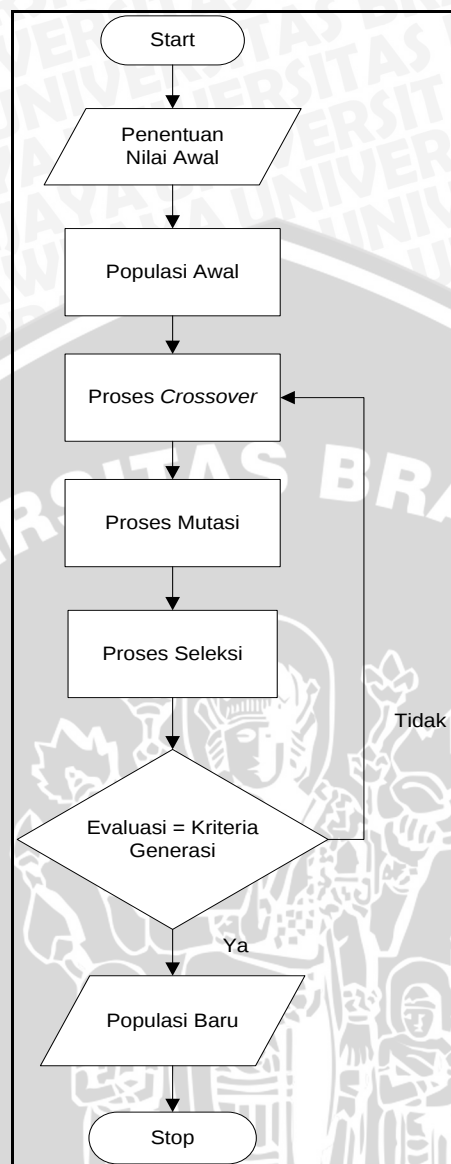
2.3.1 Struktur Algoritma Genetika

Algoritma genetika merupakan teknik pencarian sejumlah kemungkinan pada suatu populasi untuk mendapatkan solusi terbaik. Kromosom atau individu merupakan hasil solusi yang didapatkan pada suatu populasi. Dimana populasi awal dibangkitkan secara acak kemudian pada populasi berikutnya merupakan hasil evolusi pada setiap kromosom melalui proses iterasi, hal ini disebut dengan

generasi. Pada setiap generasi terdapat proses evaluasi kromosom yang nantinya akan menghasilkan nilai solusi melalui perhitungan fungsi *fitness*. Perhitungan nilai *fitness* merupakan suatu alat ukur pencapaian solusi dengan nilai terbaik yang akan menunjukkan kualitas kromosom dalam populasi tersebut terpilih sebagai generasi penerus [KUS-03].

Pada setiap proses iterasi dilakukan, satu anggota populasi baru akan dikembalikan sebagai solusi dari suatu permasalahan. Berikut merupakan proses evolusi yang dilakukan dalam setiap iterasi [SUY-11] :

1. Memilih dua individu sebagai orang tua (*parent*) yang kemudian dilakukan perkawinan silang (*crossover*) untuk menghasilkan individu baru sebagai anak (*offspring*).
2. Memilih satu individu sebagai orang tua (*parent*) yang kemudian akan dilakukan perubahan gen melalui operator mutasi (*mutation*) pada tingkat tertentu.
3. Menerapkan skema pergantian (*replacement scheme*) untuk menghasilkan populasi baru.
4. Dilakukan iterasi dengan cara mengulang seluruh proses evolusi hingga berhenti pada kondisi tertentu (*stopping condition*). Kondisi berhenti dapat ditentukan dari jumlah iterasi yang diinginkan, berdasarkan waktu tertentu atau ketika didapatkan variasi individu terbaik yang memiliki nilai terbesar dalam suatu populasi.
5. Seleksi nilai *fitness* terbesar yang dihasilkan dari iterasi didapatkan dari populasi baru yang memiliki individu-individu yang lebih baik dari individu lama sebelumnya. Mekanisme kerja algoritma genetik dapat dilihat pada Gambar 2.1 berikut [HAN-02].



Gambar 2.1 Mekanisme Kerja Algoritma Genetik

2.3.2 Representasi Kromosom

Representasi kromosom merupakan langkah awal dalam penerapan algoritma genetika. Dimana dalam suatu permasalahan direpresentasikan dalam bentuk kromosom sesuai dengan bentuk permasalahan yang akan dipecahkan. Hal ini biasa disebut dengan pengkodean (*encoding*) [SUY-05].

Pada penelitian ini menggunakan representasi permutasi yang telah diperkenalkan oleh Gen dan Cheng pada tahun 1997. Representasi permutasi bisa digunakan untuk menyatakan sebuah solusi penjadwalan dan perjalanan. Setiap gen

pada kromosom berupa angka integer yang menyatakan nomor tiap simpul. Misalkan pada sebuah kromosom [2 3 4 1 5] menyatakan bahwa perjalanan dimulai dari simpul 2 kemudian secara berurutan mengunjungi simpul [3 4 1 5] dan kemudian kembali pada simpul 2 [MAH-13].

2.4 Fungsi Evaluasi (*Fitness*)

Evaluasi dilakukan terhadap suatu individu berdasarkan fungsi tertentu sebagai ukuran tingkat keberhasilannya. Untuk menyatakan baik tidaknya suatu individu dinyatakan dalam suatu nilai *fitness*. Di dalam kenyataan evolusi alam sesungguhnya individu yang bernilai *fitness* rendah akan mati. Namun untuk masalah optimasi, nilai *fitness* dapat digunakan pada mencari solusi fungsi maksimal (masalah maksimasi) yang dapat dihitung secara langsung dengan persamaan [MAH-13].

$$fitness = f(x) \quad (2.1)$$

Serta untuk permasalahan optimasi fungsi minimal (masalah minimasi) pada VRPTW nilai *fitness* tidak dapat dihitung secara langsung, maka persamaan yang dapat digunakan adalah persamaan berikut [MAH-13].

$$fitness = C - f(x) \quad (2.2)$$

$$fitness = \frac{1}{f(x)+c} \quad (2.3)$$

Nilai C merupakan nilai konstan yang harus ditetapkan sebelumnya [MAH-13].

2.5 Seleksi

Proses seleksi dilakukan untuk melakukan pemilihan terhadap individu yang ada dalam proses reproduksi. Seleksi diperoleh dari pemilihan nilai *fitness* yang dihasilkan dengan tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang memiliki nilai *fitness* terbaik. Individu terbaik yang terpilih dari hasil seleksi akan menjadi *parent* sebagai generasi penerus untuk reproduksi berikutnya [WID-12].

Dalam proses seleksi keanekaragaman populasi memegang peran penting sebagai daerah *sampling*, probabilitas seleksi, dan mekanisme seleksi [SET-03]. Sebelum dilakukan seleksi sejumlah anggota populasi ditambah dengan hasil *offspring* dari proses operasi genetik *crossover* dan mutasi. Hasil dari operasi genetik dengan populasi semula akan dilakukan seleksi dengan metode seleksi tertentu untuk diambil sejumlah n anggota populasi yang terbaik sesuai dengan jumlah populasi awal [WID-12].

Proses seleksi dapat dilakukan secara proporsional berdasarkan nilai-nilai *fitness* yang dihasilkan. Terdapat beberapa macam metode seleksi antara lain adalah : *Tournament Selection*, *Roulette Wheel Selection*, *Rank Based Fitness Selection* [KUS-03].

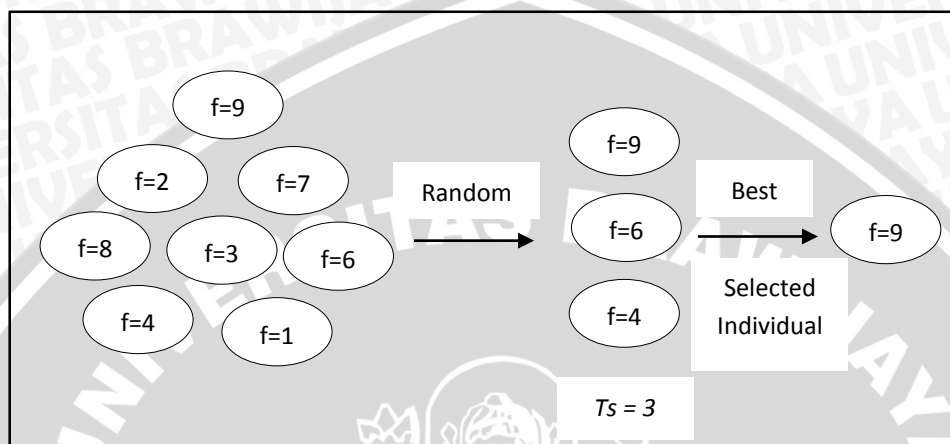
2.5.1 Elitis

Seleksi *Elitis* merupakan metode seleksi dimana individu dalam suatu populasi akan diurutkan berdasarkan hasil objektif. Pada proses seleksi ini fungsi *fitness* digunakan sebagai parameter untuk menentukan tingkat kesempurnaan suatu kromosom yang akan bertahan hidup dan bisa menjadi *parent* untuk proses reproduksi pada generasi berikutnya. Sifat pemilihan individu pada seleksi *elitism* ini adalah hanya memilih kromosom dengan nilai *fitness* tinggi yang dapat bertahan hidup dan dapat mewariskan karakteristik terbaik pada generasi berikutnya. Sebaliknya untuk individu yang memiliki nilai *fitness* rendah akan hilang pada generasi berikutnya [SUY-11].

2.5.2 Binary Tournament

Tournament selection merupakan salah satu metode seleksi yang paling populer dalam algoritma genetika karena efisiensi dan implementasi yang sederhana. Dalam seleksi turnamen, n individu dipilih secara acak sebanyak besar populasi (*pop-size*). Banyaknya perbandingan dalam turnamen terhadap individu biasanya disebut dengan *tournament size*. Satu individu akan bersaing dengan individu lain untuk menentukan nilai *fitness* tertinggi yang nantinya akan menjadi pemenang, dan individu sebagai pemenang akan terpilih dalam populasi generasi berikutnya. Seleksi turnamen juga memberikan kesempatan pada semua individu

terpilih untuk mempertahankan keragamannya. Pemilihan turnamen memiliki beberapa keunggulan yang meliputi efisien kompleksitas waktu, terutama jika dilaksanakan secara paralel, dan tidak ada persyaratan untuk skala nilai *fitness*. Pada Gambar 2.2 menggambarkan mekanisme seleksi turnamen [RAZ-11].



Gambar 2.2 Mekanisme Seleksi Turnamen

Pada gambar diatas menyatakan bahwa *tournament size* (T_s) sebanyak tiga perbandingan, yang berarti bahwa tiga kromosom bersaing satu sama lain. Hanya kromosom yang memiliki nilai terbaik akan dipilih untuk reproduksi pada generasi selanjutnya. Dalam seleksi turnamen, nilai yang lebih besar dari *tournament size* menyebabkan hilangnya keanekaragaman individu yang diharapkan. Semakin besar *tournament size* maka populasi yang terbentuk semakin kecil untuk memberikan kontribusi terhadap keanekaragaman genetik. Terdapat dua faktor yang menyebabkan hilangnya keanekaragaman dalam seleksi turnamen regular yaitu beberapa individu mungkin tidak mendapatkan sampel untuk berpartisipasi dalam turnamen sama sekali, sementara individu lain mungkin tidak dipilih untuk populasi baru karena individu tersebut telah kehilangan turnamen [RAZ-11].

2.6 Operator Genetika

Operator genetika merupakan suatu metode reproduksi pada setiap individu agar mendapatkan karakteristik individu baru melalui *parent*. Hal ini dilakukan guna mendapatkan karakteristik individu yang terbaik yang akan mewarisi karakteristik individu pada generasi selanjutnya. Terdapat dua operator genetika yaitu *crossover* dan mutasi.

2.6.1 Persilangan (Crossover)

Crossover merupakan proses persilangan yang dilakukan pada dua individu yang dipilih secara acak sebagai *parent* untuk menghasilkan individu baru (*offspring*) sebagai anak. Jumlah persilangan pada suatu populasi ditentukan oleh parameter yang disebut dengan *crossover rate/pc*, jadi tidak semua kromosom mengalami *crossover*. Dalam tahap ini harus ditentukan tingkat *crossover* (*crossover rate / pc*). Nilai menyatakan rasio *offspring* yang dihasilkan dari proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $pc \times popSize$ [MAH-13]. Terdapat beberapa operator *crossover* yaitu: *partially-mapped crossover* (PMX), *order crossover* (OX), *position-based crossover*, *order-based crossover* dan *one-cut-point crossover*. Pada penelitian ini menggunakan *Partially-Mapped Crossover* (PMX) yang dapat mencegah adanya keturunan mirip dengan munculnya gen ganda yang terdapat pada suatu individu. Berikut merupakan langkah-langkah dalam metode PMX [AZM-11]:

1. Tentukan dua titik secara random untuk membagi *parent*. Diantara dua titik tersebut akan terdapat area pemetaan yang disebut dengan *mapping* seperti pada Gambar 2.3 berikut.

Parent 1	1	2	3	4	5	6	7
Parent 2	4	3	7	2	6	1	5

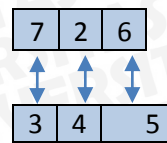
Gambar 2. 3 Area Pemetaan PMX

2. Lakukan penukaran substring antara *parent1* dan *parent2*, substring yang ditukar adalah substring yang berada dalam area *mapping*. Maka akan terbentuk dua protoChild seperti pada Gambar 2.4 berikut.

ProtoChild1	1	2	7	2	6	6	7
ProtoChild2	4	3	3	4	5	1	5

Gambar 2. 4 Hasil Protochild PMX

3. Definisikan hubungan pemetaan yang berada dalam area *mapping*. Pemetaan dilakukan untuk mengganti nilai 7, 2, 6 menjadi 3, 4, 5 seperti pada Gambar 2.5 berikut.



Gambar 2. 5 Pemetaan PMX

4. Lakukan pemetaan kromosom dari awal sesuai *ProtoChild1* dan *ProtoChild2*, pertahankan substring yang berada dalam area pemetaan, jika terdapat angka yang sama pada area dalam dan luar pemetaan, maka angka yang harus diganti adalah angka yang berada diluar area pemetaan. Pergantian angka sesuai dengan hasil definisi hubungan pemetaan sebelumnya hingga menghasilkan keturunan dengan gen yang berbeda dengan *parent* seperti pada Gambar 2.6 berikut.

<i>Child1</i>	1	4	7	2	6	5	3
<i>Child2</i>	2	7	3	4	5	1	6

Gambar 2. 6 Hasil Keturunan PMX

2.6.2 Mutasi

Operator genetika yang dilakukan setelah *crossover* adalah mutasi. Sama seperti proses *crossover* dimana nilai tingkat mutasi (*mutation rate / pm*) juga harus ditentukan. Nilai tersebut menyatakan rasio *offspring* yang dihasilkan pada proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $pm \times popSize$ [MAH-13]. Dalam penelitian ini, menggunakan metode mutasi *reciprocal exchange mutation* yaitu memilih individu sebagai induk secara acak yang akan dirubah susunan gen didalamnya. Perubahan dilakukan pada dua posisi dalam kromosom yang ditentukan secara random. Kemudian dilakukan penukaran nilai pada kedua posisi gen tersebut. Mutasi dilakukan untuk mengembangkan solusi dari populasi awal dengan menghasilkan individu baru yang lebih bervariasi. Mutasi sangat membantu dalam pemilihan solusi terbaik pada populasi baru sehingga dapat mempertahankan keanekaragaman individu dalam populasi [MAH-06]. Pada Gambar 2.7 merupakan contoh mutasi yang dilakukan pada semua gen dalam kromosom.

