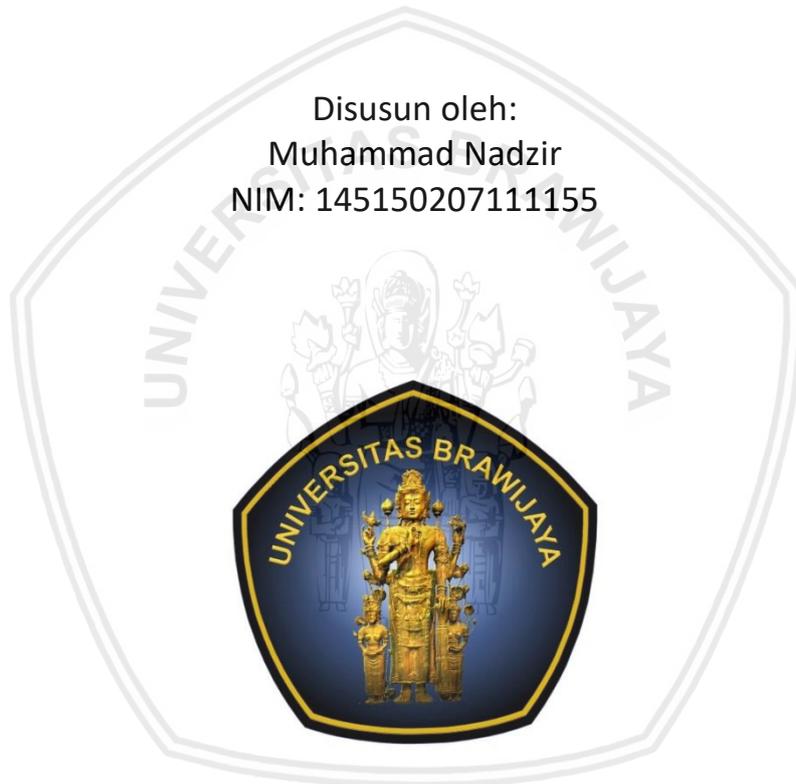


**PENERAPAN MULTI TRAVELLING SALESMAN PROBLEM
PADA OPTIMASI PENDISTRIBUSIAN BANTUAN SOSIAL
BERAS SEJAHTERA STUDI KASUS: PERUM BULOG SUBDIVRE
MALANG**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Nadzir
NIM: 145150207111155



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

PENERAPAN MULTI TRAVELLING SALESMAN PROBLEM
PADA OPTIMASI PENDISTRIBUSIAN BANTUAN SOSIAL BERAS SEJAHTERA
STUDI KASUS: PERUM BULOG SUBDIVRE MALANG

SKRIPSI

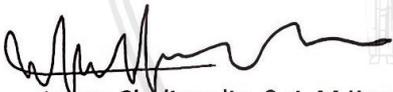
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Nadzir
NIM: 145150207111155

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Imam Cholissodin, S.si, M.Kom.
NIK. 201201 850719 1 001



Bayu Rahayudi, S.T, M.T.
NIP. 19740712 200604 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Asloto Jurdawan, S.T, M.T, Ph.D.
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 3 Januari 2019



Muhammad Nadzir

NIM: 145150207111155

PRAKATA

Alhamdulillah, segala puji syukur Penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat, karunia, serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul: **“Penerapan Multi Travelling Salesman Problem Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang”**.

Skripsi ini merupakan salah satu syarat kelulusan yang harus ditempuh di Fakultas Ilmu Komputer, Program Studi Informatika Universitas Brawijaya Malang. Dan tak lupa pula, penulis juga mengucapkan terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan bantuan selama pengerjaan skripsi ini dari awal hingga terselesaikannya laporan skripsi ini, diantaranya:

1. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., Ir. Heru Nurwasito, M.Kom., Suprpto, S.T., Edy Santoso, S.Si, M.Kom. selaku Dekan, Wakil Dekan I, Wakil Dekan II, Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya Malang. Imam Cholissodin, S.Si, M.Kom., selaku dosen pembimbing 1 dan Bayu Rahayudi, S.T, M.T., selaku dosen pembimbing 2 yang telah memberikan bimbingan, saran, serta arahan selama penyusunan skripsi ini.
2. Seluruh dosen Fakultas Ilmu Komputer yang telah mendidik dan memberikan ilmu serta wawasannya selama menempuh pendidikan dan menyelesaikan skripsi ini.
3. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberikan bantuan serta dukungan kepada penulis selama menempuh pendidikan dan menyelesaikan skripsi ini.
4. Fauzi Muhammad dan Nadiyah selaku kedua orang tua penulis yang telah mendukung penulis dari awal pendidikan hingga menyelesaikan skripsi ini baik secara moril dan materil, serta mendukung melalui setiap doa dan kasih sayangnya yang tulus.
5. Seluruh keluarga besar Ayah dan Ibu Albinhamid Ba'alawy yang selalu memberikan semangat dan bantuan selama ini.
6. Teman-teman dalam satu bidang konsentrasi Komputasi Cerdas yang telah mendukung dalam menyelesaikan skripsi ini Muhammad Misbahul Munir, M. Kevin Pahlevi, Yulfa Hadi Wicaksono, dan Anggita Mahardika yang selalu memberikan bantuan, motivasi dan waktu selama pengerjaan skripsi ini.
7. My lovely friends for you forever winny ardhian.
8. Teman-teman Second Home Studio dalam berbagi pengalaman dan ilmu selama ini Winny Ardhian Septiko, Doni Putra Purbawa, Risailin Dwi Jaka Fauzi, Yulius Firantoko.
9. Teman-teman satu kantin Creative Land (CL) Miftakhul Hanif, I Dewa Gede Ardana P., Mahardika Putra Utama, Asroful Khusna A., Alfian, Farhan Nugraha, Muhammad Faris Mas'ud, Ighustri Riesna V., Imam Nurhidayat, Febri Ramadhani, Irsya Duzh Zhilim B., Muhammad Rexa Mei B., Rio Ridho S., Rizal DC., Syahmi Rifki HP., San Sayidul AA., Sandy Dwi A., Steven Urbani,

- Fardan Ainul Y., Zamahsyari F., Arrizal Amin yang telah menambah pengalaman baru bagi penulis.
10. Seluruh teman-teman TIF-M 2014 yang telah berbagi ilmu dan pengalaman, serta memberikan bantuan selama pengerjaan skripsi ini.
 11. Seluruh teman-teman Informatika UB angkatan 2014 serta semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dan mendukung penulis selama pendidikan sampai terselesaikannya skripsi ini baik secara langsung maupun tidak langsung.
 12. Teman-teman Raion Community yang telah mendukung dalam pengerjaan skripsi ini, dan terima kasih atas pengalaman dan kekeluargaannya selama ini.
 13. Teman-teman Lembaga Pers Mahasiswa DISPLAY yang turut serta membantu dan mendukung penulis dalam menyelesaikan skripsi, terima kasih atas pengalaman dan kekeluargaan yang diberikan selama ini.
 14. Nafiani yang selalu menyempatkan waktunya untuk memberikan semangat tanpa henti untuk mendukung dan membantu penulis dalam menyelesaikan skripsi.ily.

Penulis berharap skripsi yang telah disusun ini bisa memberikan sumbangsih untuk menambah pengetahuan para pembaca, dan akhir kata, dalam rangka perbaikan selanjutnya, penulis akan terbuka terhadap saran dan masukan dari semua pihak karena penulis menyadari skripsi yang telah disusun ini memiliki banyak sekali kekurangan.

Malang, 3 Januari 2019

Penulis
nadzirmuhammad@student.ub.ac.id

ABSTRAK

Muhammad Nadzir, Penerapan Multi Travelling Salesman Problem Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang

Pembimbing: Imam Cholissodin, S.Si, M.Kom., dan Bayu Rahayudi, S.T, M.T.

Distribusi merupakan kegiatan ekonomi yang menjembatani kegiatan produksi dan konsumsi. Proses distribusi yaitu dengan menyalurkan barang dari produsen hingga sampai ke tangan konsumen. Program Bantuan Sosial Beras Sejahtera (Bansos Rastra) memiliki tujuan meningkatkan kualitas layanan masyarakat kurang mampu melalui pemenuhan kebutuhan pangan. Sistem distribusi Bansos Rastra mengirimkan barang ke setiap Titik Distribusi (TD) oleh Perum Bulog sesuai dengan data permintaan distribusi. Pada proses pendistribusian barang dibutuhkan perhitungan pada jarak rute sehingga dapat meminimalkan waktu tempuh perjalanan dengan permasalahan yang digunakan dalam mengolah data dokumen adalah *Multi Travelling Salesman Problem (m-TSP)* dengan Algoritme Genetika. Dari hasil evaluasi pengujian direkomendasikan rute distribusi pada tiap gudang dengan memenuhi batasan yang dibuat. Berdasarkan penelitian yang dilakukan, parameter-parameter optimal yang didapatkan yaitu ukuran jumlah generasi optimal sebanyak 300 generasi, ukuran jumlah populasi yang optimal yaitu 90 populasi. Nilai probabilitas *crossover* sebesar 0,1 dan probabilitas mutasi 0,9 sehingga mendapatkan rata-rata nilai *fitness* terbaik sebesar 2,583. Hasil evaluasi akhir menghasilkan kromosom terbaik dengan selisih jarak prediksi yang lebih efisien dibandingkan jarak aktual sehingga proses distribusi Bansos Rastra dapat lebih optimal.

Kata kunci: *Distribusi, Bansos Rastra, Bulog, mTSP, Algoritma Genetika*

ABSTRACT

Muhammad Nadzir, *Implementation of Multi Traveling Salesman in Optimizing the Distribution of Bantuan Sosial Beras Sejahtera Case Study: Perum Bulog Sub Divre Malang*

Advisor: Imam Cholissodin, S.Si, M.Kom., dan Bayu Rahayudi , S.T, M.T.

Distribution is an economic activity that bridges the production and consumption processes. The distribution process is distributing goods from producers to consumers. Bantuan Sosial Beras Sejahtera (Bansos Rastra) Program has a goal to improving the quality of service for the poor through fulfilling food needs. The distribution system of Bansos Rastra sending goods to each Distribution Point by Perum Bulog in accordance with the data distribution request. In the process of distributing goods, it is necessary to calculate the route distance in order to minimize travel time with the problems used in processing document data is Multi Traveling Salesman Problem (m-TSP) with Genetic Algorithm. From the evaluation results, the distribution routes for each warehouse are recommended by meeting the limits made. Based on the research carried out, the optimal parameters obtained were the size of the optimal number of generations of 300 generations, the optimal size of the population is 90 populations. The crossover probability value is 0.1 and the probability of mutation is 0.9 so that it gets the best average fitness value of 2.583. The final evaluation results produce the best chromosomes with a difference in the predicted distance that is more efficient than the actual distance so that the distribution process of Bansos Rastra can be more optimal.

Keywords: *Distribution, Bansos Rastra, Bulog, Mtsp, Genetic Algorithms*

DAFTAR ISI

| | |
|---|-----------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| PRAKATA..... | iv |
| ABSTRAK..... | vi |
| ABSTRACT..... | vii |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL..... | xi |
| DAFTAR GAMBAR..... | xii |
| DAFTAR LAMPIRAN | xiv |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar belakang..... | 1 |
| 1.2 Rumusan masalah..... | 2 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan masalah | 3 |
| 1.6 Sistematika pembahasan..... | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 5 |
| 2.1 Kajian Pustaka | 5 |
| 2.2 Dasar Teori..... | 7 |
| 2.2.1 Pelaksanaan Distribusi Bantuan Sosial Beras Sejahtera (Bansos Rastra) | 7 |
| 2.2.2 Algoritme Optimasi Heuristik..... | 9 |
| 2.2.3 Optimasi Masalah Kombinatorial..... | 10 |
| 2.2.4 Algoritme Genetika dan Multi Travelling Salesman Problem (m-TSP)..... | 11 |
| BAB 3 METODOLOGI PENELITIAN | 17 |
| 3.1 Studi Kepustakaan | 17 |
| 3.2 Pengumpulan Data | 17 |
| 3.3 Perancangan Sistem..... | 18 |
| 3.4 Implementasi Sistem | 18 |
| 3.5 Pengujian Sistem..... | 18 |



| | |
|---|----|
| 3.6 Evaluasi Sistem..... | 19 |
| BAB 4 PERANCANGAN..... | 20 |
| 4.1 Deskripsi Sistem | 20 |
| 4.2 Persiapan Data | 20 |
| 4.3 Perancangan Proses..... | 21 |
| 4.4 Penyelesaian Masalah Dengan Algoritme Genetika..... | 22 |
| 4.4.1 Insialisasi Populasi Awal | 23 |
| 4.4.2 Reproduksi | 24 |
| 4.4.3 Evaluasi..... | 27 |
| 4.4.4 Seleksi..... | 27 |
| 4.5 Perhitungan Manual | 28 |
| 4.5.1 Data Manualisasi..... | 28 |
| 4.5.2 Representasi Kromosom | 29 |
| 4.5.3 <i>Crossover</i> | 30 |
| 4.5.4 <i>Mutation</i> | 30 |
| 4.5.5 Evaluasi..... | 31 |
| 4.5.6 Seleksi..... | 31 |
| 4.6 Perancangan Antarmuka | 32 |
| 4.6.1 Perancangan Antarmuka Halaman Utama | 32 |
| 4.6.2 Perancangan Antarmuka Halaman Data Jarak..... | 33 |
| 4.6.3 Perancangan Antarmuka Halaman Detail Kromosom | 34 |
| 4.7 Perancangan Pengujian dan Evaluasi | 35 |
| 4.7.1 Perancangan Pengujian Konvergensi..... | 35 |
| 4.7.2 Perancangan Pengujian Parameter Berdasarkan Jumlah Populasi | 36 |
| 4.7.3 Perancangan Pengujian Parameter Berdasarkan Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> | 36 |
| 4.7.4 Perancangan Pengujian Akhir atau Validasi..... | 37 |
| BAB 5 Implementasi | 38 |
| 5.1 Lingkungan Implementasi..... | 38 |
| 5.1.1 Lingkungan Perangkat Keras | 38 |
| 5.1.2 Lingkungan Perangkat Lunak | 38 |
| 5.2 Batasan Implementasi | 38 |

| | |
|---|-----------|
| 5.3 Implementasi Algoritme | 39 |
| 5.3.1 Implementasi Membangkitkan Individu Awal | 39 |
| 5.3.2 Implementasi Proses <i>Crossover</i> | 40 |
| 5.3.3 Implementasi Proses Mutasi..... | 41 |
| 5.3.4 Implementasi Perhitungan Nilai <i>Fitness</i> | 41 |
| 5.3.5 Implementasi Proses Evaluasi..... | 42 |
| 5.3.6 Implementasi Proses Seleksi | 42 |
| 5.4 Implementasi Antarmuka (<i>User Interface</i>)..... | 43 |
| 5.4.1 Implementasi Antarmuka Halaman Utama | 44 |
| 5.4.2 Implementasi Antarmuka Halaman Data Jarak | 44 |
| 5.4.3 Implementasi Antarmuka Halaman Data Hasil Akhir | 45 |
| BAB 6 PENGUJIAN DAN ANALISIS..... | 46 |
| 6.1 Pengujian Konvergensi..... | 46 |
| 6.2 Pengujian Parameter Berdasarkan Jumlah Populasi | 47 |
| 6.3 Pengujian Parameter Berdasarkan Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> | 49 |
| 6.4 Pengujian Akhir atau Validasi | 50 |
| 6.5 Evaluasi Sistem..... | 50 |
| BAB 7 PENUTUP | 52 |
| 7.1 Kesimpulan..... | 52 |
| 7.2 Saran | 52 |
| Daftar REFERENSI | 54 |
| Lampiran A DATA RUTE JARAK TIAP GUDANG..... | 56 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Kajian Pustaka | 6 |
| Tabel 4.1 Data Jarak Proses Pendistribusian Pada GBB Gadang..... | 29 |
| Tabel 4.2 Populasi Awal | 29 |
| Tabel 4.3 Proses <i>Crossover</i> | 30 |
| Tabel 4.4 Proses <i>Mutation</i> | 30 |
| Tabel 4.5 Proses Evaluasi | 31 |
| Tabel 4.6 Proses Seleksi | 32 |
| Tabel 4.7 Rancangan Pengujian Konvergensi | 35 |
| Tabel 4.8 Rancangan Uji Coba Ukuran <i>Popsiz</i> e..... | 36 |
| Tabel 4.9 Rancangan Uji Coba <i>Cr</i> dan <i>Mr</i> | 37 |
| Tabel 4.10 Rancangan Pengujian Validasi..... | 37 |
| Tabel 6.1 Hasil Pengujian Konvergensi | 46 |
| Tabel 6.2 Hasil Uji Coba Jumlah Populasi..... | 48 |
| Tabel 6.3 Hasil Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i> | 49 |
| Tabel 6.4 Hasil Pengujian Akhir atau Validasi | 50 |
| Tabel 6.5 Hasil Evaluasi Selisih Jarak..... | 50 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Jarak Rute Perjalanan..... | 12 |
| Gambar 2.2 <i>Crossover</i> pada representasi permutasi..... | 14 |
| Gambar 2.3 <i>Reciprocal Exchange Mutation</i> | 14 |
| Gambar 2.4 Contoh <i>chromosome</i> untuk mTSP..... | 15 |
| Gambar 3.1 Alur Metodologi Penelitian | 17 |
| Gambar 4.1 Perancangan Sistem | 20 |
| Gambar 4.2 Diagram Alir Sistem | 21 |
| Gambar 4.3 Diagram Alir Proses Implementasi Algoritme Genetika | 23 |
| Gambar 4.4 Diagram Alir Proses Inisialisasi Populasi Awal | 24 |
| Gambar 4.5 Diagram Alir Proses <i>Crossover</i> | 25 |
| Gambar 4.6 Diagram Alir Proses <i>Mutation</i> | 26 |
| Gambar 4.7 Diagram Alir Proses Evaluasi | 27 |
| Gambar 4.8 Diagram Alir Proses Seleksi | 28 |
| Gambar 4.9 Perancangan Antarmuka Halaman Utama..... | 33 |
| Gambar 4.10 Halaman Perancangan Antarmuka Data Jarak..... | 34 |
| Gambar 4.11 Halaman Perancangan Antarmuka Detail Kromosom | 34 |
| Gambar 5.1 Implementasi Antarmuka Halaman Utama | 44 |
| Gambar 5.2 Implementasi Halaman Antarmuka Data Jarak | 45 |
| Gambar 5.3 Implementasi Antarmuka Halaman Data Hasil Akhir..... | 45 |
| Gambar 6.1 Grafik Hasil Pengujian Konvergensi..... | 47 |
| Gambar 6.2 Grafik Hasil Rata-rata Nilai <i>Fitness</i> Pengujian Konvergensi | 47 |
| Gambar 6.3 Grafik Hasil Uji Coba Banyak Populasi..... | 48 |
| Gambar 6.4 Grafik Hasil Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i> | 49 |

KODE PROGRAM

| | |
|--|----|
| Kode Program 5.1 Implementasi Membangkitkan Populasi Awal..... | 39 |
| Kode Program 5.2 Implementasi Proses <i>Crossover</i> | 40 |
| Kode Program 5.3 Implementasi Proses Mutasi..... | 41 |
| Kode Program 5.4 Implementasi Perhitungan Nilai <i>Fitness</i> | 41 |
| Kode Program 5.5 Implementasi Perhitugan Total Jarak | 42 |
| Kode Program 5.6 Implementasi Proses Evaluasi..... | 42 |
| Kode Program 5.7 Implementasi Proses Seleksi | 43 |



DAFTAR LAMPIRAN

| | |
|--|----|
| Lampiran 1 Jarak Gudang Beras Bulog Gadang | 56 |
| Lampiran 2 Jarak Gudang Beras Bulog Kebonagung..... | 56 |
| Lampiran 3 Jarak Gudang Beras Bulog Gadingrejo | 57 |
| Lampiran 4 Jarak Gudang Beras Bulog Kejapanan..... | 58 |



BAB 1 PENDAHULUAN

Bab pendahuluan berisikan problematika dari permasalahan yang dimuat dalam latar belakang sehingga dapat merumuskan masalah. Tujuan merupakan tujuan yang ingin diperoleh sehingga mendapatkan manfaat bagi peneliti dan *stakeholder*. Batasan masalah berisi batasan-batasan yang dibuat oleh peneliti dalam penyelesaian dokumen penelitian, dan sistematika penulisan yang memuat aturan penulisan dalam dokumen.

1.1 Latar belakang

Strategi distribusi yang efektif dan dapat bekerja secara optimal memiliki 3 (tiga) faktor penting. Faktor yang pertama adalah faktor wilayah, yang berarti bahwa diperlukannya pengetahuan tentang wilayah area distribusi agar distributor dapat menggarap wilayah distribusi secara optimal. Faktor kedua adalah faktor persediaan, yang berarti bahwa diperlukan adanya pertimbangan dalam pengambilan keputusan mengenai berapa jumlah persediaan untuk setiap kali pengiriman. Faktor ketiga adalah faktor transportasi, yang berarti bahwa diperlukannya proses yang mengatur perencanaan jadwal pengiriman (*scheduling*) (Eroy, 2010).

Pemerintah Indonesia telah melaksanakan penyaluran beras bersubsidi sejak 1998. Pemerintah mencanangkan program Operasi Pasar Khusus (OPK) bertujuan untuk mengatasi krisis ekonomi yang sedang terjadi. Program OPK atau program beras bersubsidi akhirnya diperluas dari segi fungsionalnya di bidang sosial untuk masyarakat berpendapatan rendah dalam upaya membantu meringankan pengeluaran serta upaya pemenuhan hak maka Program OPK mengeluarkan Program Raskin (Beras Miskin) yang berevolusi menjadi Rastra (Beras Sejahtera) dan saat ini disempurnakan menjadi Bansos Rastra (Bantuan Sosial Beras Sejahtera) (Bulog, 2018).

Rastra sebagai beras sejahtera ditransformasikan menjadi pola bantuan sosial yang berisikan kebutuhan pangan. 16 Maret 2016, Presiden RI menyampaikan Program Penanggulangan Kemiskinan dan Ketimpangan Ekonomi agar bantuan sosial dan subsidi dapat disalurkan dengan non tunai. Pada keluarga penerima manfaat (KPM), proses pendistribusian bantuan sosial secara non tunai memicu perilaku produktif bagi penerima bantuan. Dari sisi pemerintah, penggunaan sistem perbankan dalam upaya pendistribusian program bantuan sosial non tunai berfungsi sebagai upaya peningkatan dalam transparansi dan akuntabilitas program (Bulog, 2018).

Permasalahan optimasi kerap kali berkaitan dengan pencarian solusi pada suatu himpunan dengan menggunakan proses analisis dan kendala untuk mencapai tujuan optimasi. Pada pencarian solusi pendistribusian Bansos Rastra juga terkadang membutuhkan formulasi matematika untuk menemukan nilai solusi yang absolut. Algoritme genetika sebagai metode heuristik merupakan solusi optimal dalam pemecahan masalah untuk bidang optimasi. Dengan melakukan pencarian pada sejumlah titik optimal yang didasarkan pada fungsi

probabilistik, algoritme genetika tidak banyak memerlukan konsep matematika sehingga dapat melakukan semua bentuk fungsi tujuan serta kendala (Gen & Cheng, 2000).

Travelling Salesman Problem (TSP) merupakan jenis optimasi pada permasalahan kombinatorial dimana permasalahan ini melibatkan pencarian rute terbaik pada proses optimasinya. Misal yakni pada rute mana yang memiliki jarak atau waktu paling murah dan jarak paling pendek untuk dilalui seorang *salesman* yang harus mengunjungi sejumlah daerah dengan suatu kondisi *salesman* akan mengunjungi tepat satu kali dan kembali ke tempat semula (Mahmudy, 2015). Pada proses optimasi pendistribusian Bansos Rastra memiliki beberapa pemilihan jalur yang akan diproses oleh lebih dari seorang *salesman* dimana dengan menggunakan lebih dari satu *salesman* merupakan strategi untuk mempercepat proses pendistribusian Bansos Rastra menuju Titik Distribusi. Dalam hal ini permasalahan yang digunakan yaitu *Multi Travelling Salesman Problem (m-TSP)* (Widodo & Mahmudy, 2010).

Berdasarkan paparan sebelumnya, diperlukan suatu metode optimasi dalam proses pendistribusian untuk menyelesaikan permasalahan dalam penyaluran beras bersubsidi. Adapun penelitian sejenis terkait pendistribusian yaitu Optimasi *Multiple Travelling Salesman Problem* Pada Pendistribusian Air Minum Menggunakan Algoritme Genetika (Studi Kasus: UD. Tosa Malang) (Karimah et al., 2017). Dalam penelitiannya, Karimah, dkk melakukan Optimasi Pendistribusian Air Minum Menggunakan *Multiple Travelling Salesman Problem (m-TSP)* dimana representasi permutasi gen berisikan angka yang merupakan nomor pelanggan dan jumlah pesanan yang harus dikunjungi oleh tiap *sales*. Hasil pengujian menunjukkan urutan rute yang dihasilkan dari penerapan algoritme genetika lebih baik daripada penerapan yang dilakukan pada distributor dengan selisih total jarak 89,3 km dan selisih nilai fitness 10,656578. Parameter optimal yang diperoleh yaitu dengan menghasilkan ukuran populasi 180, banyak generasi 400 serta kombinasi crossover rate 0,6 dan mutation rate 0,4.

Dari penjelasan di atas dalam penelitian ini diberi judul “Penerapan Multi Travelling Salesman Problem Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang”. Maksud utama penelitian ini yakni seberapa efektif jalur angkutan dari pemasok (*supplier*) menuju ke titik distribusi. Dengan menggunakan permasalahan *m-TSP* diharapkan juga dapat membantu proses pendistribusian angkutan dari gudang menuju titik distribusi dapat lebih optimal.

1.2 Rumusan masalah

Penentuan rumusan masalah yang ditentukan dalam penelitian yaitu:

1. Bagaimana penentuan parameter algoritme genetika pada tiap pengujian untuk penerapan *Multi Travelling Salesman Problem (m-TSP)* dalam perhitungan rute pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) studi kasus: Perum Bulog Subdivre Malang ?

2. Bagaimana hasil evaluasi dari penerapan *Multi Travelling Salesman Problem (m-TSP)* pada optimasi pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) studi kasus: Perum Bulog Subdivre Malang ?

1.3 Tujuan

Penentuan tujuan yang ingin diperoleh dalam penelitian yaitu:

1. Melakukan uji parameter algoritme genetika pada hasil pengujian untuk penerapan *Multi Travelling Salesman Problem (m-TSP)* dalam perhitungan rute pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) studi kasus: Perum Bulog Subdivre Malang.
2. Mengetahui hasil evaluasi dari penerapan *Multi Travelling Salesman Problem (m-TSP)* pada optimasi pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog Subdivre Malang.

1.4 Manfaat

Diharapkan pada penelitian ini memiliki manfaat yang dapat diperoleh bagi seluruh pihak, manfaat tersebut adalah:

1. Membantu proses penyaluran Bantuan Sosial Beras Sejahtera (Bansos Rastra) dengan menggunakan sistem agar proses penyaluran dapat lebih optimal.
2. Mempermudah bagi pihak perum Bulog dalam proses penyaluran Bantuan Sosial Beras Sejahtera (Bansos Rastra).
3. Meminimalkan timbulnya kesalahan dalam pemilihan rute untuk penyaluran kepada kelompok penerima Bantuan Sosial Beras Sejahtera (Bansos Rastra).