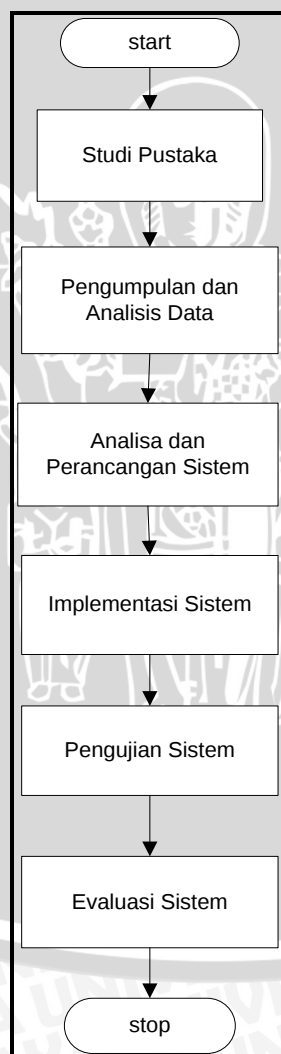
Individu 1	Individu hasil mutasi
1 2 3 4 5 6 7	1 6 3 4 5 2 7

Gambar 2. 7 Contoh Mutasi

BAB III

METODE PENELITIAN DAN PERANCANGAN

Dalam bab ini akan dijelaskan lebih rinci mengenai metodologi dan perancangan untuk implementasi *genetic algorithm* dalam *vehicle routing problem with time windows* pada distribusi beras. Penjelasan tersebut merupakan tujuan untuk memberikan gambaran mengenai sistem yang akan dibuat berdasarkan tinjauan pustaka pada bab sebelumnya. Langkah-langkah yang dilakukan dalam penelitian secara umum ditunjukkan pada Gambar 3.1



Gambar 3. 1 Diagram Alir Penelitian

3.1 Studi Pustaka

Pada bagian ini merupakan proses mencari dan mempelajari literatur yang digunakan sebagai acuan penelitian yang berkaitan dengan algoritma genetika dan VRPTW. Studi pustaka dapat digunakan dari berbagai sumber seperti jurnal, buku, dan internet yang bertujuan untuk meningkatkan pengetahuan dan pemahaman dalam sebuah proses penelitian.

3.1.1 Pengumpulan dan Analisis Data

Pada proses pengumpulan data, peneliti menggunakan data *dummy* yang disesuaikan dengan karakteristik pada kondisi nyata. Pembuatan data disesuaikan berdasarkan wawancara informal dengan pakar yang mengetahui permasalahan VRPTW di lingkungan pekerjaan pakar tersebut. Data kromosom yang dibuat bersifat dinamik. Terdapat susunan gen dalam kromosom yang diasumsikan sebagai jumlah pelanggan yang tersebar secara acak, kemudian dilakukan perjalanan untuk mengunjungi setiap pelanggan, setiap tujuan memiliki jarak dan waktu tempuh yang berbeda-beda. Pada setiap pelanggan dilengkapi dengan data *time windows*, jumlah permintaan, dan lama waktu pelayanan yang ditentukan oleh masing-masing pelanggan. Dari semua data tersebut maka dilakukan pencarian rute optimal yang harus dilalui distributor untuk melakukan pengiriman beras pada setiap kecamatan dengan menggunakan *genetic algorithm*.

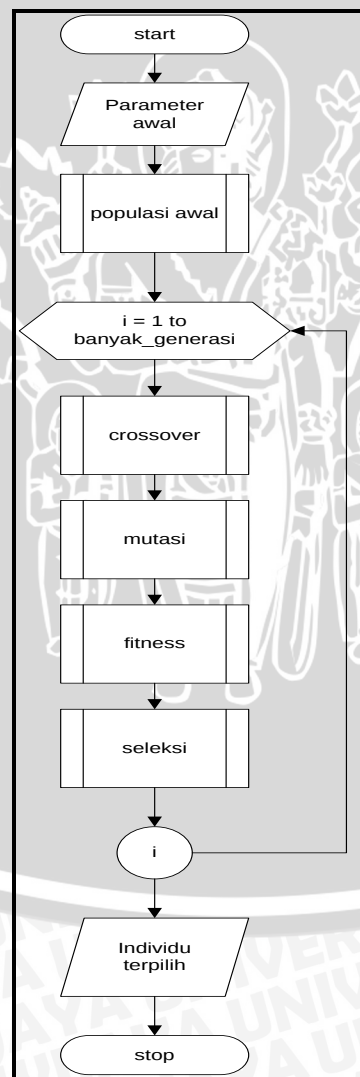
3.1.2 Analisa dan Perancangan Sistem

Sistem yang dibuat merupakan implementasi perangkat lunak dari *genetic algorithm* dengan menggunakan metode VRPTW pada proses distribusi beras bersubsidi. Sistem ini akan menghitung rute terpendek yang akan direkomendasikan pada pihak distributor dengan meminimalkan jarak tempuh, waktu tunggu dan keterlambatan (pinalti). Berikut merupakan batasan-batasan yang dimiliki oleh sistem :

1. Susunan gen dalam kromosom diasumsikan sebagai pelanggan.
2. Jumlah kendaraan yang tersedia adalah 5 kendaraan dalam kecepatan rata-rata 30 km/jam

3. Muatan pada masing-masing kendaraan disesuaikan dengan jumlah permintaan pada setiap pelanggan.
4. Setiap kendaraan berangkat pada waktu yang sama yaitu 07.00.
5. Kendaraan akan menunggu jika waktu sampai tidak sesuai dengan waktu buka pelanggan.
6. Setiap pelanggan dikunjungi tepat satu kali oleh satu kendaraan.
7. Jika kendaraan sampai melebihi waktu tutup, maka diperoleh penalti keterlambatan.

Berdasarkan analisis yang telah dilakukan, maka perancangan sistem pada algoritma genetik diilustrasikan pada Gambar 3.2 berikut.



Gambar 3. 2 Flowchart Algoritma Genetik Pada Sistem

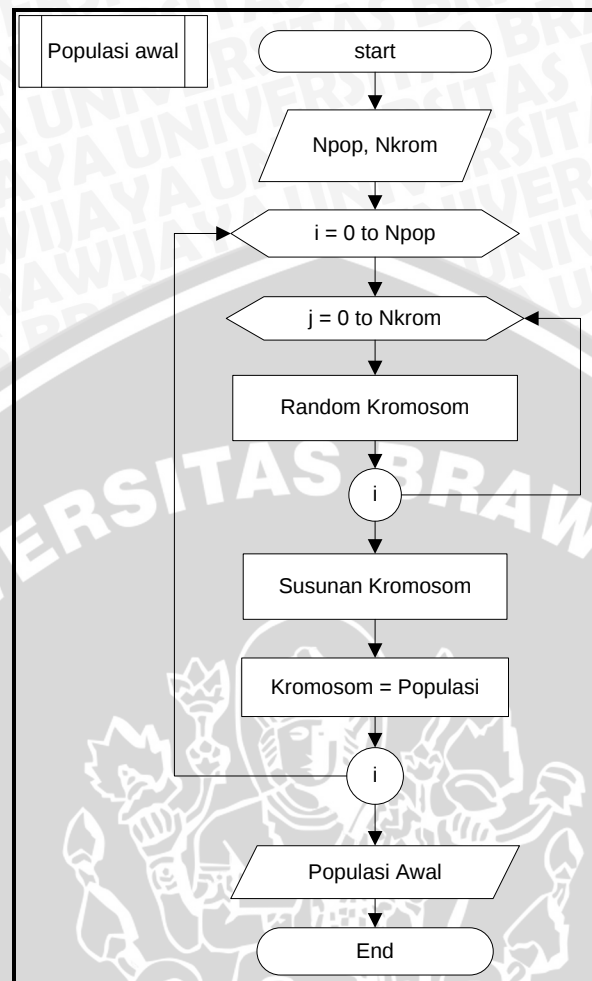
3.2 Proses Algoritma Genetika

Dalam menghitung pencarian rute terpendek menggunakan algoritma genetika memiliki langkah-langkah sebagai berikut:

1. Inisialisasi parameter awal pada :
 - Jumlah individu pada setiap populasi.
 - Jumlah generasi.
 - Tujuan dengan waktu masing-masing.
 - *Crossover rate* (cr).
 - *Mutation rate* (mr).
2. Membangkitkan populasi awal sebanyak *pop-size* yaitu sebanyak jumlah individu yang diinisialisasikan.
3. Membuat populasi baru dalam reproduksi sebanyak Cr dan Mr yang ditentukan untuk menghasilkan anak.
4. Melakukan *crossover* pada individu yang terpilih sebagai *parent* untuk melakukan reproduksi sesuai Cr yang telah ditentukan.
5. Melakukan mutasi pada individu yang terpilih sebagai *parent* untuk melakukan mutasi gen sesuai dengan Mr yang sudah ditentukan.
6. Menghitung nilai *fitness* pada setiap kromosom.
7. Melakukan seleksi untuk menemukan individu terbaik dengan metode *Elitis* dan *Binary Tournament* yang akan bertahan pada generasi selanjutnya.

3.2.1 Membuat Populasi Awal

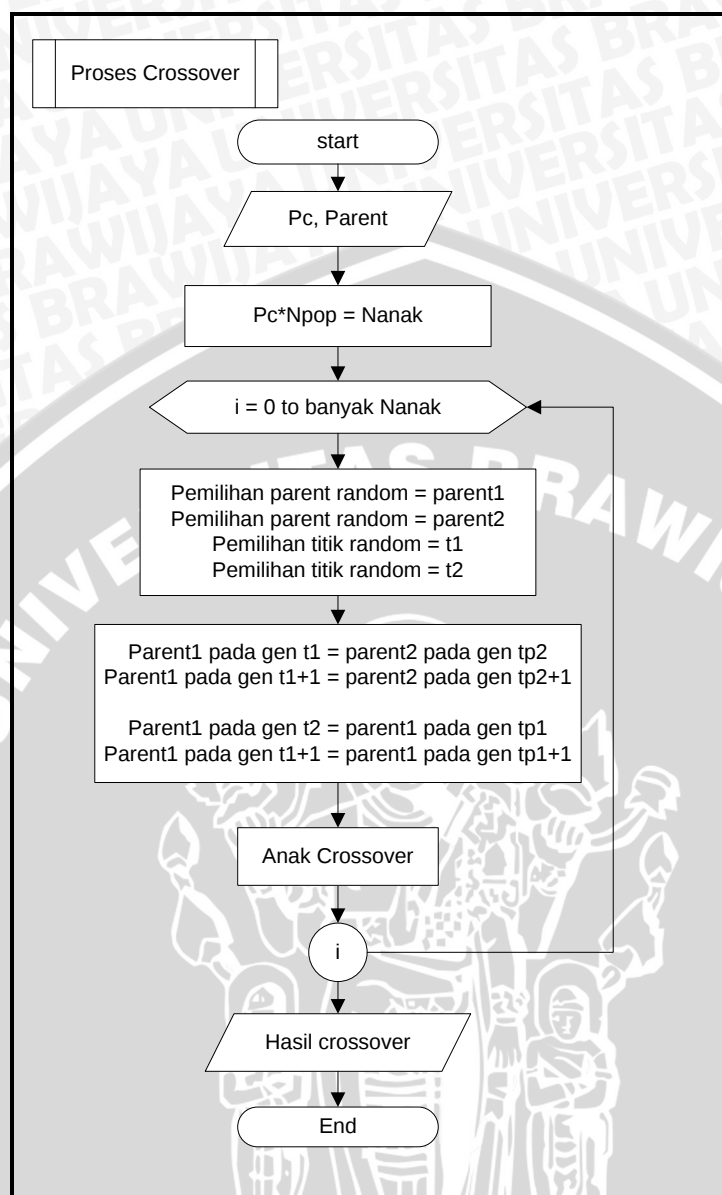
Pada tahap ini merupakan proses awal untuk membangkitkan individu secara acak sebanyak populasi (*pop-size*) yang diinisialisasikan. Setiap individu yang dibangkitkan dibentuk dalam representasi permutasi dengan jumlah kromosom pada suatu individu diasumsikan sebanyak jumlah tujuan yang akan dilewati. Proses pembuatan populasi awal ini dapat dilihat pada Gambar 3.3 berikut.



Gambar 3. 3 Flowchart Membangkitkan Populasi Awal

3.2.2 Proses Crossover

Setelah terbentuk sebuah populasi pada tahap awal, maka selanjutnya adalah melakukan reproduksi dengan cara *crossover* untuk mendapatkan individu baru. Beberapa individu dalam populasi dipilih secara acak sebagai *parent* yang akan menghasilkan anak sebanyak tingkat *crossover*. Dalam tahap ini harus ditentukan tingkat *crossover* (*crossover rate* / *pc*). Nilai menyatakan rasio *offspring* yang dihasilkan dari proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* (anak) sebanyak $pc \times popSize$ yang nantinya akan ditambahkan pada populasi baru. Langkah-langkah yang dilakukan dalam proses *crossover* dapat dilihat pada Gambar 3.4 berikut.

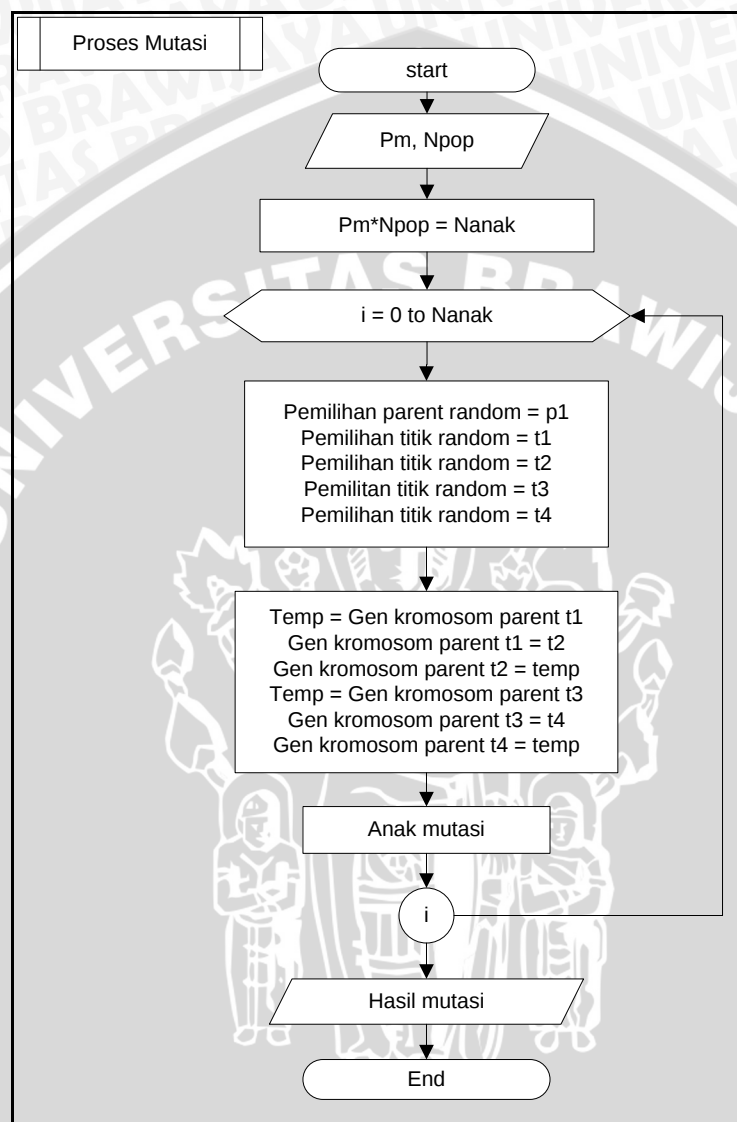


Gambar 3. 4 Flowchart Proses Crossover

3.2.3 Proses Mutasi

Proses selanjutnya dalam reproduksi untuk mendapatkan individu baru adalah proses mutasi. Dalam proses mutasi individu dipilih secara acak dalam satu populasi. Pada individu yang terpilih sebagai induk dilakukan pertukaran gen didalamnya untuk menghasilkan individu baru hasil mutasi (*offspring*). Nilai tingkat mutasi (*mutation rate/ pm*) juga harus ditentukan. Nilai tersebut menyatakan rasio *offspring* yang dihasilkan pada proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak $pm \times popSize$. Individu baru yang

dihasilkan juga akan ditambahkan dalam suatu populasi untuk membentuk populasi baru. Langkah-langkah yang dilakukan pada proses mutasi dapat dilihat pada Gambar 3.5 berikut.