1.5 Batasan masalah

Batasan terkait masalah yang dibuat oleh penulis berfokus pada proses optimasi distribusi Bansos Rastra di Bulog Sub Divre Malang. Perum Bulog sebagai badan kewajiban layanan publik (*Public Service Obligation*) memiliki kebijakan untuk memberikan subsidi dengan berbagai perhitungan sehingga menjadi faktor yang membuat proses distribusi dapat dikatakan menjadi tidak dapat optimal. Dalam hal ini Perum Bulog sebagai layanan yang bersifat *PSO* memiliki ketentuan terkait proses distribusi. Ketentuan yang dimaksud berupa letak gudang menuju titik distribusi tidak dapat ditentukan langsung dan juga faktor pasokan beras yang dilakukan dari petani padi sehingga mempengaruhi proses optimasi pada proses pendistribusian. Adapun konsep proses optimasi distribusi yang dimaksud yakni bagaimana mengoptimalkan proses penyaluran beras pada program Bansos Rastra dari gudang ke titik distribusi menggunakan faktor jarak dalam proses distribusi sehingga dapat menghasilkan proses optimal. Proses penentuan jarak juga dilakukan dengan memperoleh dari aplikasi *Google Maps* yang telah diunduh sehingga tidak terhubung dengan koneksi internet dengan tujuan memperoleh hasil jarak yang tidak dipengaruhi oleh kondisi lalu lintas yang terjadi. Metode yang digunakan yakni Algoritme Genetika dengan *Multi Travelling Salesman Problem (m-TSP)*. Penentuan nilai konstanta pada persamaan *fitness* ditentukan sebesar

100 dengan rasionalisasi untuk mendapatkan nilai *fitness* kisaran 0,01 - 0,1 sehingga tidak memiliki nilai koma yang terlalu rendah (Aditya & Mahmudy, 2016).

1.6 Sistematika pembahasan

BAB I – PENDAHULUAN

Pendahuluan berisi penjabaran atas latar belakang dari penelitian yang diangkat, lalu proses perumusan masalah sebagai dasar dari topik penelitian, manfaat berupa keuntungan yang dapat diperoleh dari penelitian, tujuan yang merupakan hasil penelitian yang ingin diperoleh, batasan-batasan yang dimuat serta sistematika pada penulisan yang sesuai dengan pedoman dalam dokumen penelitian.

BAB II – LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisikan penjabaran pada penelitian yang memiliki keterkaitan dengan topik serta dasar-dasar teori yang diampu dan penerapan metode dalam penelitian yang dilakukan sebagai upaya mendukung dan memperkuat penelitian.

BAB III – METODOLOGI

Pembahasan pada Metodologi mengenai alur yang akan dilaksanakan dalam penelitian dalam penyelesaian masalah Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang.

BAB IV – PERANCANGAN DAN IMPLEMENTASI SISTEM

Menguraikan tentang perancangan konsep penerapan *Multi Travelling Salesman Problem*, perancangan antarmuka (*User Interface*), perhitungan data manual, serta menjabarkan mengenai implementasi data sesuai untuk membangun sistem Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang.

BAB V – PENGUJIAN SISTEM

Menjabarkan pengujian dari sistem serta analisis pembahasan hasil yang didapatkan untuk mendapatkan kesimpulan dari masing-masing pengujian pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang.

BAB VI – PENUTUP

Pengambilan kesimpulan diperoleh dari penelitian serta pemberian saran untuk kelanjutan serta pengembangan penelitian dari Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang.

BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan memaparkan keterkaitan tentang kajian terhadap penelitian-penelitian sebelumnya dengan penelitian yang sedang ditempuh. Pada bab ini juga berisikan dasar teori yang menjelaskan teorema yang berhubungan dengan topik penelitian.

2.1 Kajian Pustaka

Pada landasan kepastakaan, dilakukan kajian terhadap penelitian-penelitian sebelumnya yang memiliki kesesuaian. Penelitian tersebut meliputi pemilihan metode dan juga kesesuaian masalah dengan metode yang dipakai yakni algoritme genetika dengan *Multi Travelling Salesman Problem (m-TSP)*.

Pada penelitian yang dilakukan oleh (Karimah et al., 2017) tentang Optimasi *Multiple Travelling Salesman Problem* Pada Pendistribusian Air Minum Menggunakan Algoritme Genetika yang menghasilkan pengujian dan menunjukkan urutan rute yang dihasilkan dari penerapan algoritme genetika lebih baik daripada penerapan yang dilakukan pada distributor dengan selisih total jarak 89.3 km dan selisih nilai fitness 10.656578. Parameter optimal yang didapatkan yaitu dengan menghasilkan ukuran populasi 180, banyak generasi 400 serta nilai kombinasi *crossover rate* 0.6 dan *mutation rate* 0.4. Solusi yang dihasilkan oleh rekomendasi sistem dapat memberikan urutan rute perjalanan dengan jarak paling minimum dan secara otomatis dapat mempengaruhi biaya dan waktu distribusi menjadi lebih optimal.

Penelitian lain yang dilakukan oleh (Ponraj & Amalanathan, 2014) tentang *Optimizing Multiple Travelling Salesman Problem Considering The Road Capacity* berfokus pada jarak dan rute dilalui oleh *sales* menghasilkan keluaran rute terpendek yang dimana dapat digunakan untuk memecahkan permasalahan *m-TSP* dalam perhitungan kapasitas jalan.

Penelitian yang serupa oleh (Wijayaningrum & Mahmudy, 2016) mengenai *Optimization of Ship's Route Scheduling Using Genetic Algorithm* yang mencoba mengoptimasi rute kapal melalui jarak, waktu, dan rute kapal. Hasil optimasi yang dilakukan menghasilkan parameter yang optimal dengan rincian jumlah *popsize* 30, jumlah generasi 100, hasil *crossover rate* 0.3 dan hasil *mutation rate* 0.7.

Adapun penelitian yang dilakukan oleh (Raditya & Dewi, 2018) tentang Optimasi *Multiple Travelling Salesman Problem (M-TSP)* Pada Penentuan Rute Optimal Penjemputan Penumpang Travel Menggunakan Algoritme Genetika dengan mengoptimalkan jarak, waktu, *cost*, dan rute penjemputan penumpang. Hasil optimasi menghasilkan parameter yang paling optimal dengan ukuran populasi 80 dari 30 lokasi dan 6 *salesman*, generasi 450 yaitu nilai *crossover rate* 0.6 dan *mutation rate* 0.4 dan rata-rata *fitness* yaitu 8.09338.

Penelitian lain dari (Putri, et al., 2015) mengenai Penerapan Algoritma Genetika Untuk *Vehicle Routing Problem with Time Window (VRPTW)* Pada Kasus Optimasi Distribusi Beras Bersubsidi yang meliputi permintaan, waktu jarak tempuh, dan waktu pelayanan untuk distribusi. Nilai probabilitas *crossover rate*

0.4 dan *mutation rate* 0.6 sehingga didapatkan nilai *fitness* terbaik yaitu 0.021716518.

Data perbedaan tiap kajian sebelumnya dengan penelitian yang dilakukan terdapat pada tabel 2.1.

Tabel 2.1 Kajian Pustaka

| No. | Judul dan Objek | Metode | Hasil |
|-----|---|---|--|
| 1. | <p>Judul: Optimasi <i>Multiple Travelling Salesman Problem</i> Pada Pendistribusian Air Minum Menggunakan Algoritme Genetika (Studi Kasus: UD. Tosa Malang) (Karimah et al., 2017)</p> <p>Objek: Distribusi air minum</p> | Algoritme Genetika dengan <i>Multiple Travelling Salesman Problem</i> | Hasil pengujian menunjukkan urutan rute yang dihasilkan dari penerapan algoritme genetika lebih baik daripada penerapan yang dilakukan pada distributor dengan selisih total jarak 89.3 km dan selisih nilai <i>fitness</i> 10.656578. |
| 2. | <p>Judul: <i>Optimizing Multiple Travelling Salesman Problem Considering The Road Capacity</i> (Ponraj & Amalanathan, 2014)</p> <p>Objek: Jarak dan rute yang dilalui oleh sales</p> | Algoritme Genetika dengan <i>Multiple Travelling Salesman Problem</i> | Hasil pengujian menghasilkan keluaran rute terpendek yang dimana dapat memecahkan masalah <i>m-TSP</i> dalam perhitungan kapasitas jalan. |
| 3. | <p>Judul: <i>Optimization of Ship's Route Scheduling Using Genetic Algorithm</i> (Wijayaningrum & Mahmudy, 2016)</p> <p>Objek: Jarak, waktu, dan rute kapal</p> | Algoritme Genetika dengan <i>Multiple Travelling Salesman Problem</i> | Hasil komputasi menghasilkan parameter yang optimal dengan rincian jumlah <i>popsize</i> 30, generasi 100, hasil <i>crossover rate</i> 0.3 dan hasil <i>mutation rate</i> 0.7 |
| 4. | <p>Judul: Optimasi <i>Multiple Travelling Salesman</i></p> | Algoritme Genetika dengan <i>Multiple</i> | Menghasilkan parameter yang paling optimal dengan |

| | | | |
|---|--|---|---|
| | <p><i>Problem (M-TSP) Pada Penentuan Rute Optimal Penjemputan Penumpang Travel Menggunakan Algoritme Genetika (Raditya & Dewi, 2018)</i></p> <p>Objek: Jarak, waktu, <i>cost</i>, dan rute penjemputan penumpang.</p> | <p><i>Travelling Salesman Problem</i></p> | <p>ukuran populasi 80 dari 30 lokasi dan 6 <i>salesman</i>, generasi 450 yaitu nilai crossover rate 0.6 dan mutation rate 0.4 dan rata-rata <i>fitness</i> yaitu 8.09338.</p> |
| 5 | <p>Judul: Penerapan Algoritma Genetika Untuk Vehicle Routing Problem with Time Window (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi (Putri, et al., 2015)</p> <p>Objek: Permintaan, waktu jarak tempuh, dan waktu pelayanan untuk distribusi</p> | <p>Algoritme Genetika</p> | <p>Nilai <i>fitness</i> terbaik yaitu 0.021716518 dengan kombinasi crossover 0.4 dan mutasi 0.6.</p> |

Penelitian ini membahas tentang optimasi distribusi Bantuan Sosial Beras Sejahtera (Bansos Rastra) menggunakan *Multi Travelling Salesman Problem (m-TSP)*. Dimana objek yang ditempuh berfokus pada proses distribusi oleh penyuplai kepada warga yang berhak menerima jatah Bansos Rastra. Dalam proses pendistribusian ini diharapkan dapat meminimalkan jarak yang ditempuh agar proses distribusi dapat lebih optimal. Metode yang digunakan dalam penelitian adalah algoritme genetika menggunakan *Multi Travelling Salesman Problem (mTSP)*.

2.2 Dasar Teori

2.2.1 Pelaksanaan Distribusi Bantuan Sosial Beras Sejahtera (Bansos Rastra)

Badan Urusan Logistik (Bulog) yakni perusahaan umum (perum) yang bergerak dibidang logistik pada kebutuhan pangan. Bulog sebagai perusahaan yang bertanggung jawab pada tugas publik melakukan kegiatan yaitu

menstabilkan harga dasar pada pembelian gabah dan harga pokok serta melakukan pendistribusian Beras Untuk Orang Miskin (Raskin yang sekarang menjadi Bansos Rastra) serta mengorganisir stok pangan (Bulog, 2018).

Program Bantuan Sosial Beras Sejahtera (Bansos Rastra) dapat didistribusikan dalam bentuk produk yaitu berupa beras maupun dalam bentuk non tunai. Bantuan sosial pangan dalam bentuk produk yaitu beras dan disalurkan tiap bulannya tanpa ditentukan harga atau biaya untuk menebus. Bansos Rastra memiliki tujuan meningkatkan kualitas layanan masyarakat miskin melalui pemenuhan kebutuhan pangan pokok serta bertujuan untuk mengurangi beban pengeluaran. Sedangkan bantuan pangan berbentuk non tunai disebut juga Bantuan Pangan Non Tunai (BPNT) digunakan oleh keluarga penerima manfaat (KPM) untuk memperoleh bahan pangan berupa beras dan/atau telur, yang disesuaikan dengan jumlah dan kualitas pada waktu dan tempat yang telah ditentukan (Reduction, 2018).

Bulog memiliki dua fungsi distribusi pangan pokok yakni fungsi komersial dan fungsi pelayanan publik (*Public Service Obligation*) yang berarti bahwa Bulog sebagai BUMN memiliki fokus terhadap kestabilan harga, kemandirian finansial dan profitabilitas yang salah satunya ditentukan oleh manajemen logistik pangan yang modern dan efisien. Dengan menjadi fungsi pelayanan publik (*PSO*) maka dapat dikatakan sulit untuk mendapatkan nilai efisiensi yang tinggi karena beberapa faktor perhitungan. Fungsi pelayanan publik (*PSO*) dapat dimisalkan terdapat dua buah gudang Bulog yang menyimpan stok beras, Gudang A dan Gudang B. Terdapat KPM yang siap untuk menerima pendistribusian beras yang secara lokasi lebih dekat dengan Gudang B. Tetapi, pada kasus ini terdapat gabah yang siap untuk masuk ke Gudang A. Fungsi Bulog sebagai pelayanan publik (*PSO*) akan melakukan pendistribusian pada Gudang A ke KPM yang siap menerima dengan catatan agar stok gabah dapat masuk ke Gudang A. Maka dalam hal ini proses distribusi dapat dikatakan tidak efisien namun memenuhi fungsi pelayanan publik (*PSO*) tersebut.

2.2.1.1 Distribusi

Distribusi merupakan perilaku ekonomi yang menjembatani kegiatan produksi dan konsumsi. Dari kegiatan distribusi, barang dan jasa dapat diterima oleh konsumen. Distribusi secara arti yaitu proses yang menyalurkan barang dari produsen hingga sampai ke tangan konsumen. Produsen adalah pelaku kegiatan produksi. Konsumen merupakan pengguna barang dan jasa. Seseorang yang melakukan kegiatan distribusi dinamakan distributor. Distribusi mempunyai peranan penting bagi konsumen untuk mendapatkan produk yang dihasilkan bagi pihak produsen. Sedangkan Sistem Distribusi sendiri adalah pengaturan untuk proses pendistribusian barang dan jasa dari produsen ke konsumen. Distribusi dibedakan menjadi 2 sistem (Balaipustaka, 2009):

1. Sistem distribusi langsung

Merupakan sistem distribusi secara langsung dari produsen ke konsumen dan tidak menggunakan saluran distribusi. Contoh distribusi sistem ini adalah penyaluran hasil sayuran oleh petani sayur ke pasar secara langsung.

2. Sistem distribusi tidak langsung
Merupakan sistem distribusi yang menggunakan tipe saluran distribusi dalam kegiatan distribusinya dan pada distribusi ini terdapat agen. Contoh: motor, mobil, TV.

2.2.1.2 Sistem Distribusi Bantuan Sosial Beras Sejahtera (Bansos Rastra)

Sistem distribusi Bansos Rastra yaitu mengirimkan barang ke setiap Titik Distribusi (TD) oleh Perum Bulog sesuai dengan data permintaan distribusi. Sistem distribusi beras bersubsidi meliputi (Reduction, 2018):

1. Kementerian Sosial menugaskan pada Perum Bulog dalam upaya proses distribusi Bansos Rastra.
2. Perum Bulog melakukan persediaan akomodasi seperti kendaraan berkapasitas yang dibutuhkan untuk melakukan proses distribusi.
3. Koordinasi Perum Bulog dengan Pemerintah Daerah
4. Proses distribusi yang dimulai dari gudang Bulog yang sudah berisi muatan beras sesuai permintaan.
5. Penyaluran Bansos Rastra ke Titik Distribusi oleh Perum Bulog.
6. Setelah proses distribusii dilakukan, kendaraan kembali dan berakhir pada tempat semula yaitu gudang Bulog.

2.2.1.3 Pemilihan Rute

Pemilihan rute pada setiap proses pengantaran barang merupakan faktor penting yang perlu diperhatikan. Pada proses pengiriman barang dibutuhkan perhitungan pada jarak rute sehingga dapat meminimalkan waktu tempuh perjalanan dan meminimalkan waktu transportasi yang digunakan saat proses distribusi berlangsung. Proses distribusi dilakukan dalam sehari sesuai dengan jam kerja pelaksanaan pengiriman program ke beberapa tujuan. Keberhasilan dalam suatu proses distribusi dapat diukur dari optimasi waktu jarak tempuh, waktu pembongkaran barang, waktu tunggu buka pelayanan dengan kemungkinan mengakibatkan keterlambatan kendaraan kembali. Diperlukan solusi untuk optimasi beberapa faktor yang dapat mempengaruhi penggunaan waktu dalam jarak agar proses distribusi tetap berjalan dengan baik. Pemilihan rute dengan perhitungan jarak tempuh merupakan faktor utama yang mempengaruhi penggunaan waktu dalam proses distribusi. Dimulai pada perhitungan jarak tempuh lalu mengurutkan urutan tujuan sesuai waktu terkecil sehingga dapat memperoleh rute optimum yang terpilih.

2.2.2 Algoritme Optimasi Heuristik

2.2.2.1 Proses Optimasi

Optimasi merupakan set atau kumpulan formula biasanya dalam matematis dan metode numerik dalam upaya memperoleh dan menemukan kandidat terbaik dari kumpulan alternatif yang memungkinkan tanpa harus menghitung dan mengevaluasi secara eksplisit (Santosa & Willy, 2011).

Optimasi juga dapat dibedakan menjadi dua dalam perolehan hasil akhirnya yakni *Local Optimum Solution Optimization* yaitu permasalahan optimasi yang

mempunyai beberapa titik optimum dengan nilai terbaik. Sedangkan *Global Optimum Solution Optimization* merupakan permasalahan optimasi yang hanya memiliki satu titik optimum dengan nilai terbaik (Santosa & Willy, 2011).

Optimisasi hampir dipakai pada semua bidang ilmu diantaranya bidang yang mempunyai keterkaitan dengan ilmu matematis. Optimasi digunakan pada permasalahan tersebut untuk menyelesaikan proses minimalisasi biaya, waktu, dan resiko serta memaksimalkan dari segi kualitas dan keuntungan. Optimasi juga seringkali menjadi tumpuan utama dalam mengambil keputusan, diantaranya (Talbi, 2009):

a. Merumuskan Masalah

Identifikasi masalah merupakan hal pertama yang dilakukan. Terdapat beberapa faktor yang mempengaruhi dan menemukan fungsi tujuan dari permasalahan yang perlu didata dengan cermat.

b. Memodelkan Masalah

Permasalahan dimodelkan secara matematis. Permasalahan pada kejadian sebenarnya bisa jadi kompleks dan sukar untuk dimenegerti, perlu adanya penyederhanaan supaya bisa dinyatakan secara matematis.

c. Optimalisasi

Metode optimasi diterapkan agar mendapatkan nilai *acceptable solution* yang merupakan penyelesaian yang baik sehingga menemukan solusi yang dapat diterima. Tingkat presisi dalam memodelkan suatu permasalahan sangat mempengaruhi hasil yang didapatkan.

d. Menerapkan Penyelesaian Masalah

Penyelesaian yang telah didapat selanjutnya dievaluasi apakah penyelesaian dapat diterima atau tidak. Jika penyelesaian tersebut gagal atau belum memenuhi, maka model permasalahan maupun metode optimasinya perlu diperbaiki dan melakukan pemrosesan ulang.

2.2.2.2 Heuristik dan Metaheuristik

Heuristik adalah algoritme dalam menemukan solusi optimum dengan melakukan *trial-and-error* untuk memperoleh solusi dalam hal pemecahan masalah yang rumit. Akan tetapi, permasalahan tersebut sendiri akan memakan waktu komputasi yang lama dan sulit menggunakan algoritme heuristik ke dalam komputer (Yang, 2010).

Mengenai keterbatasan pada algoritme heuristik, terdapat pendekatan yang berbeda yaitu algoritme metaheuristik. Metaheuristik adalah algoritme yang terinspirasi dari kejadian alam (*nature inspired algorithm*). Algoritme metaheuristik lebih sederhana sehingga proses implementasi ke dalam bahasa pemrograman komputer lebih mudah dan memiliki waktu komputasi lebih efektif dibanding algoritme heuristik (Yang, 2012).

2.2.3 Optimasi Masalah Kombinatorial

Kombinatorial merupakan permasalahan yang memiliki himpunan solusi *feasible* terhingga. Prinsip solusi dari optimasi masalah kombinatorial dapat dihitung dengan enumerasi lengkap, namun jika masalah yang dihadapi kompleks

maka dibutuhkan waktu yang tidak dapat diterima secara cepat (Gen & Cheng, 2000).

2.2.4 Algoritme Genetika dan Multi Travelling Salesman Problem (m-TSP)

Algoritme genetika merupakan algoritme dengan mengadaptasi istilah dalam proses genetika yang berada dalam populasi makhluk hidup. Teori genetika yang dikemukakan oleh Charles Darwin menjelaskan bahwa pada proses genetika alami, setiap individu harus melakukan adaptasi terhadap lingkungan disekitarnya (Setiawan, 2003).

Algoritme genetika merupakan upaya dalam menemukan solusi yang dapat digunakan untuk pemecahan masalah pencarian nilai dalam optimasi (Gen & Cheng, 2000). Konsep pencarian pada algoritme genetika didasarkan pada mekanisme biologis dalam berbagai macam evolusi berupa variasi kromosom pada setiap individu organisme. Variasi kromosom mempengaruhi sistem reproduksi dan tingkat kemampuan hidup organisme sebagai generasi penerus terbaik (Kusumadewi, 2003).

Algoritme genetika memiliki 5 komponen penting, diantaranya (Zbigniew, 1999):

1. Representasi genetika menjadi solusi dari permasalahan yang dihadapi.
2. Pembangkitkan populasi awal.
3. Evaluasi solusi menggunakan fungsi nilai *fitness* pada tiap individunya.
4. Penggunaan operator genetika yang nantinya menghasilkan keturunan (*offspring*).
5. Nilai parameter mencakup pada ukuran populasi (*popSize*) dan nilai probabilitas yang digunakan pada kombinasi *Crossover rate* dan *Mutation rate*.

Multi Travelling Salesman Problem (m-TSP) adalah perluasan dari permasalahan *Travelling Salesman Problem (TSP)*. Perbedaan terletak pada jumlah *salesman* yang melakukan kunjungan. Pada *m-TSP* dapat menggunakan lebih dari satu *salesman* dalam proses penyelesaian masalah. Permasalahan *m-TSP* dapat dijadikan acuan untuk menyelesaikan permasalahan yang berhubungan dengan pencarian rute terpendek maupun permasalahan optimasi (Saptaningtyas, 2012).

Secara matematis formula untuk menyelesaikan masalah minimasi perjalanan dapat dilihat pada persamaan 2.1.

$$Z = \min \{ \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \} \quad (2.1)$$

Dengan kendala (*constrain*):

$$\sum_{i=1}^n x_{ij} = 1, \text{ untuk } j = 1, 2, 3, \dots, n - 1 \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \text{ untuk } i = 1, 2, 3, \dots, n - 1 \quad (2.3)$$

Keterangan:

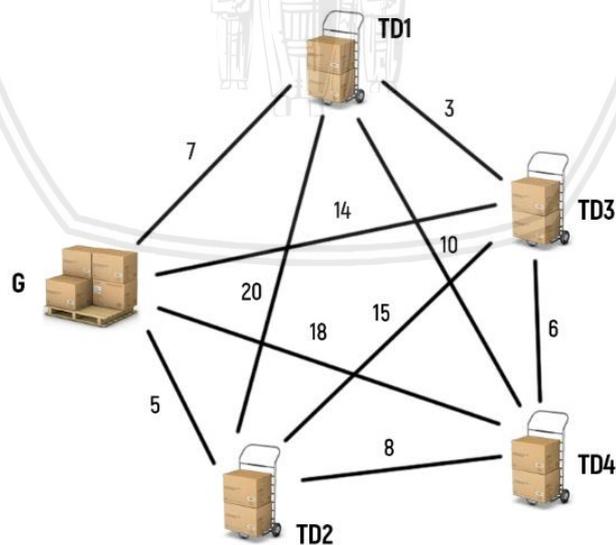
- n = menyatakan banyaknya Titik Distribusi yang dapat disebut sebagai simpul atau *node*.
- x_{ij} = 1 kondisi perjalanan salesman dari simpul i menuju simpul j , bernilai 0 jika tidak ada perjalanan.
- c_{ij} = merupakan biaya atau jarak, dilihat dari tujuan minimasi dari simpul i menuju simpul j .

Persamaan (2.2) dan Persamaan (2.3) menjelaskan setiap simpul hanya dikunjungi satu kali oleh *salesman*.

2.2.4.1 Representasi Node

Representasi *node* atau kromosom merupakan langkah awal dalam penerapan *m-TSP* pada algoritme genetika. Dimana dalam suatu permasalahan direpresentasikan dalam bentuk *node* atau kromosom sesuai dengan bentuk permasalahan yang dipecahkan. Hal ini biasa disebut dengan pengkodean (*encoding*) (Suyanto, 2005).

Penelitian ini menggunakan representasi permutasi yang telah diperkenalkan oleh Gen dan Cheng pada tahun 1997. Representasi permutasi menyatakan solusi penjadwalan dan perjalanan. Setiap gen pada kromosom berupa angka *integer* yang menyatakan nomor tiap simpul (*node*). Misalkan pada sebuah kromosom [3 1 2 4 5] menyatakan perjalanan dimulai dari simpul 3 dan secara berurutan mengunjungi simpul [1 2 4 5] kemudian kembali pada simpul awal yaitu simpul 3 (Mahmudy, 2015).



Gambar 2.1 Jarak Rute Perjalanan

Pada gambar 2.1 Terdapat 5 simpul diidentifikasi sebagai rute perjalanan dari gudang menuju titik distribusi dengan keterangan angka serta busur dengan menunjukkan jarak antar simpul.

Berdasarkan gambar 2.1 dapat dibuat tabel jarak antar simpul sebagai berikut:

Tabel 2.1 Tabel jarak antar simpul

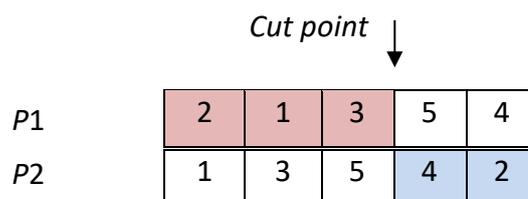
| Node | G | T1 | T2 | T3 | T4 |
|------|----|----|----|----|----|
| G | - | 7 | 5 | 14 | 18 |
| T1 | 7 | - | 7 | 5 | 18 |
| T2 | 5 | 7 | - | 20 | 10 |
| T3 | 14 | 5 | 20 | - | 8 |
| T4 | 18 | 18 | 10 | 8 | - |

Pada Tabel 2.2 menjelaskan tabel jarak antar simpul yang dimana proses distribusi dimulai dari Gudang yang dinotasikan dengan huruf G, lalu menyusuri tiap Titik Distribusi yang disimbol dengan T1, T2, T3, dan T4.

2.2.4.2 Persilangan (Crossover)

Crossover adalah upaya untuk mendapatkan solusi terbaik dengan cara persilangan pada dua individu *parent* yang dipilih secara acak yang menghasilkan individu baru (*offspring*) sebagai anak. Jumlah persilangan pada suatu populasi ditentukan oleh parameter yang disebut dengan *crossover rate / cr*, jadi tidak semua kromosom mengalami *crossover*. Pada proses ini ditentukan dulu sebelumnya tingkat *crossover (crossover rate / cr)*. Nilai yang menyatakan rasio *offspring* dihasilkan dari proses *crossover* terhadap ukuran populasi sehingga menghasilkan *offspring* sebanyak $cr \times popSize$ (Mahmudy, 2015). Beberapa metode pada *crossover* yaitu: *partially-mapped crossover (PMX)*, *order crossover (OX)*, *position-based crossover*, *order-based crossover*, dan *one-cut-point crossover* (Gen & Cheng, 1997). Pada penelitian ini menggunakan *one-cut-point crossover* yang dapat mencegah adanya keturunan mirip dengan munculnya gen ganda yang terdapat pada suatu individu.