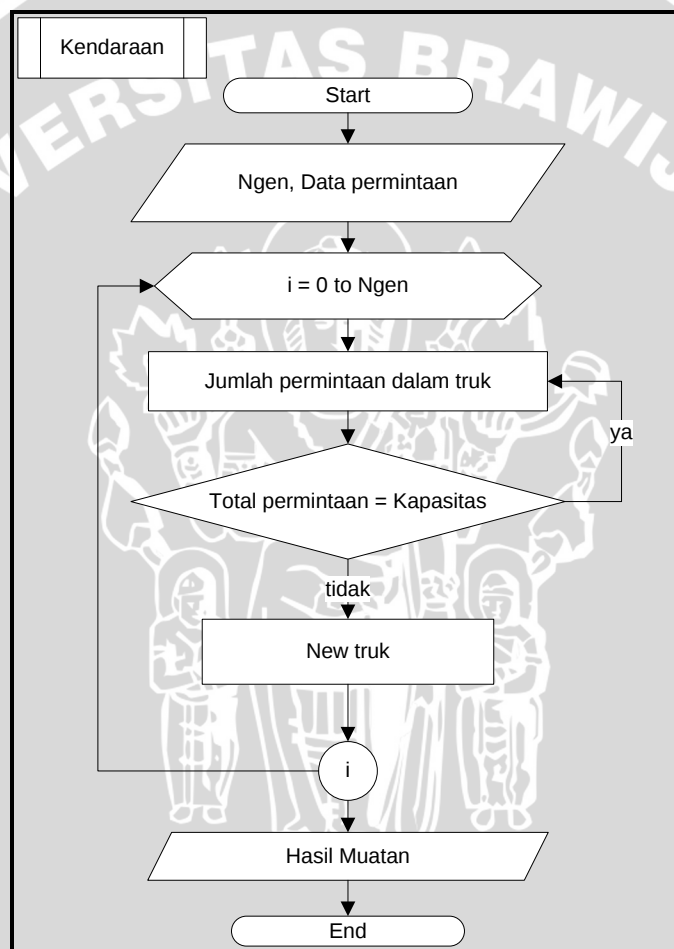
Gambar 3. 5 Flowchart Proses Mutasi

3.2.4 Hitung Nilai *Fitness*

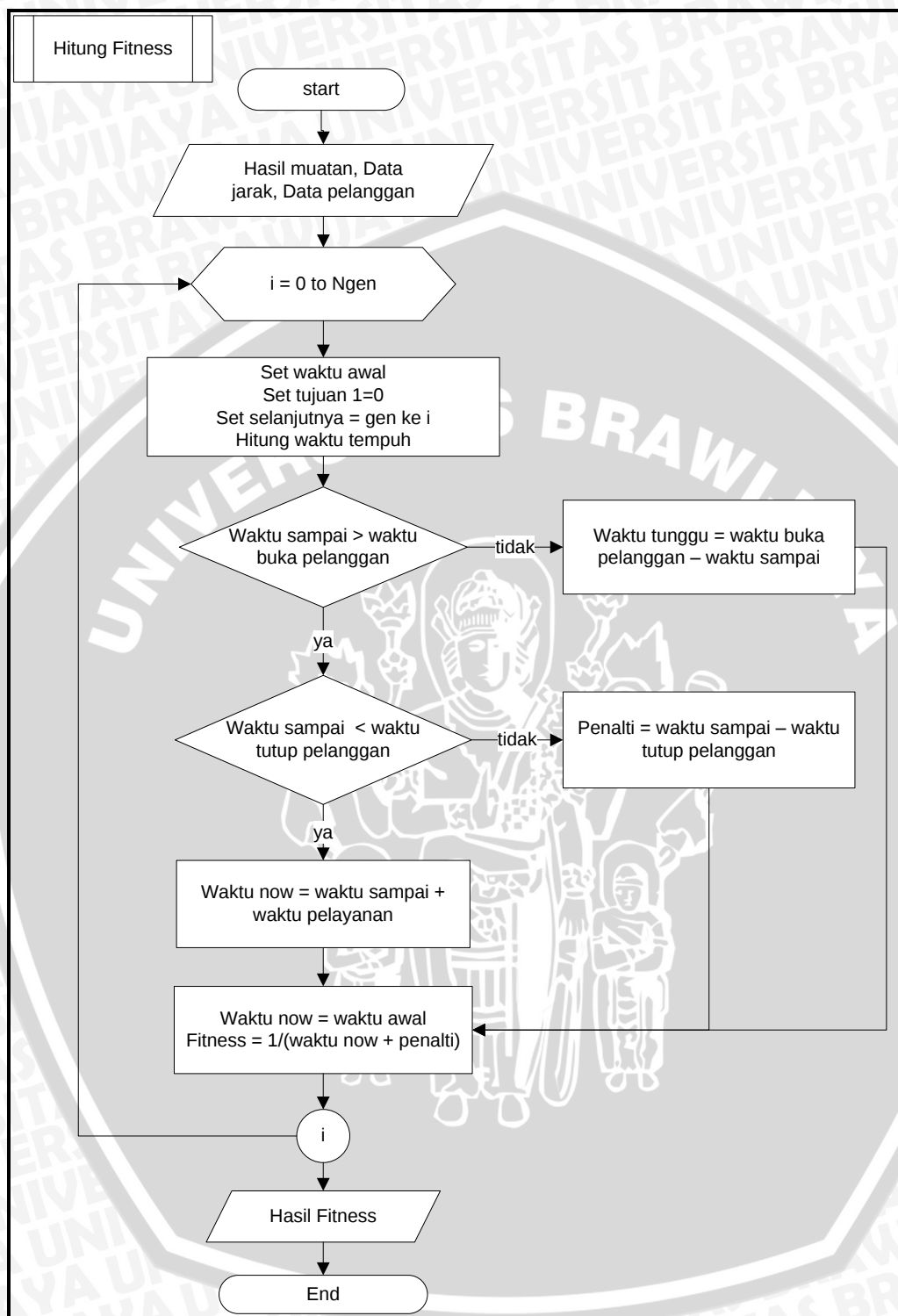
Sebelum melakukan proses seleksi, setiap individu yang terbentuk dalam populasi baru dilakukan perhitungan nilai *fitness*. Nilai *fitness* didapatkan dari persamaan berikut:

$$fitness = \frac{1}{(\alpha Pinalty + \alpha Time)} \quad (3.1)$$

Persamaan diatas merupakan persamaan minimasi VRPTW dimana $\alpha Penalty$ merupakan keterlambatan kendaraan datang yang melewati batas waktu tutup pelanggan, dan $\alpha Time$ adalah waktu perjalanan kesetiap pelanggan. Dari persamaan 3.1 suatu rute dikatakan optimal jika terdapat minimasi terhadap nilai pinalti dan jarak tempuh. Perhitungan nilai *fitness* dilakukan pada setiap gen yang dimuat dalam setiap kendaraan. Gambar 3.6-3.7 merupakan langkah-langkah yang dilakukan dalam perhitungan nilai *fitness*.



Gambar 3. 6 Flowchart Proses Pengisian Muatan



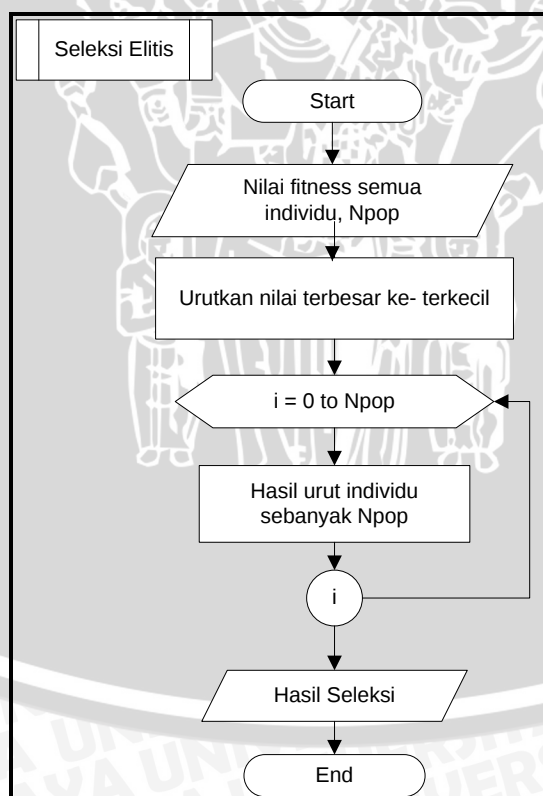
Gambar 3. 7 Proses Perhitungan Nilai Fitness

3.2.5 Proses Seleksi

Dalam algoritma genetik, tahap akhir untuk mendapatkan individu terbaik yaitu menggunakan proses seleksi. Dalam penelitian ini menggunakan dua metode seleksi yaitu seleksi Elitis dan seleksi *Binary Tournament*. Dengan penggunaan dua metode seleksi ini diharapkan dapat melakukan perbandingan nilai terbaik yang dihasilkan. Individu hasil seleksi merupakan individu yang bervariasi dan memiliki nilai *fitness* yang bagus sebagai individu terpilih untuk generasi berikutnya.

3.2.5.1 Elitis

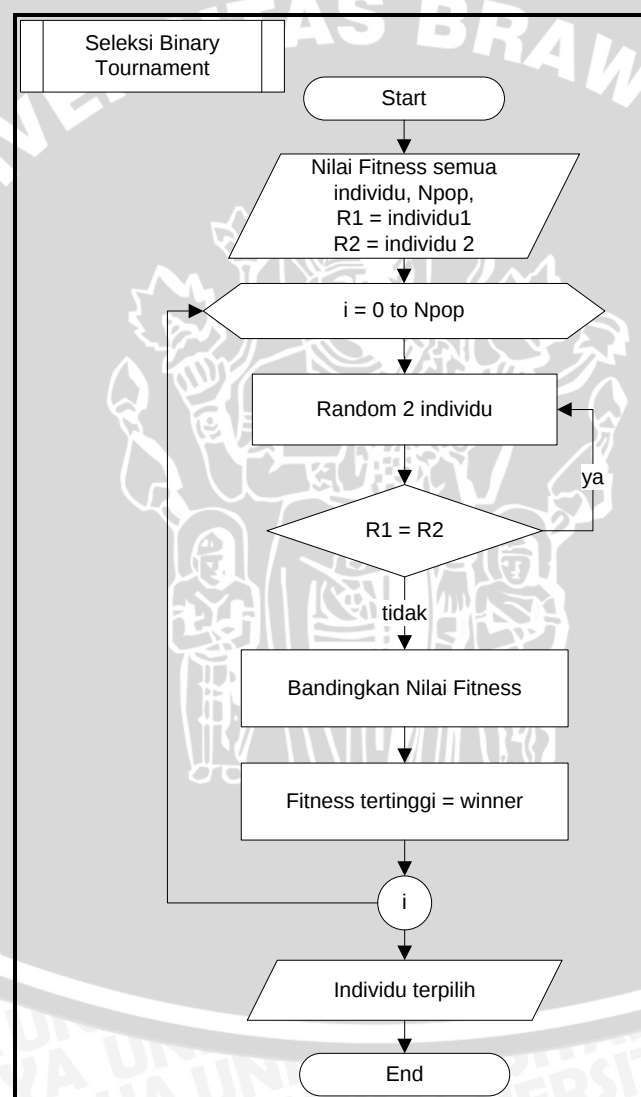
Seleksi elitis dilakukan pada semua individu termasuk *parent* dan *offspring* yang berada dalam satu populasi. Seleksi dilakukan dengan mengurutkan individu dengan nilai *fitness* terbesar sebanyak *pop-size* yang kemudian akan terpilih sebagai individu terbaik generasi berikutnya. Proses seleksi elitis dapat dilihat pada Gambar 3.8 berikut.



Gambar 3.8 Flowchart Proses Seleksi Elitis

3.2.5.2 Binary Tournament

Seleksi dilakukan pada semua individu termasuk *parent* dan *offspring* yang berada dalam suatu populasi. Pada seleksi ini melakukan perbandingan dua individu sebanyak *pop-size* dimana individu pada R_1 dirandom secara acak kemudian dibandingkan dengan individu R_2 . Perbandingan dilakukan untuk mendapatkan individu dengan nilai *fitness* terbesar sebagai pemenang. Individu pemenang terpilih sebagai individu baru yang akan meneruskan generasi berikutnya. Proses seleksi *binary tournament* dapat dilihat pada Gambar 3.9 berikut.



Gambar 3. 9 Flowchart Binary Tournament Selection

3.3 Perhitungan Manual

Dalam penerapan algoritma genetik pada permasalahan VRPTW untuk distribusi beras bersubsidi akan dilakukan contoh perhitungan manual. Misalkan, terdapat 20 pelanggan dengan jumlah permintaan, *time windows*, dan jarak dalam waktu tempuh antara depot ke pelanggan dan pelanggan ke pelanggan lain yang berbeda-beda. Berikut adalah data yang pelanggan pada Tabel 3.1.

Tabel 3. 1 Data Pelanggan

Node	Buka	Tutup	Pelayanan(min)	Permintaan(kg)
1	08.00	13.00	100	180
2	07.00	12.30	94	190
3	09.00	13.30	105	160
4	08.30	14.30	62	160
5	07.30	14.00	75	150
6	13.00	16.00	94	155
7	07.30	14.00	104	190
8	08.00	17.30	113	180
9	08.00	14.00	109	155
10	10.00	15.30	104	180
11	07.00	13.00	120	180
12	08.00	13.00	74	175
13	13.00	17.30	67	190
14	07.00	13.30	93	160
15	09.30	14.30	77	150
16	08.00	14.00	79	160
17	10.00	17.00	99	165
18	07.30	13.00	65	175
19	08.00	14.30	68	185
20	13.00	17.30	65	155
Total Permintaan				3395

Dalam perhitungan manual ini, jumlah populasi (*popSze*) ditetapkan sebagai parameter awal, *popSze* = 10. Panjang kromosom diasumsikan sebagai banyaknya tujuan yang harus dikunjungi dimulai dari keberangkatan kendaraan pukul 07:00. Semua kendaraan memiliki kapasitas yang sama yaitu 900(kg) dengan kecepatan yang diasumsikan konstan 30 km/jam. Pada Tabel 3.2 merupakan data jarak pelanggan.

Tabel 3. 2 Data Jarak Pelanggan

Data Jarak Pelanggan(km)																					
	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s	0	9	8	24	8	23	11	8	13	20	7	26	11	17	27	17	10	20	13	30	15
1	9	0	14	21	20	21	21	16	13	16	29	22	13	21	22	21	9	21	13	7	29
2	8	14	0	12	24	20	24	24	21	14	14	21	13	28	21	26	9	11	13	19	10
3	24	21	12	0	7	15	22	26	18	30	22	15	17	17	20	9	19	23	7	7	21
4	8	20	24	7	0	23	28	13	8	21	30	17	10	11	29	15	28	25	25	20	17
5	23	21	20	15	23	0	16	24	12	27	16	16	7	13	21	10	20	24	30	22	7
6	11	21	24	22	28	16	0	18	23	25	11	10	8	26	21	11	21	23	27	26	8
7	8	16	24	26	13	24	18	0	11	8	13	10	29	24	18	24	28	15	25	30	30
8	13	13	21	18	8	12	23	11	0	8	28	21	28	30	13	25	17	28	28	28	18
9	20	16	14	30	21	27	25	8	8	0	20	29	23	21	23	9	20	29	20	13	25
10	7	29	14	22	30	16	11	13	28	20	0	27	11	12	26	11	18	16	21	12	19
11	26	22	21	15	17	16	10	10	21	29	27	0	7	28	20	19	24	26	14	12	7
12	11	13	13	17	10	7	8	29	28	23	11	7	0	8	27	30	13	21	8	20	29
13	17	21	28	17	11	13	26	24	30	21	12	28	8	0	11	11	27	24	23	22	13
14	27	2	21	20	29	21	21	18	13	23	26	20	27	11	0	20	12	16	12	15	29
15	17	21	26	9	15	10	11	24	25	9	11	19	30	11	20	0	17	11	10	15	8
16	10	9	9	19	28	20	21	28	17	20	18	24	13	27	12	17	0	22	13	19	29
17	20	21	11	23	25	24	23	15	28	29	16	26	21	24	16	11	22	0	8	17	10
18	13	13	13	7	25	30	27	25	28	20	21	14	8	23	12	10	13	8	0	28	13
19	30	7	19	7	20	22	26	30	28	13	12	12	20	22	15	15	19	17	28	0	26
20	15	29	10	21	17	7	8	30	18	25	19	7	29	13	8	8	29	10	13	26	0

3.3.1 Membuat Populasi Awal

Tahap awal yang dilakukan dalam membuat populasi awal adalah membangkitkan individu secara random dari 1-20 sebanyak jumlah populasi yang ditentukan. Representasi yang digunakan adalah representasi permutasi dimana kromosom yang dibangkitkan berupa rute tujuan pelanggan dalam pengantaran beras distribusi. Contoh populasi awal yang dibangkitkan dapat dilihat pada Tabel 3.3 berikut.