Dalam tahap *one-cut-point crossover* ditentukan nilai $cr = 0,5 \times 4 = 2$ *offspring*. Jika penentuan pada setiap nilai *crossover* menghasilkan 2 anak (*offspring*) maka akan terjadi satu kali pada operasi *crossover*. Misalkan terdapat *P1* dan *P2* terpilih sebagai sebagai *parent* dan titik potong (*cut point*) adalah titik 3, maka akan didapatkan *offspring* C1 dan C2 pada Gambar 2.2.



| | | | | | |
|----|---|---|---|---|---|
| C1 | 2 | 1 | 3 | 4 | 2 |
| C2 | 1 | 3 | 5 | 5 | 4 |

Gambar 2.2 Crossover pada representasi permutasi

Selanjutnya pada *offspring* akan mewarisi susunan gen (*chromosome*) dari induknya. Pada Gambar 2.2 tiga bit pertama dari C1 didapatkan dari P1 lalu dua bit terakhir berasal dari P2. C2 mewarisi dua bit pertama dari P2 dan sisa dua bit terakhir berasal dari P1.

2.2.4.3 Mutasi (*mutation*)

Operasi genetika yang dilakukan setelah proses *crossover* ialah mutasi. Operasi mutasi memiliki proses yang sama seperti *crossover* dimana tingkat nilai mutasi (*mutation rate / mr*) harus ditentukan terlebih dahulu. Nilai tersebut menyatakan rasio *offspring* yang dihasilkan pada proses mutasi terhadap ukuran populasi sehingga menghasilkan *offspring* sebanyak $mr \times popSize$ (Mahmudy, 2015). Metode yang digunakan dalam penelitian adalah metode *reciprocal exchange mutation* dimana individu dipilih secara acak dan selanjutnya melakukan perubahan susunan gen didalamnya. Perubahan dilakukan pada individu dilakukan dengan dua posisi pada kromosom dan selanjutnya melakukan penukaran nilai. Proses mutasi dilakukan dalam upaya mengembangkan solusi dari populasi awal dan akan menghasilkan individu baru yang variatif. Penentuan solusi terbaik pada populasi baru dalam mutasi sangat membantu sehingga dapat mempertahankan tingkat variatif pada individu dalam populasi (Mahmudy & Mawaddah, 2006). Pada Gambar 2.3 merupakan contoh mutasi yang dilakukan pada semua gen dalam kromosom.

| | | | | | |
|--------|-------------------|---|-------------------|---|---|
| | XP ₁ ↓ | | XP ₂ ↓ | | |
| Parent | 2 | 1 | 3 | 5 | 4 |
| Child | 2 | 4 | 3 | 5 | 1 |

Gambar 2.3 Reciprocal Exchange Mutation

Multi Travelling Salesman Problem (m-TSP) adalah TSP dimana terdapat kondisi *salesman* dapat lebih dari satu. Nilai yang terdapat pada TSP dapat dimodifikasi pada *mTSP*. Proses modifikasi yaitu dengan menambah jumlah segmen untuk menunjukkan daerah yang dikunjungi oleh *salesman*. Pada Gambar 2.5 merupakan contoh kromosom yang dibuat dengan keterangan terdapat 10 daerah yang akan dikunjungi atau dilalui oleh 3 *salesman*.



| | | | | | | | | | | | | | |
|--------|----------|---|---|---|---|---|----|---|---|----|----------|----|----|
| Posisi | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Gen | 3 | 9 | 6 | 1 | 5 | 8 | 10 | 4 | 2 | 7 | 3 | 3 | 4 |
| | Segmen 1 | | | | | | | | | | Segmen 2 | | |

Gambar 2.4 Contoh *chromosome* untuk mTSP

Segmen 1 berisi nilai gen pada posisi 1 sampai 10 yang menampilkan urutan daerah yang dikunjungi, sedangkan segmen 2 berisi jumlah *salesman* yang mengunjungi daerah. Pada Gambar 2.5 proses rute tiap *salesman* dalam mengunjungi daerah sebagai berikut:

Rute Daerah *Salesman* 1: Kantor Pusat → 3 → 6 → 10 → Kantor Pusat

Rute Daerah *Salesman* 2: Kantor Pusat → 1 → 2 → 9 → Kantor Pusat

Rute Daerah *Salesman* 3: Kantor Pusat → 5 → 7 → 8 → 10 → Kantor Pusat

2.2.4.4 Fungsi Evaluasi (*Fitness*)

Evaluasi dilakukan pada suatu individu berdasarkan fungsi tertentu sebagai ukuran tingkat keberhasilannya. Untuk menyatakan baik tidaknya suatu individu dinyatakan dalam suatu nilai *fitness*. Perhitungan nilai *fitness* adalah solusi yang didapatkan dari pengukuran tingkat individu, dimana individu yang memberikan nilai atau solusi terbaik dapat bertahan hidup. (Widodo & Mahmudy, 2010). Rumus *fitness* ditunjukkan pada Persamaan 2.4.

$$fitness = \frac{C}{Total\ Jarak} \tag{2.4}$$

Keterangan:

Constants = Bernilai 100 yang bertujuan memiliki kemungkinan nilai *fitness* dengan kisaran 0,001 – 0,1 sehingga tidak memiliki nilai yang terlalu rendah (Aditya & Mahmudy, 2016).

Total Jarak (km) = Jarak tempuh yang dihasilkan pada satu perjalanan.

Proses evaluasi akhir juga melakukan perhitungan akurasi dengan membandingkan jarak data aktual dengan jarak data prediksi pada sistem. Perhitungan selisih absolut data jarak tersebut dapat dilihat pada persamaan 2.5.

$$Selisih\ Jarak = Jarak\ Aktual - Jarak\ Prediksi\ Program \tag{2.5}$$

Keterangan:

Selisih jarak (km) = Mencari nilai perbandingan dengan mengurangi nilai dari data aktual dengan data prediksi yang bersifat absolut.

Jarak Aktual (km) = Data yang diperoleh dari data ril yang memuat jumlah jarak dari gudang menuju titik distribusi.

Jarak Prediksi Program (km) = Data hasil program yang memuat perhitungan jumlah jarak dari gudang menuju titik distribusi.



2.2.4.5 Seleksi (*Selection*)

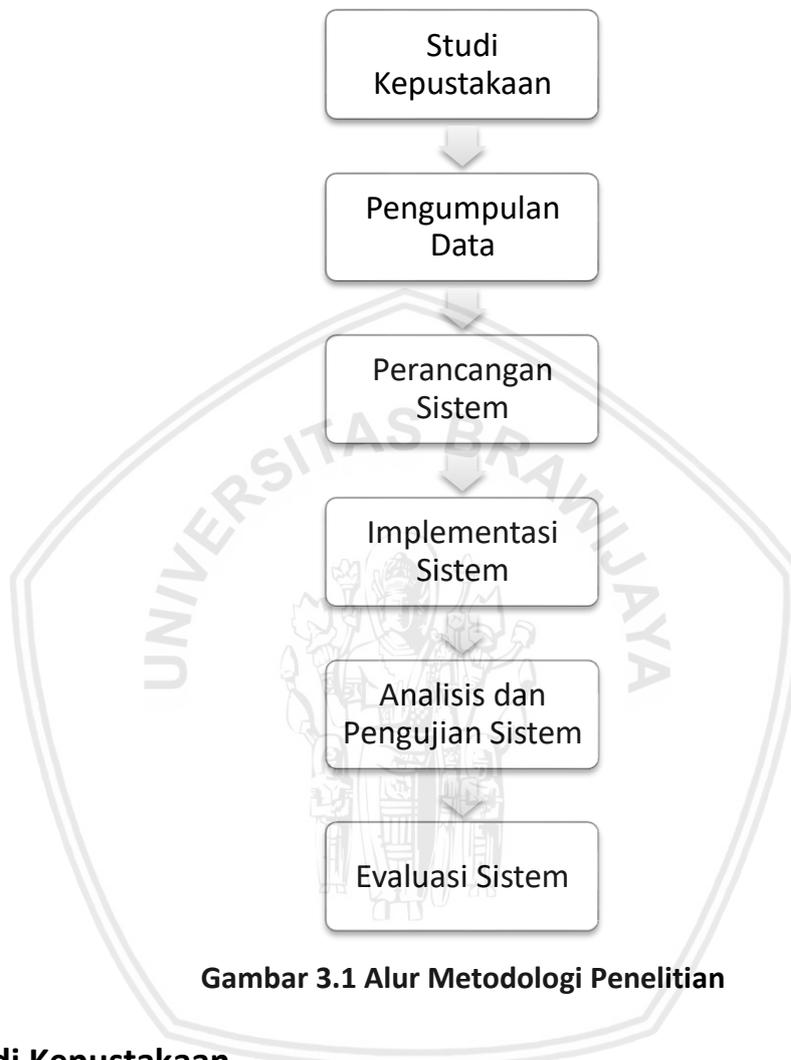
Seleksi dengan melakukan penyeleksian individu dalam proses reproduksi. Nilai *fitness* mempengaruhi proses seleksi yang dimana ketika anggota populasi memiliki nilai *fitness* terbaik maka anggota populasi tersebut mempunyai kesempatan untuk dapat melakukan reproduksi lebih besar. Sedangkan Individu terbaik merupakan individu yang dipilih dari hasil seleksi dan menjadikan *parent* sebagai generasi penerus untuk reproduksi berikutnya (Widodo, 2012).

Pada proses seleksi kondisi populasi yang variatif memegang peran sebagai daerah *sampling*, probabilitas seleksi, dan mekanisme seleksi (Setiawan, 2003). Proses awal seleksi dilakukan penambahan banyak anggota populasi dengan hasil *offspring* dari proses operasi genetika persilangan (*crossover*) dan mutasi (*mutation*). Hasil operasi genetika dengan populasi semula dilakukan seleksi menggunakan metode seleksi tertentu untuk mengambil sejumlah n anggota populasi yang terbaik sesuai dengan jumlah populasi awal (Widodo, 2012).

Seleksi dilakukan berdasarkan hasil nilai *fitness*. Seleksi memiliki beberapa metode diantaranya *Tournament Selection*, *Roulette Wheel Selection*, *Rank Based Fitness Selection*, dan *Elitism* (Mahmudy, 2015). Pada penelitian ini menggunakan metode seleksi *elitism* dimana melakukan pengurutan pada individu terpilih berdasarkan hasil objektif. Fungsi *fitnees* pada proses seleksi digunakan sebagai parameter untuk menentukan tingkat kesempurnaan suatu kromosom yang bertahan hidup dan bisa menjadi *parent* untuk proses reproduksi pada generasi berikutnya. Seleksi *elitism* pada pemilihan individu bersifat memilih nilai *fitness* tinggi pada kromosom dan individu terpilih dapat mewarisi kromosom untuk melakukan pemrosesan pada generasi selanjutnya. Sedangkan nilai *fitness* terendah yang terdapat pada suatu individu akan hilang di generasi berikutnya (Suyanto, 2011).

BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian menjabarkan alur tahapan yang dilalui dalam memperoleh tujuan penelitian. Alur tahapan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Metodologi Penelitian

3.1 Studi Kepustakaan

Studi kepustakaan merupakan tahapan awal yang dilakukan pada penelitian. Studi kepustakaan dilakukan dengan mengumpulkan sumber-sumber yang berkaitan dengan topik penelitian. Diharapkan dari pengumpulan sumber yang terkait pada studi kepustakaan dapat menambah pengetahuan pada teori yang sedang diteliti. Studi kepustakaan pada penelitian ini mengenai proses distribusi, teknik optimasi, Algoritme Genetika, *Travelling Salesman Problem (TSP)*, dan *Multi Travelling Salesman Problem (mTSP)*.

3.2 Pengumpulan Data

Langkah selanjutnya adalah proses mengumpulkan data yang selanjutnya digunakan sebagai data untuk proses optimasi. Pengumpulan data adalah teknik ataupun cara yang digunakan dalam proses pengumpulan data dan dalam

penelitian ini data yang diperoleh berupa data primer. Data primer yang dimaksud yakni data diperoleh langsung dari perum Bulog Sub Divre Malang dengan mengumpulkan data berupa biodata persona beserta data spesifikasi untuk dijadikan data dalam proses optimasi. Data yang didapat berupa pelaksanaan proses distribusi yang dijadwalkan melakukan proses distribusi dalam 4 kali untuk kurun waktu satu minggu. Pelaksanaan distribusi program Bansos Rastra berlangsung dari pukul 08.00-16.00 WIB. Proses distribusi meliputi angkutan truk yang membawa beras dari gudang menuju Titik Distribusi untuk selanjutnya disalurkan kepada masyarakat penerima program Bansos Rastra. Data inilah yang nantinya digunakan sebagai proses penelitian yang dilakukan untuk mencapai tujuan. Pada pengumpulan data jarak untuk Gudang ke Titik Distribusi menggunakan peta Kota Malang dan sekitarnya dengan perhitungan jarak yang diperoleh dari aplikasi *Google Maps* sehingga diharapkan dapat memperoleh data yang akurat dan jelas.

Aplikasi *Google maps* merupakan aplikasi yang berfokus pada layanan peta digital dimana *Google maps* dapat menampilkan gambaran suatu daerah atau denah yang sesuai dengan skala peta tertentu. Data yang diperoleh selanjutnya diolah sehingga dapat melakukan proses penerapan *Multi Travelling Salesman Problem*.

3.3 Perancangan Sistem

Perancangan sistem meliputi rancangan implementasi sistem yang digunakan dalam memperoleh solusi untuk penentuan rute optimal dalam penyaluran beras bersubsidi. Alur sistematis proses perancangan sistem yang dilakukan pengguna yakni memasukan parameter berupa Jumlah Gudang, Titik Distribusi, dan parameter dari Algoritme Genetika pada permasalahan *Multi Travelling Salesman Problem (m-TSP)* berupa *representasi kromosom*, *crossover*, dan *mutation*.

3.4 Implementasi Sistem

Implementasi sistem yakni dengan membangun sistem dari rancangan yang telah ditetapkan. Pada implementasi melakukan penerapan metode penelitian yang dipilih dalam penyelesaian masalah. Pada proses implementasi pada langkah ini digunakan Bahasa pemrograman *java* dengan *software* yang digunakan yaitu *Java Netbeans*.

3.5 Pengujian Sistem

Pengujian sistem merupakan langkah penelitian dengan tujuan menemukan kesalahan atau kekurangan pada sistem dan metode yang digunakan. Pengujian diproses dengan menggantikan nilai tertentu pada parameter kromosom *Multi Travelling Salesman Problem (m-TSP)* untuk memperoleh solusi yang optimal. Pengujian meliputi pengujian konvergensi atau iterasi, pengujian dengan parameter jumlah populasi, pengujian dengan kombinasi nilai *cr* dan *mr*, dan pengujian akhir atau pengujian validasi. Selanjutnya dilakukan proses evaluasi yang bertujuan untuk mendapatkan rute optimal dalam penyaluran beras bersubsidi.

3.6 Evaluasi Sistem

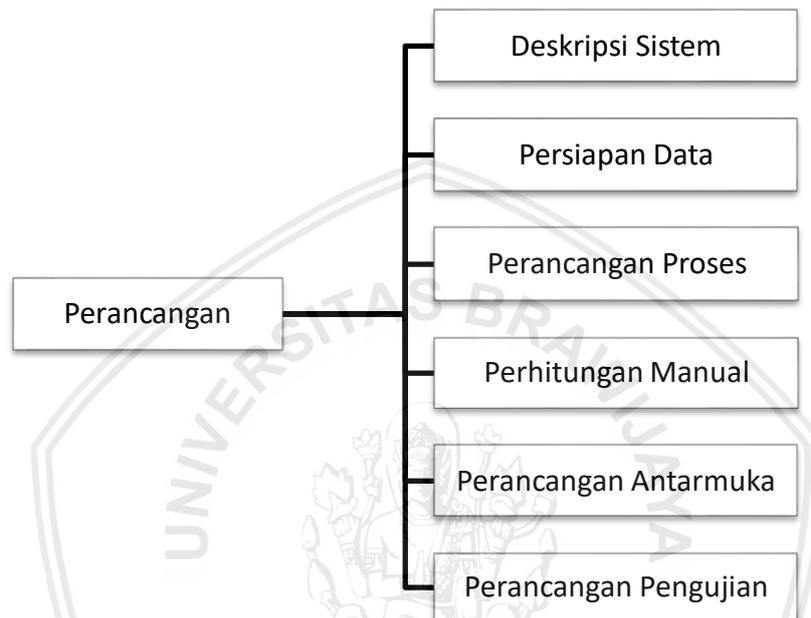
Evaluasi sistem merupakan langkah terakhir dalam penelitian yang dibuat. Evaluasi adalah pengukuran pada tingkat efektifitas sistem untuk memperoleh tujuan awal pada penelitian. Evaluasi pada penelitian ini bermaksud mengetahui tingkat efektivitas dari teknik optimasi dalam implementasi permasalahan *Multi Travelling Salesman Problem (m-TSP)* pada permasalahan proses pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra). Pada evaluasi sistem juga melakukan evaluasi pada sistem apakah telah diimplementasikan dengan baik atau tidak.

Parameter tingkat atau kualitas dari masing-masing solusi dalam pengujian akan dilihat bagaimana tingkat akurasi dari beberapa variasi pada langkah evaluasi ini. Selanjutnya dilakukan perbandingan hasil kinerja sistem untuk digunakan dalam permasalahan *Multi Travelling Salesman Problem (m-TSP)*.



BAB 4 PERANCANGAN

Bab perancangan berfokus pada rancangan yang diimplementasikan pada sistem “Penerapan Multi Travelling Salesman Problem Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang”. Gambar 4.1 merupakan diagram perancangan sistem dalam penelitian ini.



Gambar 4.1 Perancangan Sistem

4.1 Deskripsi Sistem

Pembangunan sistem dengan program komputer menggunakan pemrograman bahasa Java dalam implementasinya. Sistem bertujuan dalam mengoptimasi proses penyaluran beras pada program Bansos Rastra Perum Bulog Sub Divre Malang dari gudang menuju titik distribusi. Untuk kontribusi, penelitian ini akan menguji teori penggunaan masalah *Multi Travelling Salesman Problem (m-TSP)* dalam proses optimasi data. Manfaat utama dari penelitian ini adalah agar dalam proses penyaluran beras pada program Bansos Rastra dapat menghasilkan langkah yang lebih optimal sehingga proses distribusi dari gudang menuju titik distribusi dapat ditempuh dengan waktu yang lebih efisien.

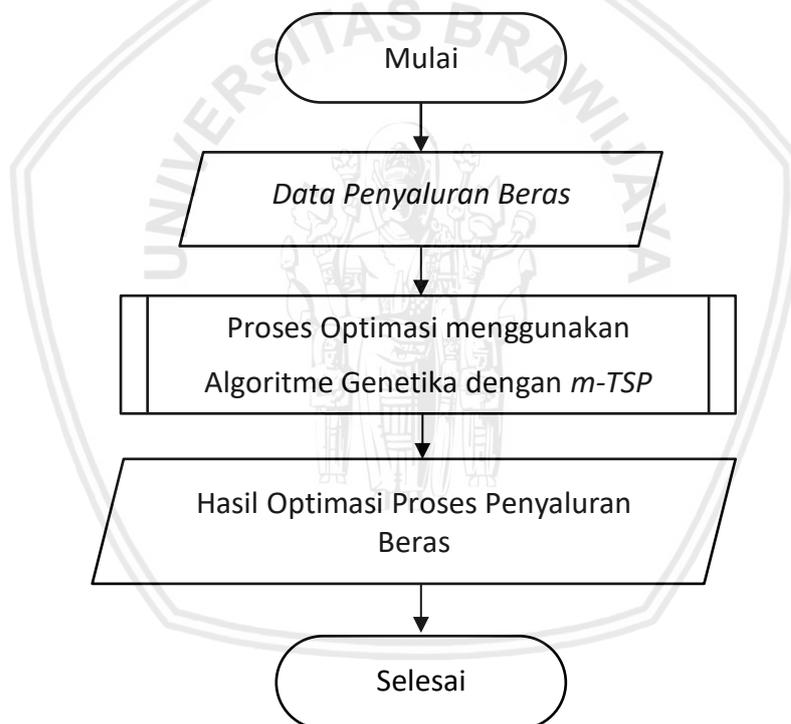
4.2 Persiapan Data

Terdapat data yang disiapkan untuk proses implementasi sistem berupa data primer dimana data diperoleh langsung dari Bulog Subdivre Kota Malang. Data yang didapat berupa pelaksanaan proses distribusi yang dijadwalkan melakukan proses distribusi 4 kali dalam seminggu. Pelaksanaan distribusi program Bansos Rastra berlangsung dari pukul 08.00-16.00 WIB. Proses distribusi menggunakan angkutan truk bermuatan yang membawa beras dari gudang menuju Titik

Distribusi untuk selanjutnya disalurkan kepada masyarakat penerima program Bansos Rastra. Data yang diperoleh selanjutnya diproses dalam penelitian ini, dimana data diproses menggunakan permasalahan *Multi Travelling Salesman Problem (m-TSP)*.

4.3 Perancangan Proses

Perancangan proses menjabarkan identifikasi pada penyaluran beras yang berasal dari gudang Bulog yang berada di wilayah Subdivre Kota Malang. Setelah proses identifikasi dilakukan maka sistem melakukan proses menggunakan permasalahan *Multi Travelling Salesman Problem (m-TSP)*. Sistem melakukan keluaran yang memuat hasil jumlah terbanyak yang dapat disalurkan pada setiap titik distribusi oleh truk yang mengangkut beras dari gudang. Untuk proses dari sistem identifikasi penyaluran beras yang berasal dari gudang menuju titik distribusi disajikan dalam diagram alir Gambar 4.2:



Gambar 4.2 Diagram Alir Sistem

Langkah-langkah perancangan proses pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) berdasarkan Gambar 4.2:

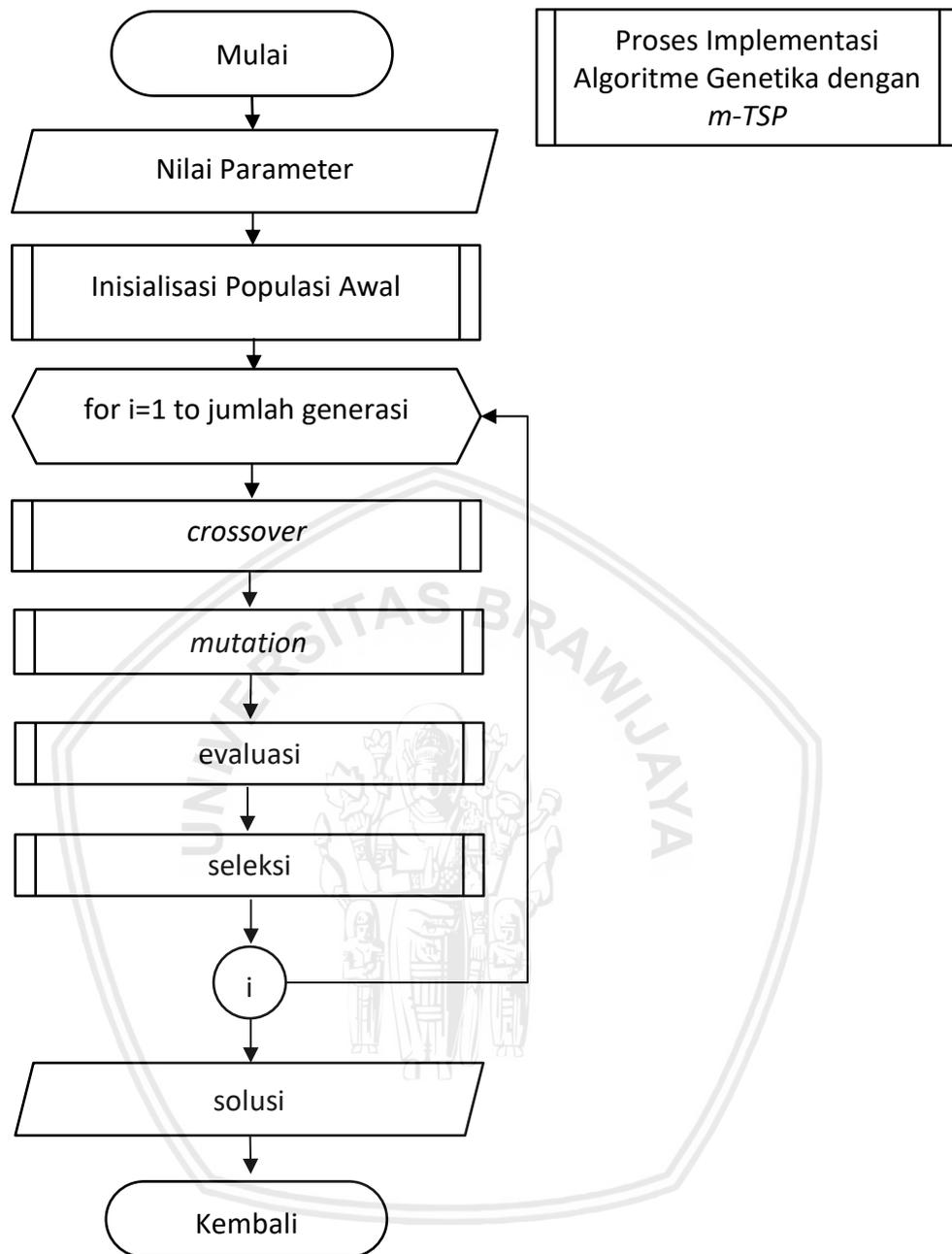
1. Sistem menerima Input Data Penyaluran Beras berupa jarak dari Gudang menuju Titik Distribusi.
2. Melakukan proses Optimasi menggunakan *Multi Travelling Salesman Problem (m-TSP)* meliputi Inisialisasi Populasi Awal, proses *Crossover*, proses *Mutation*, dan Proses Seleksi.
3. Sistem mengeluarkan Output Hasil Optimasi Proses Penyaluran Beras.

4.4 Penyelesaian Masalah Dengan Algoritme Genetika

Algoritme genetika digunakan untuk memperoleh solusi terbaik berupa rute optimal pada proses pendistribusian. Alur siklus penyelesaian algoritma genetika yaitu:

1. Inisialisasi parameter berupa:
 - 1) Jumlah populasi (*popsize*) dimana jumlah banyak populasi yang akan digunakan.
 - 2) *Crossover rate* (*Cr*) digunakan untuk menentukan banyaknya keturunan (*offspring*) yang dihasilkan. Berupa bilangan acak kombinasi dari 0 hingga 1.
 - 3) *Mutation rate* (*Mr*) untuk menentukan banyaknya keturunan (*offspring*) yang dihasilkan. Berupa bilangan acak kombinasi dari 0 hingga 1
 - 4) Jumlah generasi yang merupakan total generasi yang dibuat dalam proses penyelesaian.
2. Inisialisasi pada populasi awal yakni membangkitkan nilai populasi awal dengan menentukan jumlah ukuran populasi (*popsize*) sebelumnya.
3. Membentuk populasi baru dari proses reproduksi. Tahapan pembentukan populasi meliputi:
 - 1) Persilangan (*crossover*) menggunakan metode *one-cut-point crossover*.
 - 2) Mutasi (*mutation*) dengan *reciprocal exchange mutation*.
4. Evaluasi yakni perhitungan pada nilai *fitness* untuk tiap kromosom dan melakukan penyeleksian menggunakan *elitism* dalam upaya pemilihan individu terbaik dan selanjutnya mengurutkan nilai *fitness* tertinggi ke terendah sehingga didapatkan individu dengan jumlah nilai *fitness* tertinggi.
5. Jika pada kondisi akhir telah memenuhi kriteria, kromosom dipilih dengan nilai *fitness* tertinggi yang menjadi solusi terbaik. Gambar 4.3 merupakan alur penyelesaian algoritme genetika dalam proses pendistribusian Bansos Rastra.