Tabel 3. 3 Populasi Awal

Parent	Kromosom																			
P1	1	12	19	5	2	11	10	9	3	8	15	17	20	13	7	6	18	4	14	16
P2	15	7	11	2	17	8	10	16	12	3	14	18	9	5	1	4	19	13	20	6
P3	7	19	4	9	14	2	6	18	17	10	8	13	16	20	11	12	3	1	5	15
P4	14	10	9	5	11	19	3	7	1	13	18	20	17	12	16	8	4	15	6	2
P5	12	6	7	18	8	10	1	4	20	16	2	19	9	15	13	3	11	14	17	5
P6	17	4	2	11	19	12	15	13	10	3	1	9	8	5	20	6	14	18	7	16
P7	18	6	3	10	2	5	11	20	14	4	15	7	9	19	12	17	1	16	8	13
P8	6	8	5	11	20	2	18	3	10	9	13	15	17	12	4	16	1	7	14	19
P9	10	3	13	7	4	15	16	9	2	19	14	20	1	5	8	17	6	18	11	12
P10	19	9	7	12	11	2	18	8	6	16	13	17	10	14	20	4	15	1	5	3

3.3.2 Menghitung Nilai *Fitness*

Nilai *fitness* yang digunakan dalam permasalahan VRPTW pada algoritma genetik adalah optimasi rute dengan meminimalkan waktu tunggu dan total jarak tempuh, maka persamaan yang digunakan adalah persamaan (3.1). Dalam perhitungan nilai *fitness* terdapat total jarak (km) yang diakumulasikan dalam satuan waktu (menit) dengan persamaan berikut :

$$\text{waktu tempuh} = \frac{\text{jarak(km)}}{30} * 60 \quad (3.2)$$

Dari persamaan 3.2 menyatakan bahwa kecepatan rata-rata kendaraan adalah 30 km/jam. Pembagian kendaraan dalam kromosom dilakukan dengan cara meletakkan permintaan setiap pelanggan (gen) secara berurutan. Jumlah permintaan yang dimasukkan kedalam truk dihitung secara konstan sesuai kapasitas truk yaitu 900kg. Jika muatan pada truk satu sudah mencapai kapasitas maksimum, maka gen selanjutnya akan berpindah truk. Pada Gambar 3.10 merupakan contoh peletakan permintaan pelanggan kesetiap truk untuk P1.

P1	1	12	19	5	2	11	10	9	3	8	15	17	20	13	7	6	18	4	14	16
Pr	180	175	185	150	190	180	180	155	160	180	150	165	155	190	190	155	175	160	160	160
TP	880					855					850					810				
Tr	1					2					3					4				

Gambar 3. 10 Proses Pembagian Gen dalam Truk

Keterangan istilah pada Tabel 3.4 :

P1 : kromosom yang digunakan sebagai contoh perhitungan

Pr : Permintaan setiap pelanggan

TP : Total permintaan yang disesuaikan dengan kapasitas truk

Tr : Truk yang akan digunakan.

Setelah dilakukan pembagian truk seperti pada Tabel 3.4, maka masing-masing truk melakukan perjalanan ke pelanggan yang sudah ditentukan. Dalam proses tersebut setiap truk memiliki waktu tempuh dan *time windows* yang berbeda-beda. Truk akan mendapat pinalti jika mengalami keterlambatan apabila waktu

sampai kendaraan lebih dari waktu tutup pelanggan yaitu dalam arti pelanggan sudah tutup. Pada kondisi tersebut maka terdapat jumlah truk yang melewati pelanggan dengan layanan = 0. Contoh perhitungan ditampilkan pada Tabel 3.4 – Tabel 3.7 berikut.

Tabel 3. 4 Data Tujuan Truk 1

Truk 1									
Node	Waktu(min)	Sampai	Buka	Tunggu	Mulai	Layanan	Selesai	Tutup	Pinalti
1	18	07.18	08.00	42	08.00	100	09.40	13.00	0
12	26	10.06	08.00	0	10.06	74	11.20	13.00	0
19	40	12.00	08.00	0	12.00	68	13.08	14.30	0
5	44	13.52	07.30	0	13.52	75	15.07	14.00	0
2	40	15.51	12.30	0	0	0	15.51	12.30	201
Total							531		201

Tabel 3. 5 Data Tujuan Truk 2

Truk 2									
Node	Waktu(min)	Sampai	Buka	Tunggu	Mulai	Layanan	Selesai	Tutup	Pinalti
11	52	07.52	07.00	0	07.52	120	09.52	13.00	0
10	54	10.46	10.00	0	10.46	104	12.30	15.30	0
9	40	13.10	08.00	0	13.10	109	14.59	14.00	0
3	60	15.59	09.00	0	0	0	15.59	13.30	149
8	36	16.35	08.00	0	16.35	113	18.28	17.30	0
Total							688		149

Tabel 3. 6 Data Tujuan Truk

Truk 3									
Node	Waktu(min)	Sampai	Buka	Tunggu	Mulai	Layanan	Selesai	Tutup	Pinalti
15	34	07.34	09.30	64	09.30	77	10.47	14.30	0
17	22	11.09	10.00	0	11.09	99	12.48	17.00	0
20	20	13.08	13.00	0	13.08	65	14.13	17.30	0
13	26	14.39	13.00	0	14.39	67	15.46	17.30	0
7	48	16.34	07.30	0	0	0	16.34	14.00	154
Total							574		154

Tabel 3. 7 Data Tujuan Truk 4

Truk 4									
Node	Waktu(min)	Sampai	Buka	Tunggu	Mulai	Layanan	Selesai	Tutup	Pinalti
6	22	07.22	13.00	338	13.00	95	14.35	16.00	0
18	54	15.29	07.30	0	0	0	15.29	13.00	149
4	50	16.19	08.30	0	0	0	16.19	14.30	131
14	58	17.17	07.00	0	0	0	17.17	13.30	253
16	24	17.41	08.00	0	0	0	17.41	14.00	221
Total							641		754

Perhitungan pada Tabel 3.4-3.7 dilakukan untuk semua individu dengan menambahkan permintaan secara berurutan sesuai kapasitas kendaraan hingga tujuan terakhir. Setelah hasil pinalti dan hasil waktu perjalanan diketahui, maka tahap selanjutnya adalah menghitung nilai *fitness* setiap truk dengan persamaan 3.1 berikut :

$$fitness = \frac{1}{(\alpha Pinalty + \alpha Time)} \quad (3.1)$$

Nilai *fitness* didapat dari perhitungan setiap truk yang digunakan pada satu individu. Berikut merupakan contoh perhitungan nilai *fitness* untuk kromosom P1 pada Tabel 3.8.

Tabel 3. 8 Perhitungan Nilai Fitness

Kendaraan	Pinalti	Time	Fitness
Truk 1	201	531	$\frac{1}{201+531} = 0.00137$
Truk 2	149	688	$\frac{1}{149+688} = 0.00119$
Truk 3	154	574	$\frac{1}{154+574} = 0.00137$
Truk 4	754	641	$\frac{1}{754+641} = 0.00071$
Fitness P1			0.004651

Karena semua kendaraan tersebut terdapat dalam satu individu yaitu P1, maka semua hasil *fitness* yang didapat dari setiap truk di jumlahkan, sehingga nilai

fitness untuk P1 = 0.004651. Proses perhitungan sesuai alur pada Tabel 3.4 – 3.9 dilakukan sebanyak individu dalam satu populasi untuk mendapatkan nilai *fitness* masing-masing individu. Berikut merupakan nilai *fitness* yang didapat dari setiap individu pada Tabel 3.9.

Tabel 3. 9 Hasil Fitness Setiap Individu

Individu	Fitness
P1	0.00465134
P2	0.00603284
P3	0.00518187
P4	0.01119146
P5	0.00518991
P6	0.00500154
P7	0.00476289
P8	0.00511106
P9	0.00421867
P10	0.00561001
P11	0.0057257
P12	0.00560248
P13	0.00482878
P14	0.00515024

3.3.3 Crossover

Proses reproduksi yang pertama dilakukan adalah *crossover* dengan menggunakan metode PMX untuk membangkitkan keturunan baru (*offspring*) dari hasil persilangan antara dua *parent* yang diambil secara acak. Tingkat *crossover* (*crossover rate/pc*) merupakan jumlah kromosom dalam suatu populasi yang diharapkan akan mengalami *crossover* hingga menghasilkan keturunan baru yang memiliki perbedaan sifat dari induknya seperti pada proses berikut. Terdapat P7 dan P10 yang merupakan individu terpilih sebagai *parent*. Kemudian menentukan dua titik potong secara acak sebagai area pemetaan. Titik potong P7 terdapat pada titik 6 dan 16. Titik potong P10 terdapat pada titik 7 dan 20. Titik potong yang sudah terpilih dari dua *parent* tersebut merupakan daerah mapping. Setelah itu lakukan pertukaran area pemetaan pada *parent1* dan *parent2*. Jika terdapat nilai yang sama dalam satu individu maka nilai yang harus diganti adalah nilai yang berada diluar area pemetaan. Nilai diganti sesuai dengan ketentuan pemetaan.

Proses *crossover* PMX ini memiliki nilai $Cr = 0.2$,hanya melakukan satu kali *crossover* karena $Child = 10 * 0.2 = 2$, maka keturunan yang dihasilkan adalah C1 dan C2 seperti pada Gambar 3.11 berikut.

P7	18	6	3	10	2	5	11	20	14	4	15	7	9	19	12	17	1	16	8	13
P10	19	9	7	12	11	2	18	8	6	16	13	17	10	14	20	4	15	1	5	3
Proto Child																				
P7	18	6	3	10	2	18	11	20	14	4	15	7	9	19	12	3	1	16	8	13
P10	19	9	7	12	11	2	5	8	6	16	13	17	10	14	20	4	15	1	5	17
Mapping																				
18		3																		
5		17																		
Child																				
P11	5	6	17	10	2	18	11	20	14	4	15	7	9	19	12	3	1	16	8	13
P12	19	9	7	12	11	2	5	8	6	16	13	3	10	14	20	4	15	1	18	17

Gambar 3. 11 Hasil Crossover PMX

3.3.4 Mutasi

Proses reproduksi selanjutnya adalah mutasi. Pada operator mutasi tidak perlu menggunakan dua *parent* untuk menghasilkan keturunan. Mutasi dilakukan oleh satu individu sebagai induk yang dipilih secara acak kemudian dilakukan penukaran gen dalam individu tersebut hingga menghasilkan susunan gen baru pada individu baru hasil mutasi. Pada proses mutasi ini gen yang akan ditukar sebanyak 4 titik yang diambil secara acak. $Mr = 0.2$, maka dilakukan proses mutasi pada dua individu. Individu yang terpilih sebagai induk adalah P8 dengan penukaran gen pada titik-5, ditukar dengan den pada titik-11 dan gen pada titik-14 ditukar dengan gen pada titik-19. Kemudian P2 dengan penukaran gen pada titik-5 ditukar dengan den pada titik-11 dan gen pada titik-15 ditukar dengan gen pada titik-20. Hal ini dilakukan agar anak yang dihasilkan tidak mengalami banyak kemiripan dengan induk. Contoh proses mutasi dapat dilihat pada Gambar 3.12 berikut.

P8	6	8	5	11	20	2	18	3	10	9	13	15	17	12	4	16	1	7	14	19
C3	6	8	5	11	13	2	18	3	10	9	20	15	17	14	4	16	1	7	12	19

P2	15	7	11	2	17	8	10	16	12	3	14	18	9	5	1	4	19	13	20	6
C4	15	7	11	2	14	8	10	16	12	3	17	18	9	5	6	4	19	13	20	1

Gambar 3. 12 Hasil Mutasi

3.3.5 Evaluasi

Evaluasi dilakukan untuk mengetahui hasil *fitness* yang didapat dari semua *parent* dan anak yang berada dalam satu populasi baru. Populasi baru dihasilkan dari populasi awal ditambah dengan individu baru hasil reproduksi (*crossover* dan *mutasi*). Nilai *fitness* terbaik dinyatakan dari besarnya nilai *fitness* yang didapat. Semakin besar nilai *fitness* maka semakin baik sifat yang dimiliki individu untuk calon generasi berikutnya. Hasil perhitungan *fitness* yang didapat dari populasi baru dapat dilihat pada Tabel 3.10 berikut.

Tabel 3. 10 Hasil Fitness Populasi Baru

Individu	Jumlah Truk	Terlewat	Fitness
P1	4	7	0.00465134
P2	4	6	0.00603284
P3	4	7	0.00518187
P4	5	5	0.01119146
P5	4	7	0.00518991
P6	4	9	0.00500154
P7	4	8	0.00476289
P8	4	8	0.00511106
P9	4	9	0.00421867
P10	4	8	0.00561001
P11	4	8	0.0057257
P12	4	7	0.00560248
P13	4	8	0.00482878
P14	4	8	0.00515024

3.3.6 Seleksi

Proses seleksi adalah proses pemilihan individu terbaik sebagai calon generasi selanjutnya. Proses seleksi dilakukan menggunakan dua metode seleksi yaitu *elitis* dan *binary tournament*. Dengan menggunakan dua metode seleksi maka dapat diketahui solusi terbaik yang dihasilkan dari perbandingan hasil antara dua metode tersebut.

3.3.6.1 Elitis

Pada proses seleksi elitis, fungsi *fitness* digunakan sebagai parameter untuk menentukan tingkat kesempurnaan suatu kromosom yang akan bertahan pada generasi selanjutnya. Nilai *fitness* terbaik diambil dari populasi baru dengan hasil

reproduksi sebanyak *pop-size* awal. Dalam penelitian ini, *pop-size* awal = 10, maka terdapat 10 individu dengan nilai *fitness* terbaik. Untuk mengetahui 10 individu terbaik maka dilakukan sorting dari nilai terbesar terlebih dahulu, seperti ditunjukkan pada Tabel 3.11 berikut.

Tabel 3. 11 Proses Seleksi Elitis

Individu	Fitness
P4	0.011191461
P2	0.00603284
P11	0.0057257
P10	0.00561001
P12	0.005602476
P5	0.005189912
P3	0.005181866
P14	0.005150243
P8	0.005111063
P6	0.005001536
P13	0.004828776
P7	0.004762886
P1	0.004651336
P9	0.004218667

Kemudian pilih 10 individu sesuai *pop-size* dengan nilai *fitness* terbesar sebagai calon generasi selanjutnya. Berikut adalah hasil individu yang terpilih dari seleksi elitis ditunjukkan pada Tabel 3.12.

Tabel 3. 12 Hasil Seleksi Elitis

P	Fitness
P4	0.011191
P2	0.006033
P11	0.005726
P10	0.00561
P12	0.005602
P5	0.00519
P3	0.005182
P14	0.00515
P8	0.005111
P6	0.005002

3.3.6.2 Binary Tournament Selection

Dalam proses seleksi *binary tournament* dilakukan perbandingan antara individu sebanyak *tournament size*. Pada penelitian ini $T_s = 2$, maka banyaknya

individu yang melakukan turnamen adalah 2 individu. Individu pertama sebagai R_1 dan individu kedua sebagai R_2 . Pemilihan individu untuk melakukan turnamen dipilih secara acak sebanyak populasi awal *pop-size*. Individu yang terpilih dalam R_1 tidak boleh sama dengan individu dalam R_2 , jika muncul nilai yang sama, maka dilakukan random kembali hingga kondisi tidak sama seperti contoh pada Tabel 3.13 berikut.

Tabel 3. 13 Proses Pemilihan Individu

R1	R2	Kondisi	R2
9	6	tdk sama	
13	13	sama	10
5	12	tdk sama	
5	9	tdk sama	
11	2	tdk sama	
3	7	tdk sama	
7	4	tdk sama	
5	11	tdk sama	
8	3	tdk sama	
1	1	sama	4

Sesuai tabel 3.14 jika terdapat kondisi yang sama maka dilakukan pemilihan ulang individu secara acak pada R_2 . Individu R_1 dan R_2 akan bersaing dengan nilai *fitness* masing-masing. Pemenang dari turnamen ini adalah individu dengan nilai *fitness* terbesar yang biasa disebut sebagai *winner*. Individu yang terpilih sebagai *winner* merupakan individu yang akan melakukan reproduksi lagi sebanyak generasi yang diinginkan. Pada Tabel 3.14 adalah hasil turnamen dengan nilai *fitness* terbesar.

Tabel 3. 14 Individu Yang Terpilih Sebagai Winner

Fitness R1	Fitness R2	Winner
0.004219	0.005002	P6
0.005008	0.00561	P10
0.00519	0.00561	P12
0.00519	0.004219	P5
0.00524	0.006033	P2
0.005182	0.004763	P3
0.004763	0.011191	P7
0.00519	0.00524	P11
0.005111	0.005182	P3
0.004651	0.011191	P1

3.3.7 Memilih Kromosom Terbaik

Dalam setiap metode seleksi akan didapatkan individu terbaik sebanyak *pop-size*. Dari hasil seleksi tersebut maka akan dipilih satu kromosom terbaik dengan kriteria yaitu dimana suatu individu tersebut memiliki nilai *fitness* yang tinggi dengan sedikitnya jumlah pelanggan yang terlewati. Pada Tabel 3.15 berikut merupakan hasil kromosom terbaik yang terpilih.

Tabel 3. 15 Hasil Kromosom Terbaik

Individu	Kromosom																			
P4	15	7	11	2	17	18	10	16	12	3	14	18	9	5	1	4	19	13	20	6
Fitness	0.01119146																			
Terlewati	5																			

3.4 Perancangan User Interface

Perancangan *user Interface* dari perangkat lunak untuk permasalahan optimasi distribusi beras bersubsidi terdiri dari tiga halaman. Halaman pertama berisi *input* data parameter GA yang ingin diproses seperti yang ditunjukkan pada Gambar 3.13 berikut.

Gambar 3. 13 Halaman Utama Perancangan User Interface

Keterangan :

1. Input banyak populasi yang diinginkan
2. Input banyak generasi yang diinginkan
3. Input nilai *crossover rate*

4. Input nilai *mutation rate*
5. Pilih seleksi yang akan digunakan
6. Tombol untuk memilih data yang akan diproses
7. Tombol untuk memproses semua data yang sudah di masukkan
8. *TextBox* untuk menampilkan semua hasil dari “proses”.
9. Kolom untuk menampilkan kromosom terbaik
10. Kolom untuk menampilkan nilai *fitness* dari kromosom terbaik
11. Kolom untuk menampilkan berapa banyak pelanggan yang terlewat.

Halaman kedua pada perancangan *user interface* merupakan halaman yang menampilkan data jarak pelanggan yang akan diproses seperti pada Gambar 3.14 berikut.

Title 1	Title 2	Title 3	Title 4
1			

Gambar 3. 14 Tampilan Data Jarak

Keterangan :

1. Tabel untuk menampilkan data jarak kesetiap pelanggan
2. Kolom untuk menampilkan *map* lokasi setiap pelanggan

Setelah dilakukan “proses” maka semua hasil akan diketahui dan kromosom terbaik telah terpilih kemudian klik kolom 8, maka akan muncul tampilan seperti Gambar 3.15 sebagai berikut.

The screenshot shows a software window with a title bar. Inside, there's a table with two columns: 'Truk' and 'Kromosom'. To the right of the table is a large empty box labeled 'Perhitungan Manual'. Above the table, there's a label 'Krom =' followed by a text input field. Red annotations are present: '1' points to the input field, '2' points to the table, and '13' points to the 'Perhitungan Manual' box.

Truk	Kromosom

Gambar 3. 15 Detail Kromosom Terbaik

Keterangan :

1. Menampilkan kromosom terbaik yang dihasilkan
2. Menampilkan detail truk dengan tujuannya (kromosom)
3. Menampilkan perhitungan manual dari setiap truk

3.5 Perancangan Uji Coba dan Evaluasi

Proses pengujian merupakan evaluasi perangkat lunak yang akan diuji dengan berbagai macam parameter GA. Pengujian bertujuan untuk mengetahui pengaruh GA terhadap hasil optimasi rute distribusi beras bersubsidi. Hasil yang didapatkan berupa individu dengan nilai *fitness* terbaik. Uji coba dilakukan dalam 4 skenario, dimana setiap skenario akan dilakukan pengujian sebanyak 10 kali lalu diambil rata-rata nilai *fitness* terbaik dari setiap skenario uji. Pengujian tersebut antara lain adalah:

1. Uji coba untuk menentukan banyaknya generasi yang optimal untuk proses algoritma genetika pada optimasi distribusi beras bersubsidi.
2. Uji coba untuk menentukan banyaknya populasi yang optimal untuk proses algoritma genetika pada optimasi distribusi beras bersubsidi.
3. Uji coba untuk mencari kombinasi *mutation rate* dan *crossover rate* yang terbaik untuk menyelesaikan permasalahan VRPTW untuk optimasi distribusi beras bersubsidi.
4. Uji coba untuk mengetahui seleksi mana yang terbaik untuk permasalahan VRPTW untuk optimasi distribusi beras bersubsidi.

3.5.1 Uji Coba Banyaknya Generasi

Uji coba banyaknya generasi terhadap program dilakukan untuk mengetahui seberapa banyak generasi yang optimal pada permasalahan VRPTW. Uji coba akan dilakukan sebanyak 10 kali dengan generasi kelipatan 500 yang dimulai dari 500-3500 generasi. Semakin banyak jumlah generasi, maka semakin meningkat kemampuan algoritma genetika dalam mencari solusi terbaik [MAH-13]. Jumlah populasi yang digunakan sebanyak 60 populasi dengan Cr dan Mr 0.5 pada metode seleksi elitis. Tabel uji coba banyaknya generasi dapat dilihat pada Tabel 3.16 berikut.

Tabel 3. 16 Skenario Uji Coba Banyak Generasi

Generasi	Percobaan										Rata-rata Fitness
	1	2	3	4	5	6	7	8	9	10	
500											
1000											
1500											
2000											
2500											
3000											

3.5.2 Uji Coba Banyaknya Populasi

Uji coba banyaknya populasi pada program dilakukan untuk mengetahui banyaknya populasi dengan solusi optimasi terbaik pada permasalahan VRPTW. Semakin banyak jumlah populasi maka semakin berpengaruh terhadap peningkatan kemampuan algoritma genetika untuk menemukan solusi terbaik [MAH-13]. Uji coba banyaknya populasi dilakukan pada generasi terbaik. Skenario uji coba dilakukan sebanyak 1500 generasi dengan nilai Cr dan Mr 0.5 dengan metode seleksi elitis. Banyaknya populasi adalah kelipatan 20 yaitu dari 20 populasi hingga 120 populasi. Tabel uji coba banyaknya populasi dapat dilihat pada Tabel 3.17 berikut.

Tabel 3. 17 Skenario Uji Coba Banyaknya Populasi

Banyak Populasi	Nilai Fitnes										Rata-rata <i>Fitness</i>
	Percobaan banyaknya populasi										
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											
120											

3.5.3 Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Pengujian ini dilakukan setelah menemukan jumlah populasi pada generasi beberapa yang menghasilkan individu terbaik. Metode tersebut akan dilakukan pada pengujian untuk mengukur kombinasi *crossover rate* dan *mutation rate* yang paling tepat untuk menyelesaikan permasalahan VRPTW. Dalam pengujian ini, *crossover rate* dan *mutation rate* akan memiliki nilai yang berbeda pada parameter genetik. Nilai *crossover rate* dan *mutation rate* yang akan diujikan berskala 0 hingga 1. Pada pengujian ini, jumlah populasi yang digunakan adalah 60 populasi dengan jumlah generasi 1500 pada metode seleksi elitis, masing-masing kombinasi akan dilakukan uji coba sebanyak 10 kali agar mendapatkan solusi terbaik pada setiap percobaan. Tabel skenario uji coba *crossover rate* dan *mutation rate* dapat dilihat pada Tabel 3.18 berikut

Tabel 3. 18 Skenario Uji Coba Nilai Kombinasi Cr dan Mr

Kombinasi		Nilai fitness										Rata-rata Fitness
		Percobaan										
Cr	Mr	1	2	3	4	5	6	7	8	9	10	
1	0											
0.9	0.1											
0.8	0.2											
0.7	0.3											
0.6	0.4											
0.5	0.5											
0.4	0.6											
0.3	0.7											
0.2	0.8											
0.1	0.9											
0	1											

3.5.4 Uji Coba Perbandingan Seleksi Menggunakan Metode Elitis dan *Binary Tournament*

Pengujian seleksi dilakukan untuk membandingkan metode seleksi mana yang lebih optimal untuk optimasi rute distribusi beras bersubsidi dengan algoritma genetika. Pengujian setiap metode seleksi dilakukan dengan skenario percobaan seperti berikut :

- 1) Menguji program dengan metode seleksi Elitis dengan jumlah generasi sebanyak 1500 generasi, 60 populasi serta nilai Cr dan Mr adalah 0:5 0:5. Pengujian dilakukan sebanyak 10 kali percobaan.
- 2) Menguji program dengan metode seleksi *Binary Tournament* dengan jumlah generasi sebanyak 1500 generasi, 60 populasi serta nilai Cr dan Mr adalah 0:5 0:5. Pengujian dilakukan sebanyak 10 kali percobaan.

Tabel perancangan uji coba dua metode seleksi yaitu elitis dan *binary tournament* dapat dilihat pada Tabel 3.19.

Tabel 3. 19 Skenario Percobaan Metode Seleksi

Metode	Nilai Fitnes Percobaan									
	1	2	3	4	5	6	7	8	9	10
Elitis										
<i>Binary</i>										

BAB IV

IMPLEMENTASI

Pada bab implementasi ini merupakan tahap penerapan sistem, dimana aplikasi siap dijalankan pada keadaan sebenarnya. Penerapan sistem yang akan dijelaskan terdiri dari lingkungan implementasi, implementasi aplikasi dan implementasi tampilan *user interface*.

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan ruang lingkup kebutuhan dalam penerapan sistem Algoritma Genetik pada VRPTW untuk distribusi beras bersubsidi. Terdapat dua lingkungan yang digunakan dalam membangun sistem aplikasi tersebut, yaitu lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi aplikasi algoritma genetik untuk beras bersubsidi adalah sebagai berikut :

1. Prosesor Intel(R) Core(TM) i5-2430M 2,4 GHz.
2. *Memory* (RAM) 4 GB.
3. *Harddisk* dengan kapasitas 500 GB.
4. *Monitor* 14 inch.

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi aplikasi algoritma genetik untuk beras bersubsidi adalah sebagai berikut :

1. Sistem operasi yang digunakan adalah Windows8 Pro 64 bit.
2. *Tools* yang digunakan adalah JDK1.7 dan JRE7.
3. *Editor* yang digunakan adalah Netbeans 8.0.

4.2 Implementasi Program

Berdasarkan metodologi dan perancangan sistem yang telah dijelaskan pada Bab 3, maka selanjutnya akan dijelaskan implementasi proses-proses tersebut kedalam bentuk sistem dengan bahasa pemrograman Java.

4.2.1 Struktur Data

Struktur data yang digunakan dalam implementasi Optimasi Didtribusi Beras Bersubsidi pada Algoritma Genetika dapat dilihat pada *Source code* 4.1 berikut.

```
String [][] dataJarak ;  
String [][] dataPermintaan;  
int [][][] Generasi;  
ArrayList[]Bagitruk;  
File image;
```

Source code 4. 1 Struktur Data

Keterangan :

1. String [][] dataJarak = menyimpan data jarak setiap pelanggan berupa array dua dimensi dengan tipe data String.
2. String [][] dataPermintaan = menyimpan data kebutuhan setiap pelanggan berupa array dua dimensi dengan tipe data String.
3. int [][][] Generasi = menyimpan hasil proses pada setiap generasi berupa array tiga dimensi dengan tipe data Integer.
4. ArrayList<String>[]Bagitruk = menyimpan pembagian gen untuk setiap truk berupa array satu dimensi dengan tipe data ArrayList<String>.
5. File image = Menyimpan gambar map dengan tipe data File.

4.2.2 Membangkitkan Individu Awal

Membangkitkan populasi awal dimulai dengan membangkitkan individu. Suatu individu diperoleh dengan cara *random* sebanyak panjang kromosom yaitu 20. Proses membangkitkan individu dilakukan pengecekan pada setiap gen agar tidak terdapat nilai yang sama dalam satu kromosom. Pada *Source code* 4.2 berikut merupakan proses membangkitkan populasi awal.

```

public int [] Individu (int Ndata){
    Random Irandom = new Random();
    int [] krom = new int[Ndata];
    for (int i =1;i<=Ndata;i++)
    {
        int rand1 = Irandom.nextInt(Ndata-1)+1;
        while (cek(krom, rand1)== true){
            rand1 = Irandom.nextInt(Ndata)+1;
        }
        krom[i-1]=rand1;
    }
    return krom;
}

```

Source code 4. 2 Membangkitkan Populasi Awal

4.2.3 Pembagian Kapasitas Kendaraan

Pembagian kapasitas kendaraan dilakukan pada setiap kromosom. Kendaraan akan mengunjungi beberapa pelanggan sesuai dengan pembagian permintaan pada setiap pelanggan. Jumlah permintaan pelanggan disesuaikan dengan kapasitas kendaraan. Maka dalam satu kromosom akan terdapat beberapa kendaraan yang mengunjungi setiap gen sebagai tujuan. Pada *Source code 4.3* merupakan proses pembagian kapasitas kendaraan.

```

public ArrayList [] Truk (int [][] pop, String [][] jarak, String
    [][]permintaan){
    ArrayList []a=new ArrayList [pop.length];
    ArrayList<String> r = new ArrayList<>();
    String krom = "";
    for (int i = 0; i < pop.length; i++) {
        int totalP = 0;
        int jmlTruk = 0;
        int jmlgen=0;
        for (int j = 0; j < pop[i].length; j++) {
            totalP+=Integer.parseInt(permintaan[pop[i][j]-1][4]);
            if(jmlgen<pop[i].length-1){
                if(totalP>=900){
                    totalP=Integer.parseInt(permintaan[pop[i][j]-1][4]);
                    jmlTruk++;
                    r.add(krom);
                    krom=pop[i][j]+" ";
                }
                else{
                    krom=krom+pop[i][j]+" ";
                }
            }
        }
        else{
            if(totalP<900){
                krom=krom+pop[i][j];
                r.add(krom);
            }
            else{
                r.add(krom);
                krom=pop[i][j]+"";
                r.add(krom);
            }
        }
    }
}

```

```

    }
    jmlgen++;
}
krom="";
a[i]=r;
r=new ArrayList<>();
}
return a;

```

Source code 4. 3 Pembagian Kapasitas Kendaraan

4.2.4 Perhitungan Nilai Fitness

Perhitungan nilai *fitness* diperoleh dari total waktu tempuh kesetiap tujuan yang dimulai dari depot hingga tujuan akhir ditambah dengan total pinalti yang didapat pada setiap pelanggan. Perhitungan nilai *fitness* dilakukan pada setiap truk yang bekerja dalam suatu kromosom. Proses perhitungan *fitness* dapat dilihat pada Source code 4.4 berikut.

```

public String [] Fitness (String Krom, String [][]jarak, String [][]
permintaan){
    String [] Fitnes =new String[3];
    double Wpelayanan;
    double taw;
    double ta;
    double tn=07.00;
    int prev=0;
    int next;
    int error = 0 ;
    double penalti;
    double wtempuh;
    double fitness=0;
    String kromosom[] = Krom.split(" ");
    for(int i = 0 ; i < kromosom.length;i++){
        next= Integer.parseInt(kromosom[i] + "");
        double dttujuan[]=Tujuan(next, permintaan);
        taw=dtujuan[0];
        ta=dtujuan[1];
        if(i!=0){
            prev=Integer.parseInt(kromosom[i-1]+"");
        }
        wtempuh = jrkTempuh(prev, next, jarak);
        tn = (conJam(tn-7)) + wtempuh;
        tn=Double.valueOf(twoDForm.format(conMenit(tn)+7));
        Wpelayanan = dtService(next, permintaan);
        penalti = getPinalti(tn, taw, ta);
        //tepat waktu
        if(penalti==0){
            tn=(conJam(tn-7))+Wpelayanan;
        }
        //terlalu cepat
        else if (penalti<0){
            tn=(conJam(tn-7))+Wpelayanan+(penalti*-1);
        }
        //terlambat
        else if(penalti>0){
            tn=(conJam(tn-7));
            fitness+=(int) (penalti);
        }
    }
}

```

```

        error++;
    }
    tn=Double.valueOf(twoDForm.format(conMenit(tn)+7));
    fitness = conJam(tn-7)+penalti;
}
fitness= 1/fitness;
Fitnes[0]=Krom;
Fitnes[1]=Double.valueOf(sixDForm.format(fitness))+"";
Fitnes[2]=error+"";
return Fitnes;}

```

Source code 4. 4 Perhitungan Nilai Fitness

4.2.5 Mengambil Data Jarak Pelanggan

Proses pengambilan data jarak didapatkan dari membaca file “jarak.txt”. File tersebut berisi data jarak tempuh kesetiap pelanggan yang kemudian akan dikonversikan kedalam satuan menit menjadi waktu tempuh seperti *Source code 4.5* berikut.

```

public double jrkTempuh (int a, int b, String [][]jarak){
    double jt;
    jt = Double.parseDouble(jarak[a][b]);
    jt = ((jt/30)*60);
    return jt;}