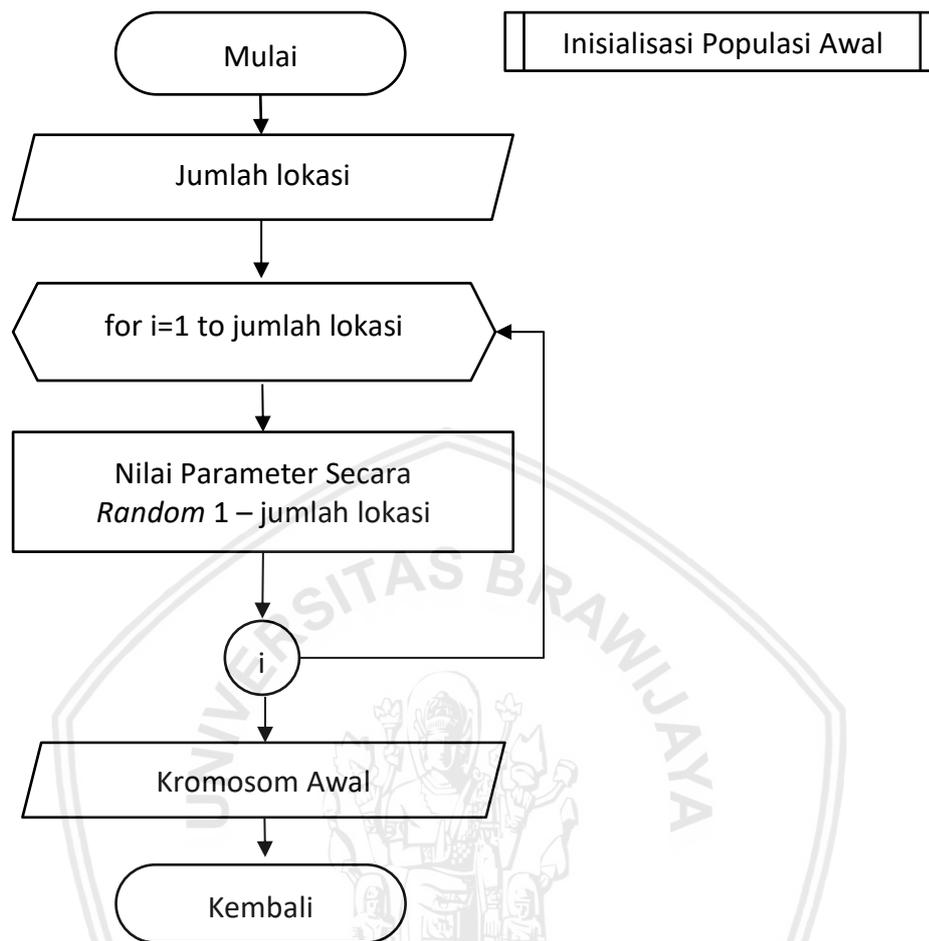
Gambar 4.3 menjelaskan diagram alir dari algoritme genetika. Tahapan dari diagram dimulai dengan memasukkan nilai parameter berupa nilai *cr*, *mr*, dan jumlah generasi. Selanjutnya melakukan proses inisialisasi populasi awal untuk membangkitkan himpunan solusi baru secara acak ke dalam sebuah populasi dimana kromosom memiliki panjang 12 kromosom berisikan titik distribusi pada tiap gudang beras. Proses reproduksi dibagi menjadi dua tahapan, yaitu proses *crossover* dan *mutation* dimana pada proses *crossover* menggunakan metode *one-cut-point crossover* sedangkan pada proses operasi *mutation* akan menggunakan metode *reciprocal exchange mutation*. Setelah proses reproduksi selesai, akan dilanjutkan ke proses evaluasi dimana proses evaluasi akan menyeleksi individu terbaik dan menemukan solusi optimum.



Gambar 4.3 Diagram Alir Proses Implementasi Algoritme Genetika

4.4.1 Inisialisasi Populasi Awal

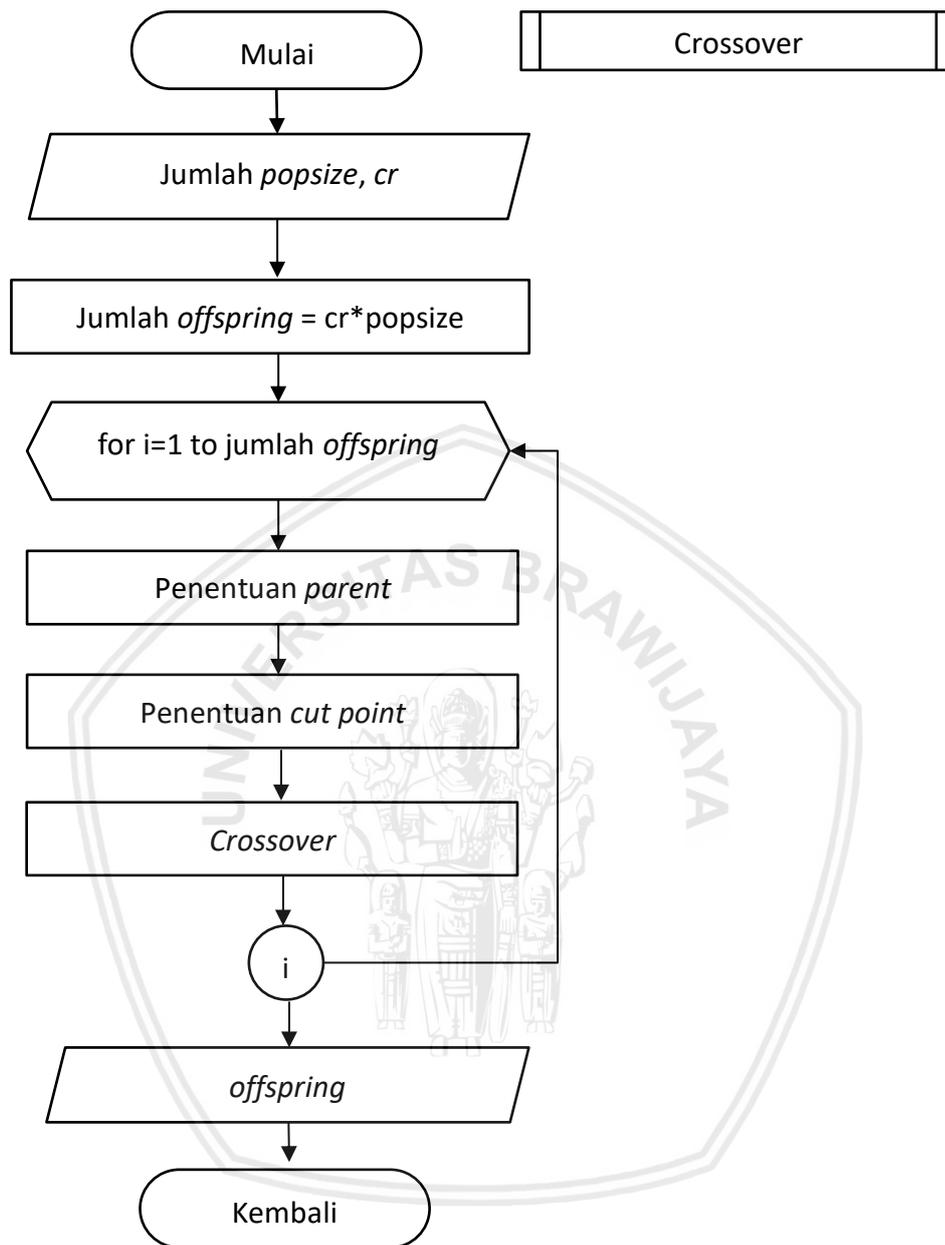
Inisialisasi populasi awal yakni pembentukan individu baru serta penentuan jumlah kromosom yang digunakan. Solusi didapatkan bernilai beda jika individu baru dibentuk dengan secara acak. Terjadi perbedaan pada solusi yang diberikan namun hasil yang diperoleh adalah individu terbaik pada setiap generasinya. Proses inisialisasi populasi awal dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir Proses Inisialisasi Populasi Awal

4.4.2 Reproduksi

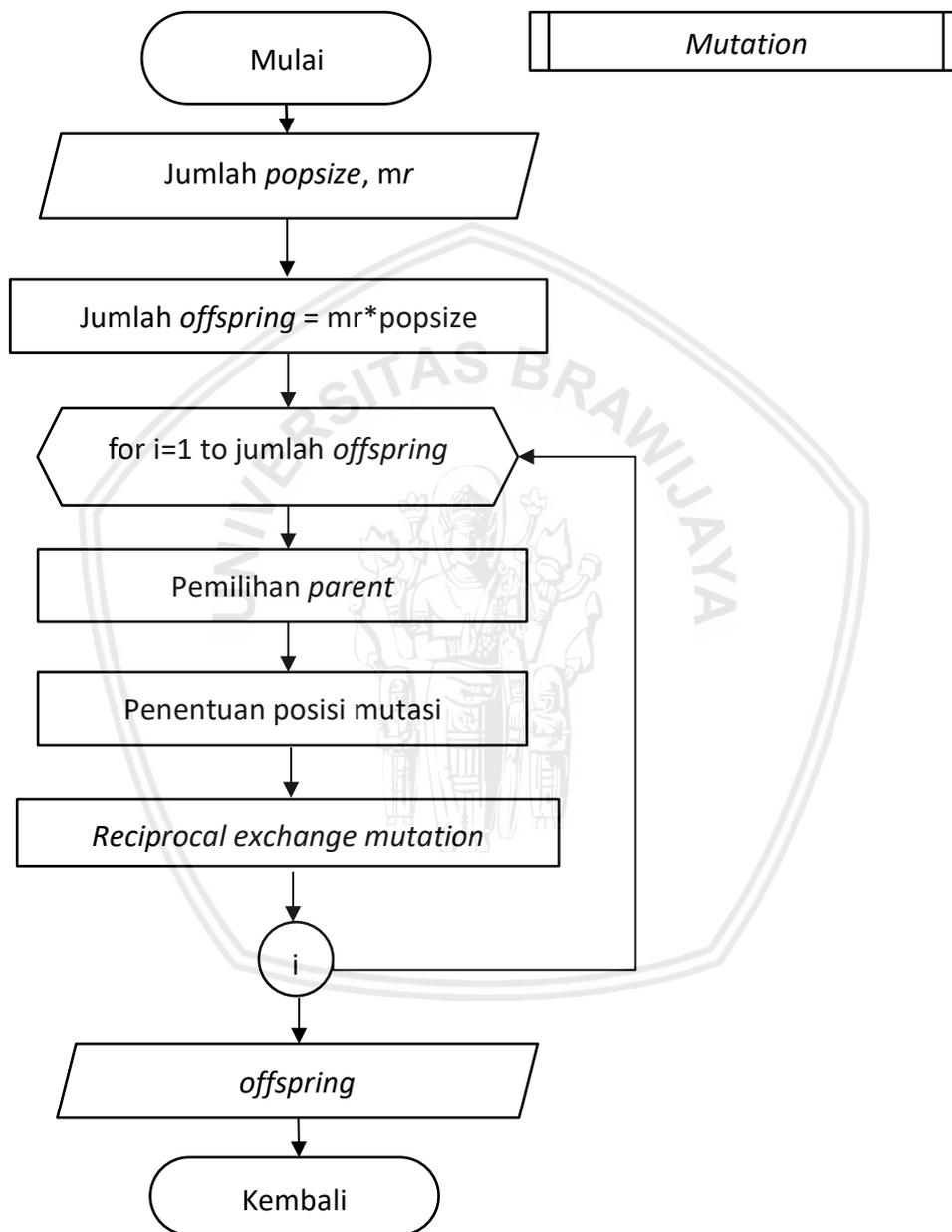
Reproduksi berfungsi mendapatkan keturunan selanjutnya yang menyebabkan penambahan pada variasi individu. Proses reproduksi dengan persilangan (*crossover*) dan mutasi (*mutation*). Reproduksi dengan persilangan (*crossover*) menggunakan metode *one-cut-point crossover*. Sedangkan reproduksi dengan mutasi (*mutation*) akan menggunakan metode *reciprocal exchange mutation*. Sehingga diharapkan hasil dari proses operasi genetika akan menyebabkan keturunan (*offspring*) baru. Gambar 4.5 menyatakan proses *crossover* pada Gambar 4.6 menjelaskan proses *mutation*.



Gambar 4.5 Diagram Alir Proses Crossover

Gambar 4.5 menjabarkan prosedur *crossover* dimana memasukkan nilai *cr* dan jumlah *popsize* untuk mendapatkan jumlah *offspring*, selanjutnya melakukan penentuan *parent* secara acak pada proses *crossover*. Penentuan posisi pada titik potong (*cut point*) dengan metode *one-cut-point crossover* sehingga mendapatkan satu titik potong kromosom. Proses *crossover* dilakukan dengan memotong kromosom *parent 1* (P1) dengan titik potong terpilih dan menggabungkannya dengan potongan kromosom *parent 2* (P2) sehingga mendapatkan kromosom *child* (C1) dan sebaliknya melakukan pemotongan di titik potong untuk kromosom P2 dan menggabungkannya dengan P1 sehingga memperoleh kromosom *child* (C2).