```

Source code 4. 5 Mengambil Data Jarak Pelanggan

4.2.6 Proses Crossover

Metode *crossover* yang digunakann adalah metode PMX *crossover*. Sebelum dilakukan proses *crossover*, maka dilakukan pemilihan induk yang akan mengalami *crossover* sesuai dengan nilai crossover rate yang telah ditentukan. Proses *crossover* dapat dilihat pada *Source code 4.6* berikut.

```

public int [][] Crossover (int [][] pop, double cr){
    int ChildC = (int)(cr* pop.length );
    int [][] cross = new int [ChildC+1][pop[0].length];
    int [][] hasil = new int [ChildC][pop[0].length];
    if (ChildC == 0){
        return pop;
    }
    else {
        Random Random = new Random();
        for (int i = 0; i < ChildC; i=i+2) {
            int parent1 = Random.nextInt(pop.length-1);
            int parent2 = Random.nextInt(pop.length-1);
            while (parent1 == parent2){
                parent2 = Random.nextInt(pop.length-1);
            }
            int [] ank1 = pop [parent1];
            int [] ank2 = pop [parent2];

```

```

int tp = Crandom.nextInt(pop[i].length-1);
int tp2 = Crandom.nextInt(pop[i].length-1);
while(tp==tp2){
    tp2 = Crandom.nextInt(pop[i].length-1);
}
int map1 [] = new int [2];
int map2 [] = new int [2];

map1 [0] = pop[parent1][tp];
map1 [1] = pop[parent1][tp2];
map2 [0] = pop[parent2][tp];
map2 [1] = pop[parent2][tp2];
for (int j = 0; j < pop[i].length; j++) {
    if (ank1[j]== map2[0]){
        if (j == tp){
            ank1 [j] = map2 [0];
        }
        else {
            if (map1 [0] == map2 [1])
                ank1 [j] = map1 [1];
            else
                ank1 [j] = map1 [0];
        }
    }
    else if (ank1 [j] == map2 [1]){
        if (j == tp2){
            ank1 [j] = map2 [1];
        }
        else {
            if (map1 [1] == map2 [0])
                ank1 [j] = map1 [0];
            else
                ank1 [j] = map1 [1];
        }
    }
}
//anak k-2
if (ank2 [j] == map1[0]){
    if (j == tp){
        ank2 [j] = map1 [0];
    }
    else{
        if (map1 [1] == map2 [0])
            ank2 [j]=map2 [1];
        else
            ank2 [j] = map2 [0];
    }
}
else if (ank2 [j] == map1 [1]){
    if (j == tp2){
        ank2 [j] = map1 [1];
    }
    else{
        if (map1 [0] == map2 [1])
            ank2 [j]= map2 [0];
        else
            ank2 [j] = map2 [1];
    }
}
}
ank1 [tp] = map2[0];
ank1 [tp2] = map2[1];
ank2 [tp] = map1[0];
ank2 [tp2] = map1[1];
cross [i] = ank1;

```

```

        cross [i+1] = ank2;
    }
    System.arraycopy(cross, 0, hasil, 0, ChildC);
    }
    return hasil;
}

```

Source code 4. 6 Proses Crossover

4.2.7 Proses Mutasi

Metode mutasi yang digunakan adalah metode *reciprocal exchange mutation*. Sebelum dilakukannya proses mutasi, maka dilakukan pemilihan induk terlebih dahulu. Dimana induk yang terpilih akan mengalami proses mutasi sesuai dengan *mutation rate* yang telah ditentukan .Source sode 4.7 merupakan proses mutasi .

```

public int [][] Mutasi (int [][] pop, double mr ) {
    int ChildM = (int)(mr*pop.length);
    int [][] mutasi = new int [ChildM+1][pop[0].length];
    int [][] hasil = new int [ChildM][pop[0].length];
    if (ChildM == 0) {
        return pop;
    }
    else {
        Random Mrandom = new Random();
        for (int i = 0; i < ChildM; i++) {
            int pp = Mrandom.nextInt(pop.length-1);
            int tp1 = Mrandom.nextInt(pop[i].length-1);
            int tp2 = Mrandom.nextInt(pop[i].length-1);
            while (tp1 == tp2) {
                tp2 = Mrandom.nextInt(pop[i].length-1);
            }
            int tp3 = Mrandom.nextInt(pop[i].length-1);
            while (tp3 == tp2 || tp3 == tp1) {
                tp3 = Mrandom.nextInt(pop[i].length-1);
            }
            int tp4 = Mrandom.nextInt(pop[i].length-1);
            while (tp4 == tp2 || tp4 == tp1 || tp4 == tp3) {
                tp4 = Mrandom.nextInt(pop[i].length-1);
            }

            for (int j = 0; j < pop[i].length; j++) {
                if (j == tp1)
                    mutasi [i][j] = pop [pp][tp2];
                else if (j == tp2)
                    mutasi [i][j] = pop [pp] [tp1];
                else if (j == tp3)
                    mutasi [i][j] = pop [pp][tp4];
                else if (j == tp4)
                    mutasi [i][j] = pop [pp] [tp3];
                else
                    mutasi [i][j] = pop [pp] [j];
            }
        }
        System.arraycopy(mutasi, 0, hasil, 0, ChildM);
    }
    return hasil;
}

```

Source code 4. 7 Proses Mutasi

4.2.8 Proses Seleksi Elitis

Proses seleksi untuk memilih individu terbaik pada generasi selanjutnya dengan menggunakan metode seleksi elitis. Pada seleksi ini, individu yang terpilih merupakan individu dengan nilai *fitness* terbaik yang akan berpeluang terpilih dan menjadi induk untuk generasi selanjutnya. Pemilihan nilai *fitness* dimulai dengan mengurutkan nilai *fitness* terbesar hingga terkecil pada setiap individu. Kemudian seleksi dilakukan dengan memilih *fitness* terbaik sebanyak ukuran populasi. Berikut merupakan proses seleksi elitis pada *Source code* 4.8.

```
public String [][] GetSeleksiElitis (String [][]data, int pop-size){
    String[][]seleksi=new String[1][3];
    seleksi[0][0]=data[0][0];
    seleksi[0][1]=data[0][1];
    seleksi[0][2]=data[0][2];

    for (int i = 0; i < data.length; i++) {
        boolean done = false;
        for (int j = 0; j < seleksi.length; j++) {

            if(Double.parseDouble(data[i][1])>Double.parseDouble(seleksi[j][1])){
                String[][]temp = new String[seleksi.length+1][3];
                System.arraycopy(seleksi, j, temp, j+1,(seleksi.length-j));
                System.arraycopy(seleksi, 0, temp, 0, j);
                temp[j][0]=data[i][0];
                temp[j][1]=data[i][1];
                temp[j][2]=data[i][2];
                seleksi=temp;
                done = true;
                break;
            }
            else {
                if(j==seleksi.length-1&&done==false){
                    String[][]temp = new String[seleksi.length+1][3];
                    System.arraycopy(seleksi, 0, temp, 0, seleksi.length);
                    temp[j+1][0]=data[i][0];
                    temp[j+1][1]=data[i][1];
                    temp[j+1][2]=data[i][2];
                    seleksi=temp;
                    done = true;
                    break;
                }
            }
        }
    }

    String[][]akhir = new String[pop-size][3];
    for (int i = 0; i < pop-size; i++) {
        akhir[i][0]=seleksi[i][0];
        akhir[i][1]=seleksi[i][1];
        akhir[i][2]=seleksi[i][2];
    }
    return akhir;
}
```

Source code 4. 8 Proses Seleksi Elitis

4.2.9 Proses Seleksi Binary Tournament

Proses seleksi untuk memilih individu terbaik pada generasi selanjutnya dengan menggunakan metode seleksi *binary tournament*. Pada seleksi ini, proses awal yang dilakukan adalah memilih dua individu secara acak yang akan dibandingkan besar *fitness*-nya. Dua individu yang dibandingkan diambil sebanyak ukuran populasi. Pemenang dari perbandingan *fitness* dua individu tersebut merupakan individu terpilih yang akan menjadi induk pada generasi selanjutnya. Berikut merupakan proses seleksi *binary tournament* pada *Source code* 4.9.

```
public String [][] GetSeleksiBiner (String [][]data, int pop-size){
    String [][]akhir = new String[pop-size][3];
    Random randoman = new Random();
    for (int i = 0; i < pop-size; i++) {
        int rand1=randoman.nextInt(data.length);
        int rand2=randoman.nextInt(data.length);
        while(rand1==rand2){
            rand2=randoman.nextInt(data.length);
        }
        if((Double.parseDouble(data[rand1][1])>(Double.parseDouble(data[rand2]
[1])))){
            akhir[i][0]=data[rand1][0];
            akhir[i][1]=data[rand1][1];
            akhir[i][2]=data[rand1][2];
        }
        else{
            akhir[i][0]=data[rand2][0];
            akhir[i][1]=data[rand2][1];
            akhir[i][2]=data[rand2][2];
        }
    }
    return akhir;
}
```

Source code 4. 9 Proses Seleksi *Binary Tournament*

4.2.10 Proses Pemilihan Kromosom Terbaik

Pemilihan kromosom terbaik diperoleh dari individu yang memiliki nilai *fitness* terbesar. Pemilihan dilakukan dengan membandingkan nilai *fitness* yang dihasilkan oleh semua kromosom hasil algoritma genetika. Proses pemilihan kromosom terbaik dapat dilihat pada *Source code* 4.10 berikut.

```
public String [][] getBestKrom(String [][] data,String [][] best){
    for (int i = 0; i < data.length; i++) {
        if(Double.parseDouble(data[i][1])>Double.parseDouble(best[0][1])){
            best[0][0]=data[i][0];
            best[0][1]=data[i][1];
            best[0][2]=data[i][2];
        }
    }
}
```

```

        else
        if(Double.parseDouble(data[i][1])==Double.parseDouble(best[0][1])){
        if(Double.parseDouble(data[i][2])<Double.parseDouble(best[0][2])){
            best[0][0]=data[i][0];
            best[0][1]=data[i][1];
            best[0][2]=data[i][2];
        }
        }
    }
    return best;
}

```

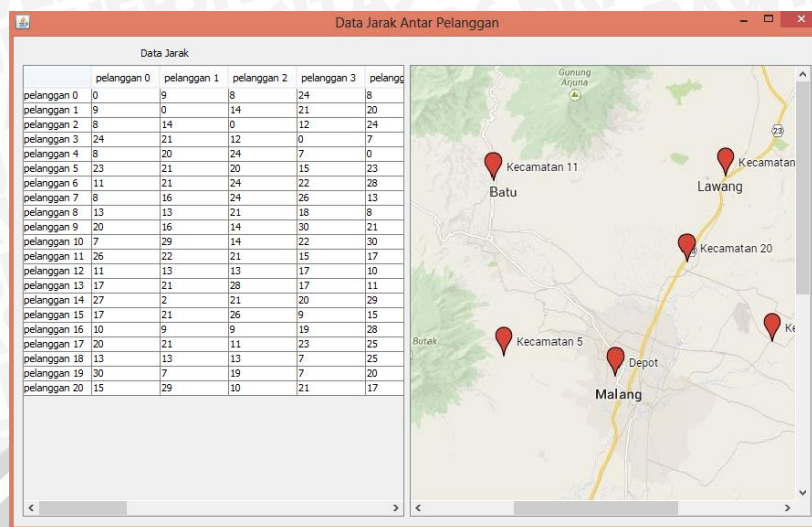
Source code 4. 10 Proses Pemilihan Kromosom Terbaik

4.3 Implementasi User Interface

User Interface dari sistem optimasi distribusi beras bersubsidi terdiri dari tiga halaman. Halaman utama berisi *input* data parameter GA yang berfungsi untuk menjalankan proses perhitungan optimasi rute distribusi beras. Pada Gambar 4.1 merupakan implementasi *user interface* halaman utama.

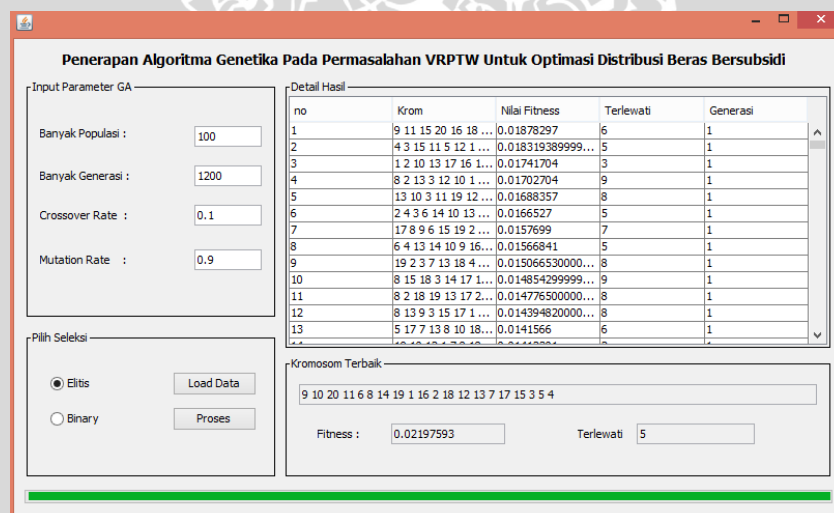
Gambar 4. 1 Implementasi User Interface Halaman Utama

Setelah halaman utama muncul tekan tombol “Load Data” untuk memasukkan data yang akan diproses. Maka akan muncul halaman data seperti Gambar 4.2 berikut.



Gambar 4. 2 Implementasi User Interface Halaman Data

Setelah data berhasil dimasukkan maka kembali lagi pada halaman utama Gambar 4.1 untuk melakukan proses perhitungan dengan menekan tombol “proses” hingga muncul hasil seperti pada Gambar 4.3 berikut.



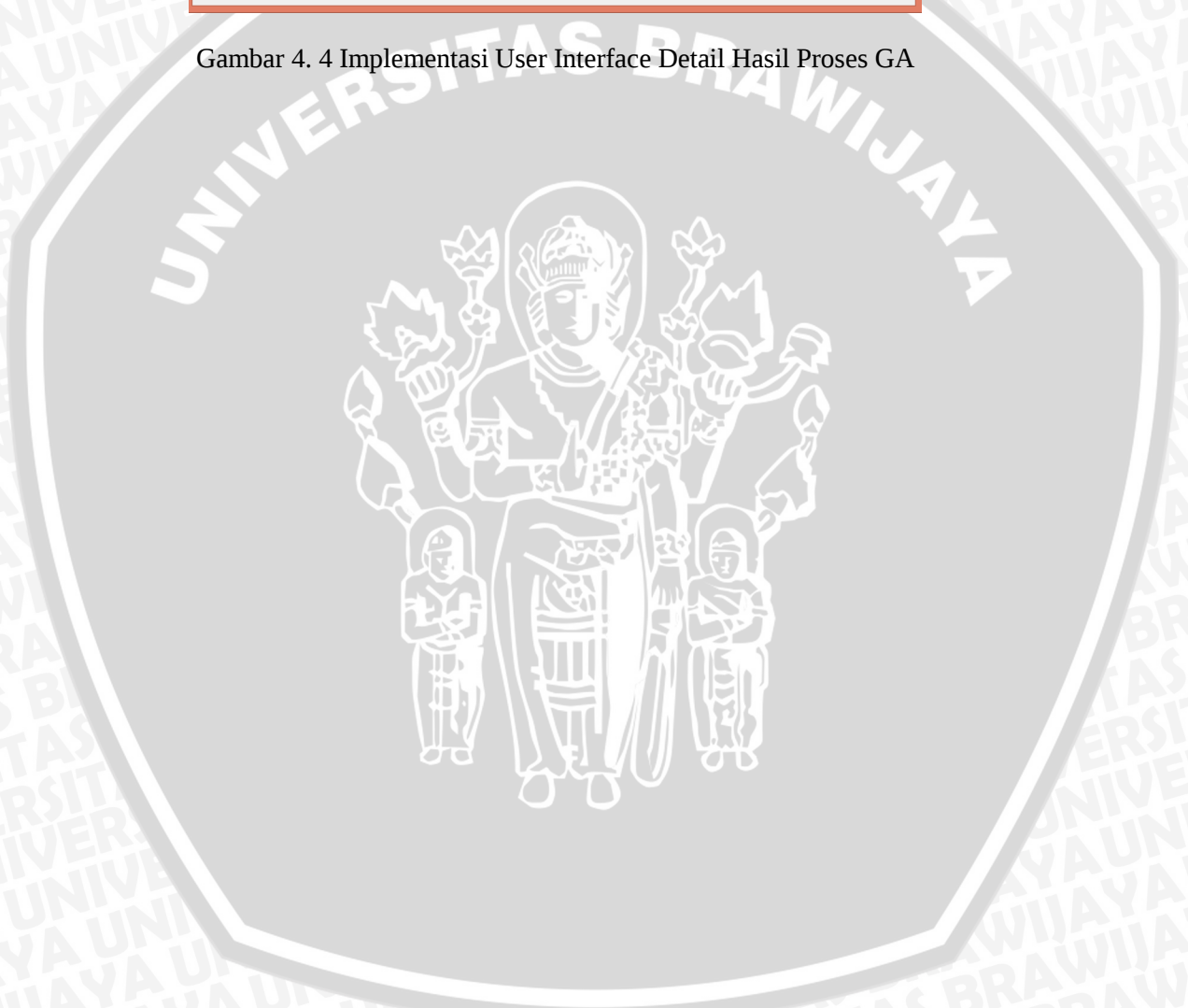
Gambar 4. 3 Implementasi User Interface Hasil Proses

Jika semua hasil proses berhasil ditampilkan, maka halaman ketiga muncul ketika klik pada bagian kromosom. Halaman yang ditampilkan merupakan detail hasil yang ada pada kromosom tersebut yaitu detail truk yang digunakan dengan tujuan masing-masing dan hasil perhitungan manual yang diperoleh dari setiap truk. Pada Gambar 4.4 merupakan implementasi *user interface* detail hasil proses algoritma genetika.

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

Truk	Kromosom	Tujuan	Waktu Tempuh	Waktu Datang	penalti	Waktu Akhir
1	9 10 20 11 6	0	0.0	7.0	0.0	7.0
2	8 14 19 1 16	9	40.0	7.4	0.0	9.49
3	2 18 12 13	10	40.0	10.29	0.0	12.13
4	7 17 15 3 5	20	38.0	12.51	0.0	14.05
5	4	11	14.0	14.19	79.0	14.19
		6	20.0	14.39	0.0	16.13

Gambar 4. 4 Implementasi User Interface Detail Hasil Proses GA



BAB V

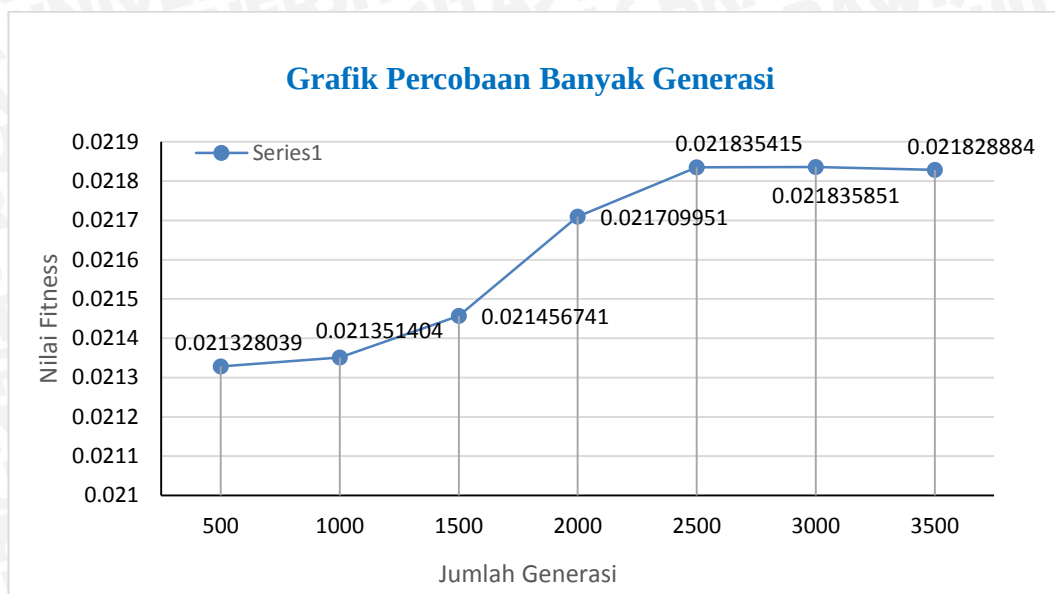
PENGUJIAN DAN ANALISA

5.1 Pengujian dan Analisa Banyaknya Generasi

Pengujian ini dilakukan untuk mengetahui seberapa banyak generasi yang optimal pada permasalahan optimasi distribusi beras dengan metode VRPTW. Pengujian dilakukan sebanyak 10 kali dengan banyak generasi kelipatan 500 yang dimulai dari 500-3500 generasi. Jumlah populasi dalam pengujian sebanyak 60 populasi. Kombinasi Cr dan Mr yang digunakan yaitu 0.5 : 0.5. Proses pengujian menggunakan metode elitis. Hasil dari setiap percobaan akan didapatkan nilai rata-rata *fitness* untuk mengetahui solusi terbaik dari generasi yang optimal. Pada Tabel 5.1 merupakan tabel hasil percobaan banyaknya generasi.

Tabel 5. 1 Hasil Uji Coba Banyak Generasi

Generasi	Percobaan										Rata-rata
	1	2	3	4	5	6	7	8	9	10	Fitness
500	0.019823	0.021766	0.021813	0.021813	0.019349	0.02166	0.021661	0.021661	0.021859	0.021877	0.02132804
1000	0.021705	0.02187	0.019547	0.021734	0.021841	0.019747	0.021796	0.0219	0.021653	0.021721	0.0213514
1500	0.021778	0.02197	0.021798	0.021845	0.021894	0.01994	0.021961	0.021631	0.021922	0.019828	0.02145674
2000	0.021573	0.021838	0.021753	0.02146	0.021596	0.021827	0.021762	0.021685	0.021911	0.021694	0.02170995
2500	0.021728	0.021604	0.021895	0.021884	0.02195	0.021775	0.021932	0.021838	0.02202	0.021728	0.02183542
3000	0.021908	0.021869	0.02212	0.02178	0.021863	0.021694	0.0218	0.021828	0.021715	0.021783	0.02183585
3500	0.021804	0.021728	0.021896	0.021778	0.021778	0.021901	0.021767	0.021882	0.022012	0.021745	0.02182888



Gambar 5. 1 Grafik Hasil Uji Coba Banyak Generasi

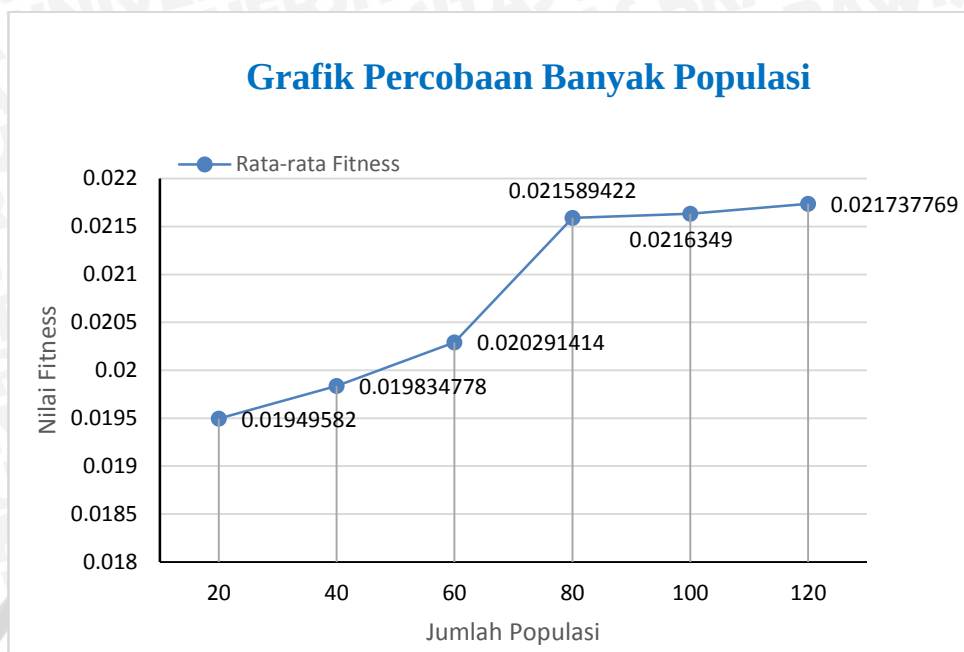
Gambar 5.1 merupakan grafik hasil uji coba banyak generasi dalam 10 kali percobaan. Pada grafik tersebut perilaku GA terlihat dalam penyelesaian masalah, dimana pada setiap generasi GA selalu memperbaiki nilai *fitness* yang didapat. Dapat dilihat pada generasi 500-3500 generasi, nilai *fitness* selalu mengalami peningkatan. Peningkatan yang signifikan berada pada generasi 500-2500. Sedangkan untuk generasi 2500 keatas hingga 3500 nilai *fitness* yang diperoleh tidak mengalami peningkatan yang signifikan sehingga terjadi konvergensi. Kondisi seperti ini mengakibatkan proses reproduksi menghasilkan *offspring* yang hampir sama dengan induknya. Seperti yang dikatakan [MAH-13] bahwa terlalu banyak generasi belum tentu menghasilkan solusi yang lebih baik serta membutuhkan waktu komputasi lebih lama yang kurang efisien. Maka solusi optimal yang diperoleh pada penelitian ini adalah generasi 2500.

5.2 Pengujian dan Analisa Ukuran Populasi

Pada pengujian ini dilakukan untuk mengetahui banyaknya populasi dengan solusi terbaik pada permasalahan VRPTW untuk optimasi distribusi beras bersubsidi. Pengujian ini dilakukan sebanyak 10 kali percobaan pada generasi 1500 dengan kombinasi Cr dan Mr 0.5:0.5. Ukuran populasi yang diujikan adalah kelipatan 20, yaitu dari 20 populasi hingga 120 populasi. Proses pengujian menggunakan metode elitis. Hasil dari setiap percobaan akan didapatkan nilai rata-rata *fitness* untuk mengetahui solusi terbaik dari ukuran populasi yang optimal. Pada Tabel 5.2 merupakan tabel hasil percobaan ukuran populasi.

Tabel 5. 2 Hasil Uji Coba Ukuran Populasi

Banyak Populasi	Nilai Fitness										Rata-rata
	Percobaan banyaknya populasi										<i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
20	0.0190037	0.0186002	0.0195582	0.0191776	0.0191281	0.0191754	0.0192591	0.0198606	0.0195723	0.021623	0.01949582
40	0.0175341	0.021673	0.0197957	0.0215413	0.0215413	0.0193382	0.0195987	0.0197748	0.0196279	0.0179228	0.01983478
60	0.0216215	0.0197353	0.0195668	0.0195835	0.0217765	0.0198318	0.0196514	0.0198229	0.0196534	0.0216711	0.02029141
80	0.0218769	0.0213858	0.0216228	0.0216168	0.021937	0.0220231	0.0218363	0.0218038	0.0198046	0.0219871	0.02158942
100	0.0218927	0.0218743	0.021762	0.0199438	0.0219208	0.0218068	0.0217839	0.0216644	0.0219867	0.0217134	0.0216349
120	0.02188	0.0216779	0.0216976	0.0217876	0.0216974	0.0216193	0.02168	0.0217545	0.0219011	0.0216824	0.02173777



Gambar 5. 2 Grafik Hasil Uji Coba Banyak Populasi

Pada Gambar 5.2 merupakan grafik hasil uji coba banyak populasi dalam 10 kali percobaan. Percobaan pada setiap populasi selalu mengalami peningkatan nilai *fitness*. Kenaikan nilai *fitness* yang signifikan terjadi pada jumlah populasi 20 hingga 80 populasi, sedangkan pada ukuran populasi 80 keatas (populasi 100-120) tidak mengalami kenaikan *fitness* yang signifikan sehingga terjadi konvergensi. Pada kondisi konvergensi, proses eksplorasi tidak berjalan dengan baik sehingga saat reproduksi *offspring* yang dihasilkan akan mirip dengan induknya [MAH-13]. Dari hasil grafik permasalahan optimasi distribusi beras menunjukkan bahwa jumlah populasi yang optimal dihasilkan pada 80 populasi.

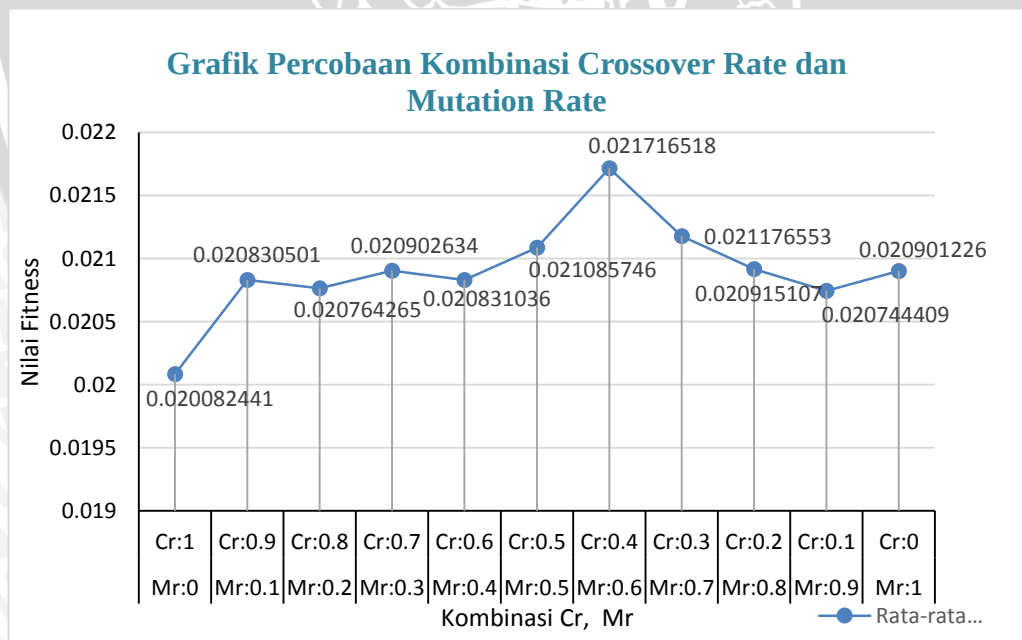
5.3 Pengujian dan Analisa Kombinasi *Crossover rate* dan *Mutation rate*

Pada pengujian ini dilakukan untuk mengukur kombinasi *crossover rate* dan *mutation rate* yang paling tepat pada permasalahan VRPTW untuk optimasi distribusi beras bersubsidi. Pengujian ini dilakukan sebanyak 10 kali percobaan pada 1500 generasi dengan ukuran populasi sebanyak 60 populasi. Nilai *crossover*

rate dan *mutation rate* yang diujikan berskala 0 hingga 1. Proses pengujian menggunakan metode elitis. Hasil dari setiap percobaan akan didapatkan nilai rata-rata *fitness* untuk mengetahui kombinasi *crossover rate* dan *mutation rate* yang optimal dari masing-masing kombinasi. Pada Tabel 5.3 merupakan tabel hasil percobaan kombinasi *Cr* dan *Mr*.

Tabel 5. 3 Hasil Uji Coba Kombinasi *Crossover rate* dan *Mutation rate*

Kombinasi		Nilai fitness										Rata-rata <i>Fintess</i>
		Percobaan										
Cr	Mr	1	2	3	4	5	6	7	8	9	10	
1	0	0.0200632	0.01975	0.020471	0.021203	0.020337	0.020844	0.020563	0.01851	0.018043	0.02104	0.02008244
0.9	0.1	0.021792	0.019532	0.021664	0.019616	0.021607	0.019374	0.021567	0.021582	0.019922	0.021648	0.0208305
0.8	0.2	0.02183	0.021987	0.021662	0.019536	0.021784	0.019932	0.021714	0.019795	0.019433	0.019968	0.02076427
0.7	0.3	0.021768	0.020074	0.019541	0.021951	0.019867	0.021568	0.021555	0.021686	0.021838	0.019179	0.02090263
0.6	0.4	0.021731	0.021977	0.021845	0.019539	0.019125	0.021605	0.021654	0.019464	0.021796	0.019574	0.02083104
0.5	0.5	0.021773	0.021597	0.021836	0.021716	0.019289	0.019644	0.019644	0.021912	0.021702	0.021745	0.02108575
0.4	0.6	0.022047	0.021936	0.021592	0.021591	0.021837	0.021346	0.02171	0.021654	0.021753	0.0217	0.02171652
0.3	0.7	0.021756	0.019565	0.021671	0.021908	0.021905	0.021693	0.019775	0.01952	0.021897	0.022075	0.02117655
0.2	0.8	0.019589	0.021692	0.022116	0.019058	0.021848	0.021824	0.021926	0.019554	0.021567	0.019977	0.02091511
0.1	0.9	0.021854	0.0217	0.021952	0.022166	0.019803	0.019931	0.019871	0.02187	0.019342	0.018956	0.02074441
0	1	0.02166	0.019696	0.021473	0.021612	0.021713	0.021983	0.019768	0.021682	0.019592	0.019832	0.02090123



Gambar 5. 3 Grafik Hasil Uji Coba Kombinasi Mutation rate dan Crossover rate

Pada Gambar 5.3 merupakan grafik hasil uji coba kombinasi *crossover rate* dan *mutation rate* dalam 10 kali percobaan. Rata-rata nilai *fitness* yang diperoleh dalam penelitian ini sangat bervariasi. Tidak ada suatu ketetapan nilai *Cr* maupun *Mr* yang digunakan untuk memperoleh solusi optimal. Suatu permasalahan sangat mempengaruhi nilai kombinasi yang tepat [MAH-13]. Dalam permasalahan VRPTW ini, kombinasi dengan hasil *fitness* terendah terdapat pada *crossover rate* 1 dan *mutation rate* 0 dengan rata-rata *fitness* 0.020082441. Solusi optimal diperoleh pada kombinasi *crossover rate* 0.4 dan *mutation rate* 0.6 dengan rata-rata *fitness* 0.021716518.

5.4 Pengujian dan Analisa Perbandingan Seleksi Metode Elitis dan *Binary Tournament*

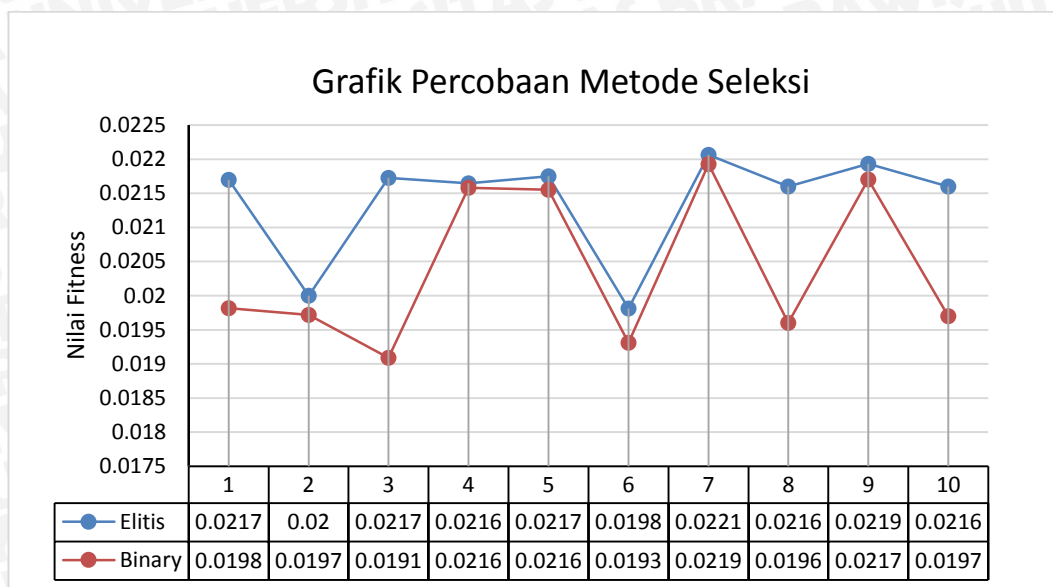
Pada pengujian ini dilakukan untuk membandingkan metode seleksi mana yang terbaik pada permasalahan VRPTW untuk optimasi distribusi beras bersubsidi. Pengujian setiap metode seleksi dilakukan dengan skenario uji seperti berikut :

- 1) Menguji program dengan metode seleksi Elitis dengan jumlah generasi sebanyak 1500 generasi, 60 populasi serta nilai *Cr* dan *Mr* adalah 0.5. Pengujian dilakukan sebanyak 10 kali percobaan.
- 2) Menguji program dengan metode seleksi *Binary Tournament* dengan jumlah generasi sebanyak 1500 generasi, 60 populasi serta nilai *Cr* dan *Mr* adalah 0.5. Pengujian dilakukan sebanyak 10 kali percobaan.

Hasil dari setiap metode seleksi akan didapatkan nilai *fitness* pada masing-masing percobaan untuk mengetahui metode seleksi mana yang lebih baik untuk menghasilkan nilai *fitness* optimal. Pada Tabel 5.4 merupakan tabel hasil percobaan metode seleksi elitis dan *binary tournament*.

Tabel 5. 4 Hasil Uji Coba Metode Seleksi

Metode	Nilai Fitness Percobaan										Rata-rata
	1	2	3	4	5	6	7	8	9	10	Fitness
Elitis	0.021698	0.019998	0.021725	0.021649	0.021748	0.01981	0.022065	0.021599	0.021935	0.021598	0.021383
<i>Binary</i>	0.019817	0.01972	0.019089	0.02158	0.021555	0.019312	0.021927	0.019602	0.021705	0.019702	0.020401



Gambar 5. 4 Grafik Hasil Uji Coba Metode Seleksi

Pada Gambar 5.4 merupakan grafik hasil uji coba metode seleksi elitis dan *binary tournament* dengan 10 kali percobaan. Dari grafik tersebut dapat dilihat bahwa hasil *fitness* metode elitis selalu berada diatas metode *binary tournament*. Dengan rata-rata nilai *fitness* metode elitis adalah 0.021383 lebih besar dibandingkan dengan rata-rata nilai *fitness* metode *binary tournament* yaitu sebesar 0.020401. Hingga percobaan ke-10 metode *binary tournament* mengalami penurunan nilai rata-rata *fitness* jauh dibandingkan metode elitis yang selalu mengalami peningkatan. Hal ini membuktikan bahwa metode seleksi yang terbaik untuk permasalahan optimasi distribusi beras adalah metode seleksi elitis. Karena pada seleksi elitis semua individu terbaik sebanyak *pop-size* langsung terpilih sebagai generasi selanjutnya. Sedangkan untuk metode *binary tournament* dilakukan random individu terlebih dahulu sehingga untuk individu yang tidak bagus masih mempunyai peluang sebagai induk untuk generasi selanjutnya.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil uji coba mengenai penerapan algoritma genetika pada permasalahan VRPTW untuk distribusi beras bersubsidi, terdapat beberapa kesimpulan yaitu:

1. Algoritma genetika dapat menyelesaikan permasalahan VRPTW pada distribusi beras bersubsidi dengan menggunakan representasi permutasi dalam bentuk integer sepanjang 20 gen, *crossover* PMX, *Reciprocal Exchange Mutation*, dan metode seleksi Elitis. Hasil *fitness* yang diperoleh dengan waktu tempuh dan jumlah pinalti yang minimum.
2. Dalam sistem penyelesaian masalah VRPTW pada distribusi beras bersubsidi dengan menggunakan algoritma genetika, seleksi elitis menghasilkan keturunan yang lebih baik dengan rata-rata *fitness* 0.021383 jika dibandingkan dengan metode seleksi *binary tournament* dengan rata-rata *fitness* 0.020401. Hal ini disebabkan bahwa metode elitis memiliki kestabilan individu terbaik yang diurutkan berdasarkan *fitness* tertinggi kemudian diambil sejumlah individu dengan *fitness* terbaik sebanyak *popsi*ze.
3. Perubahan parameter algoritma genetika mempengaruhi rata-rata hasil *fitness*. Pada data berukuran besar (generasi dan populasi), algoritma genetika memperoleh solusi yang optimal dan waktu proses yang sangat lama.
4. Pada ukuran populasi 20-80 populasi, perubahan nilai *fitness* yang dihasilkan selalu mengalami peningkatan yang signifikan. Waktu komputasi yang dibutuhkan berkisar antara 20-60 detik. Berbeda dengan ukuran populasi yang lebih besar dari 80-120 yang hanya mengalami sedikit perubahan pada nilai *fitness*, dengan waktu komputasi berkisar antara 1-3 menit.

5. Pada ukuran generasi 500-2500 menghasilkan nilai *fitness* yang selalu mengalami peningkatan secara signifikan dengan waktu komputasi berkisar antara 25-60 detik. Sedangkan untuk generasi yang lebih besar dari 2500-3500 generasi hanya mengalami sedikit perubahan nilai *fitness* yang dihasilkan, dan waktu komputasi yang dibutuhkan sekitar 1-4 menit.
6. Hasil dari kombinasi *crossover rate* dan *mutation rate* sangat dipengaruhi oleh permasalahan yang telah diselesaikan. Perubahan parameter *Cr* dan *Mr* tersebut yang akan mempengaruhi hasil nilai *fitness*. Dari hasil serangkaian pengujian, parameter yang tepat diperoleh pada kombinasi *Cr* : 0.4 dan *Mr* : 0.6 dengan rata-rata *fitness* sebesar 0.02171652.
7. Solusi optimal diperoleh pada nilai *fitness* yang dihasilkan setelah terjadinya konvergensi. Nilai optimal yang dihasilkan dari penelitian ini terdapat pada populasi : 80 dengan rata-rata *fitness* = 0.02158942, generasi : 2500 dengan rata-rata *fitness* = 0.02183542, *Cr* : 0.4, dan *Mr* : 0.6 dengan rata-rata *fitness* = 0.02171652. Kondisi ini menunjukkan bahwa algoritma genetika sulit mendapatkan solusi yang lebih baik pada penambahan iterasi yang hanya membuang waktu.

6.2 Saran

Aplikasi ini dapat dikembangkan untuk menyelesaikan permasalahan optimasi distribusi beras bersubsidi dengan menggunakan metode *crossover* dan metode mutasi, serta metode seleksi yang berbeda hingga mempengaruhi nilai *fitness* suatu individu. Pada penelitian lebih lanjut agar menambahkan jumlah pelanggan dan jarak yang bervariasi dengan memperhatikan perhitungan jarak tempuh sesuai kondisi jalan satu arah dan arah berlawanan

DAFTAR PUSTAKA

- [AZM-11] Azmi, N., Jamaran, I., Arkeman, Y., & Mangunwidjaja, D. (2011). Penjadwalan Pesanan Menggunakan Algoritma Genetika Untuk Tipe Produksi Hybrid And Flexible Flowshop Pada Industri Kemasan Karton . *Teknik Industri*, 183.
- [BAF-13] Bafita, R., & Sujianto. (2013). Evaluasi Pelaksanaan Program Bantuan Beras Bersubsidi. *Administrasi Pembangunan*, 165.
- [GEN-00] Gen, M., & Cheng, R. (2000). *Genetic Algorithm and Engineering Optimization*. New York: John Wiley & Sons.
- [HAN-02] Hannawati, A., & Thiang, E. (2002). Pencarian Rute Optimum Menggunakan Algoritma Genetika. *Teknik Elektro Vol 2 No.2*.
- [KAL-01] Kallehauge, B. L., & Madsen, O. B. (2001). *Lagrangean Duality Applied On Vehicle Routing With Time Windows*. Lyngby: Technical University of Denmark.
- [KUS-03] Kusumadewi, S. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- [KUS-08] Kusumawardhani, A. D. (2008). *Studi Implementasi Kebijakan Beras Untuk Rumah Tangga Miskin (Raskin) Di Kelurahan Barusari Semarang* . Semarang: Universitas Diponegoro.
- [MAH-08] Mahmudy, W. F. (2008). Optimasi Multi Travelling Salesman Problem (M-TSP) Dengan Algoritma Genetika. *Seminar Nasional Basic Science V, FMIPA, Universitas Brawijaya, Malang 16 February*.
- [MAH-13] Mahmudy, W. F. (2013). *Algoritma Evolusi Program Teknologi Informasi dan Ilmu Komputer*. Malang: Universitas Brawijaya.
- [MAH-06] Mahmudy, W. F., & Mawaddah, N. K. (2006). Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika. *Kursor, vol. 2, no. 2, pp. 1-8. Jurusan Matematika, FMIPA Universitas Brawijaya, Malang 65145*.
- [MIC-99] Michalewicz, Z. (1999). *Genetic Algorithm + Data Structures = Evolution Programs*. New York: Springer Verlag Berlin Heidelberg.

- [PUT-08] Putri, A. N. (2008). *Penentuan Rute Optimal Untuk Pengangkutan Sampah Menggunakan Algoritma Genetik*. Malang: Universitas Brawijaya.
- [RAZ-11] Razali, N. M., & Geraghty, J. (2011). Genetic Algorithm Performance with Different Selection Strategies in Solving TSP . *Proceedings of the World Congress on Engineering Vol II* , 3.
- [SET-03] Setiawan, K. (2003). *Paradigma Sistem Cerdas*. Surabaya: Bayumedia.
- [SUN-11] Sungkar, Z. (2011). *Analisis Kelayakan Investasi Alat Angkut Raskin Perum Bulog Drive DKI Jakarta Melalui Optimasi Rute Dan Jumlah Kendaraan Menggunakan Metode Vehicle Routing Problem Algoritma Diferential Evolution*. Depok: Universitas Indonesia.
- [SUY-05] Suyanto. (2005). *Algoritma Genetik Dalam Matlab*. Yogyakarta: Andi.
- [SUY-11] Suyanto. (2011). *Artificial Intelligence (Searching, Reasoning, Planning and Learning)*. Bandung: Informatika.
- [TOT-02] Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem*. Italy: University of Dgli Studi Di Bologna.
- [WID-10] Widodo, A., & Mahmudy, W. (2010). Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner. *Kursor*, vol 5, no.4, pp , 205-211.
- [WID-12] Widodo, T. (2012). *Komputasi Evolutioner*. Yogyakarta: Graha Ilmu.
- [YOZ-13] Yoza, H., Susanty, S., & Imran, A. (2013). Usulan Perbaikan Rute Pendistribusian Beras Bersubsidi Menggunakan Algoritma Genetika . *Institut Teknologi Nasional*, 11-12.

LAMPIRAN DATA JARAK DAN DATA PELANGGAN

1. Data Jarak

Data Jarak Pelanggan(km)																					
	s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s	0	9	8	24	8	23	11	8	13	20	7	26	11	17	27	17	10	20	13	30	15
1	9	0	14	21	20	21	21	16	13	16	29	22	13	21	22	21	9	21	13	7	29
2	8	14	0	12	24	20	24	24	21	14	14	21	13	28	21	26	9	11	13	19	10
3	24	21	12	0	7	15	22	26	18	30	22	15	17	17	20	9	19	23	7	7	21
4	8	20	24	7	0	23	28	13	8	21	30	17	10	11	29	15	28	25	25	20	17
5	23	21	20	15	23	0	16	24	12	27	16	16	7	13	21	10	20	24	30	22	7
6	11	21	24	22	28	16	0	18	23	25	11	10	8	26	21	11	21	23	27	26	8
7	8	16	24	26	13	24	18	0	11	8	13	10	29	24	18	24	28	15	25	30	30
8	13	13	21	18	8	12	23	11	0	8	28	21	28	30	13	25	17	28	28	28	18
9	20	16	14	30	21	27	25	8	8	0	20	29	23	21	23	9	20	29	20	13	25
10	7	29	14	22	30	16	11	13	28	20	0	27	11	12	26	11	18	16	21	12	19
11	26	22	21	15	17	16	10	10	21	29	27	0	7	28	20	19	24	26	14	12	7
12	11	13	13	17	10	7	8	29	28	23	11	7	0	8	27	30	13	21	8	20	29
13	17	21	28	17	11	13	26	24	30	21	12	28	8	0	11	11	27	24	23	22	13
14	27	2	21	20	29	21	21	18	13	23	26	20	27	11	0	20	12	16	12	15	29
15	17	21	26	9	15	10	11	24	25	9	11	19	30	11	20	0	17	11	10	15	8
16	10	9	9	19	28	20	21	28	17	20	18	24	13	27	12	17	0	22	13	19	29
17	20	21	11	23	25	24	23	15	28	29	16	26	21	24	16	11	22	0	8	17	10
18	13	13	13	7	25	30	27	25	28	20	21	14	8	23	12	10	13	8	0	28	13
19	30	7	19	7	20	22	26	30	28	13	12	12	20	22	15	15	19	17	28	0	26
20	15	29	10	21	17	7	8	30	18	25	19	7	29	13	8	8	29	10	13	26	0

2. Data Pelanggan

Node	Buka	Tutup	Pelayanan(min)	Permintaan(kg)
1	08.00	13.00	100	180
2	07.00	12.30	94	190
3	09.00	13.30	105	160
4	08.30	14.30	62	160
5	07.30	14.00	75	150
6	13.00	16.00	94	155
7	07.30	14.00	104	190
8	08.00	17.30	113	180
9	08.00	14.00	109	155
10	10.00	15.30	104	180
11	07.00	13.00	120	180
12	08.00	13.00	74	175
13	13.00	17.30	67	190
14	07.00	13.30	93	160
15	09.30	14.30	77	150
16	08.00	14.00	79	160
17	10.00	17.00	99	165
18	07.30	13.00	65	175
19	08.00	14.30	68	185
20	13.00	17.30	65	155
Total Permintaan				3395