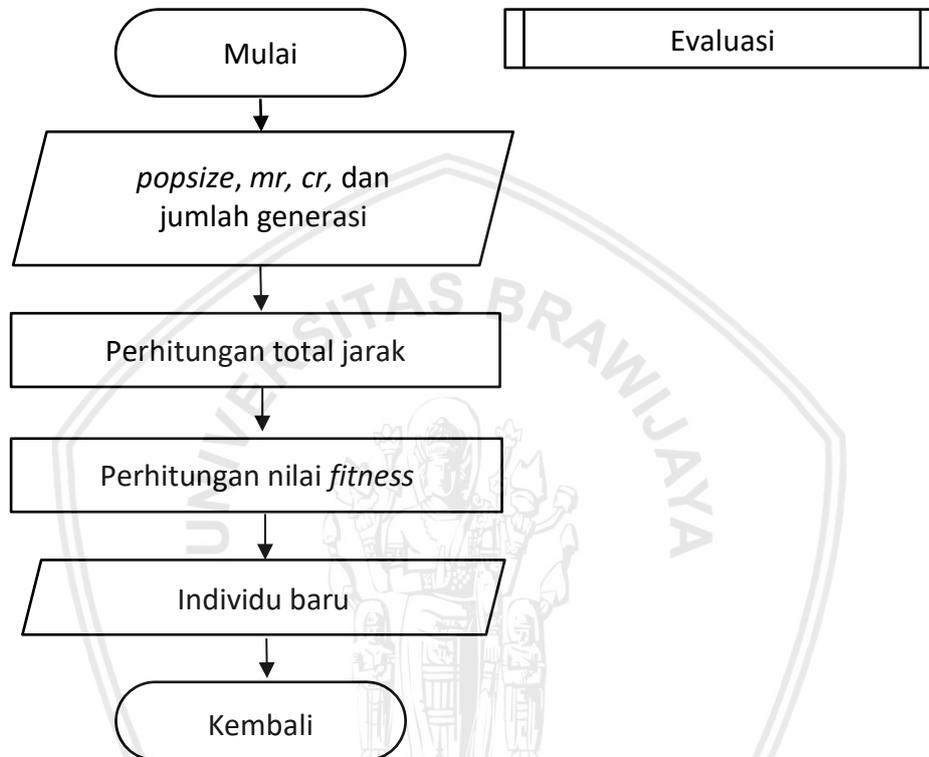
Gambar 4.6 menjelaskan proses mutasi dimulai dengan memasukkan nilai mr dikalikan jumlah $popsiz$ untuk mendapatkan jumlah $offspring$. Metode *Reciprocal Exchange Mutation* digunakan untuk penentuan posisi dimana penentuan tersebut memilih dua titik secara acak.



Gambar 4.6 Diagram Alir Proses *Mutation*

4.4.3 Evaluasi

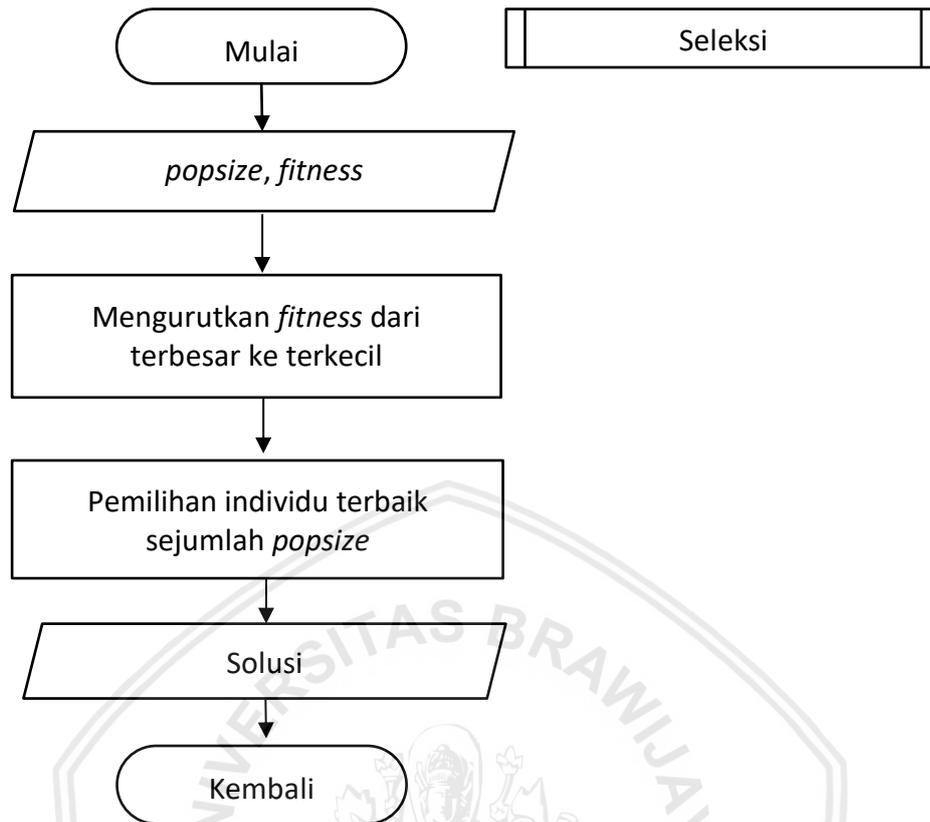
Evaluasi adalah proses pengumpulan semua individu dalam populasi. Evaluasi melakukan pengurutan pada hasil nilai *fitness* yang didapat pada masing-masing kromsوم. Kualitas solusi dari individu yang terbentuk berasal dari nilai *fitness* yang didapatkan (Widodo & Mahmudy, 2010). Proses evaluasi dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Proses Evaluasi

4.4.4 Seleksi

Seleksi merupakan proses pengurutan dari nilai *fitness* terbaik pada seluruh individu dari semua alternatif yang diperoleh. Proses seleksi bertujuan mendapatkan nilai induk terbaik didalam populasi. Metode seleksi yang digunakan adalah *elitism selection* dimana pada metode ini melakukan pengurutan individu pada nilai *fitness* tertinggi ke terendah berdasarkan *popsize* yang telah ditetapkan. Gambar 4.8 menunjukkan alur dari proses seleksi.



Gambar 4.8 Diagram Alir Proses Seleksi

4.5 Perhitungan Manual

Perhitungan manual menggunakan penggunaan parameter seperti jumlah *salesman*, jumlah populasi (*popsize*), banyak generasi dan angka permutasi yang menunjukkan banyak daerah yang dikunjungi oleh tiap *salesman*-nya. Parameter yang digunakan pada perhitungan manualisasi diantaranya:

- Jumlah *salesman* = 3
- Jumlah Populasi (*popsize*) = 3
- Probabilitas *Crossover rate* (*Cr*) = 0,7
- Probabilitas *Mutation rate* (*Mr*) = 0,3
- Jumlah Generasi = 1

4.5.1 Data Manualisasi

Data pada manualisasi yang digunakan adalah data jarak rute dari gudang menuju titik distribusi. Pada perhitungan manual, menggunakan salah satu dari 4 gudang yakni Gudang Beras Bulog (GBB) Gadang. Data jarak dari gudang ke titik distribusi direpresentasikan sebagai kromosom dengan jumlah 12 lokasi dengan menggunakan 3 *salesman*. Nama lokasi titik distribusi juga direpresentasikan dengan angka agar proses pada algoritme genetika dapat dilakukan dengan lebih mudah. Tabel 4.1 merupakan data manualisasi pada rute jarak.

Tabel 4.1 Data Jarak Proses Pendistribusian Pada GBB Gadang

Tabel Jarak Antar Simpul

| Node | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|------|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 50 | 12 | 56 | 9.4 | 9.2 | 11 | 11 | 5.9 | 8.1 | 4.9 | 51 | 44 |
| 1 | 50 | 0 | 46 | 17 | 46 | 50 | 52 | 45 | 49 | 56 | 49 | 11 | 10 |
| 2 | 12 | 46 | 0 | 53 | 17 | 6.1 | 5.3 | 2.8 | 6.8 | 11 | 8.4 | 47 | 40 |
| 3 | 56 | 17 | 53 | 0 | 52 | 56 | 58 | 52 | 55 | 62 | 55 | 7.1 | 11 |
| 4 | 9.4 | 46 | 17 | 52 | 0 | 14 | 16 | 16 | 11 | 17 | 9.8 | 46 | 39 |
| 5 | 9.2 | 50 | 6.1 | 56 | 14 | 0 | 3.4 | 6.6 | 4 | 6.3 | 5.7 | 50 | 43 |
| 6 | 11 | 52 | 5.3 | 58 | 16 | 3.4 | 0 | 6.7 | 6.1 | 5.6 | 7.7 | 52 | 45 |
| 7 | 11 | 45 | 2.8 | 52 | 16 | 6.6 | 6.7 | 0 | 5.6 | 13 | 7.3 | 46 | 39 |
| 8 | 5.9 | 49 | 6.8 | 55 | 11 | 4 | 6.1 | 5.6 | 0 | 8 | 2.3 | 49 | 42 |
| 9 | 8.1 | 56 | 11 | 62 | 17 | 6.3 | 5.6 | 13 | 8 | 0 | 9 | 56 | 49 |
| 10 | 4.9 | 49 | 8.4 | 55 | 9.8 | 5.7 | 7.7 | 7.3 | 2.3 | 9 | 0 | 49 | 42 |
| 11 | 51 | 11 | 47 | 7.1 | 46 | 50 | 52 | 46 | 49 | 56 | 49 | 0 | 5.7 |
| 12 | 44 | 10 | 40 | 11 | 39 | 43 | 45 | 39 | 42 | 49 | 42 | 5.7 | 0 |

4.5.2 Representasi Kromosom

Representasi kromosom mendefinisikan individu dalam populasi yang membutuhkan pengkodean pada kromosom. Representasi kromosom pada algoritme genetika dinyatakan sebagai tujuan atau hasil akhir (Mahmudy, 2015). Representasi kromosom pada metode *m-TSP* selanjutnya dapat disebut dengan representasi node. Representasi kromosom *m-TSP* pada penelitian ini sebanyak 12 dengan menggunakan nilai gen jarak tiap lokasi titik distribusinya. Panjang node ditentukan pada jumlah lokasi pendistribusian beras, dimana perhitungan manualisasi ini menggunakan 3 *salesman*. Tabel 4.2 melakukan pembangkitan pada populasi awal.

Tabel 4.2 Populasi Awal

| Individu | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------|----|---|---|---|----|---|----|----|---|----|----|----|----|----|----|
| P1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 4 | 4 | 4 |
| P2 | 3 | 5 | 8 | 1 | 2 | 4 | 11 | 10 | 7 | 9 | 12 | 6 | 4 | 4 | 4 |
| P3 | 12 | 7 | 9 | 5 | 10 | 2 | 1 | 4 | 3 | 6 | 8 | 11 | 4 | 4 | 4 |

Segmen 1

Segmen 2

Segmen 1 pada Tabel 4.2 menunjukkan banyaknya titik lokasi yang dilalui untuk proses pendistribusian beras dan segmen 2 menunjukkan maksimal titik yang ditempuh oleh tiap *salesman*, dimana keberangkatan setiap *salesman* dimulai dari gudang dan kembali ke gudang semula. Terdapat pewarnaan pada nomor tabel yang merupakan perwakilan pada tiap *salesman* pada segmen 1 dan segmen 2 yang masing masing dilalui oleh 3 *salesman*.

4.5.3 Crossover

Pada proses persilangan (*Crossover*) digunakan metode *one-cut-point crossover* dengan memilih dua induk secara acak, selanjutnya dilakukan penentuan titik gen yang akan dijadikan titik potong pada kromosom. Proses *crossover* dilakukan dengan memadukan panjang gen induk pertama bagian kiri dengan panjang gen induk kedua bagian kanan. Pada segmen 1 menunjukkan banyaknya titik lokasi yang dilalui untuk proses pendistribusian beras sebanyak 12 titik dan pada segmen 2 menunjukkan maksimal titik yang ditempuh oleh tiap *salesman*.

Segmen kedua menggunakan nilai yang statis (4 4 4) dikarenakan dalam proses manualisasi menggunakan 12 titik dan jumlah maksimal *salesman* dalam satu rute ada 4 titik, sehingga untuk mendapatkan hasil yang optimal segmen kedua menggunakan nilai yang statis. Proses *crossover* tersebut menghasilkan individu baru atau *child*. Penggunaan *crossover rate* sebesar 0,7 dan banyaknya populasi (*popsize*) yaitu 3, maka akan dihasilkan sekitar 2 *offspring* pada proses *crossover* karena $popsize = 0,7 * 3 = 2,1 = 2$. Individu baru atau *child* yang dihasilkan dapat dilihat pada Tabel 4.3.

Tabel 4.3 Proses *Crossover*

| | | | | | | | | | | | | | | | | | |
|----------|----|---|---|---|----|----|----------------|----|---|----|----|----|---|---|---|--|--|
| | | | | | | | Cut point ↓ | | | | | | | | | | |
| Parent 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 4 | 4 | 4 | | |
| Parent 2 | 3 | 5 | 8 | 1 | 2 | 4 | 11 | 10 | 7 | 9 | 12 | 6 | 4 | 4 | 4 | | |
| Parent 3 | 12 | 7 | 9 | 5 | 10 | 2 | 1 | 4 | 3 | 6 | 8 | 11 | 4 | 4 | 4 | | |
| Child 1 | 1 | 2 | 3 | 4 | 5 | 6 | 12 | 9 | 7 | 11 | 8 | 10 | | | | | |
| Child 2 | 3 | 5 | 8 | 1 | 2 | 11 | 7 | 12 | 4 | 10 | 9 | 6 | | | | | |

4.5.4 Mutation

Mutasi merupakan proses reproduksi setelah persilangan (*Crossover*). Mutasi yang diterapkan pada penelitian ini adalah *reciprocal exchange mutation* dimana melakukan pemilihan pada dua titik gen secara acak diantara titik pertama dan titik kedua, lalu pada kedua titik gen tersebut ditukarkan posisinya. Penggunaan *mutation rate* yaitu 0,3 dan *popsize* sebesar 3, maka terjadi 1 kali proses *mutation* dengan perhitungan $= 0,3 * 3 = 0,9 = 1$. Contoh proses mutasi dapat dilihat pada Tabel 4.4.

Tabel 4.4 Proses *Mutation*

| | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|----|----|---|---|----|---|---|---|---|
| Parent 2 | 3 | 5 | 8 | 1 | 2 | 4 | 11 | 10 | 7 | 9 | 12 | 6 | 4 | 4 | 4 |
| Child 3 | 3 | 5 | 8 | 9 | 2 | 4 | 11 | 10 | 7 | 1 | 12 | 6 | | | |

4.5.5 Evaluasi

Evaluasi digunakan untuk melakukan perhitungan pada nilai *fitness* dari tiap individu pada satu populasi. Dalam perhitungan nilai *fitness* terdapat perhitungan dari nilai konstanta yang dibagi dengan total jarak (km). Setelah didapatkan hasil perhitungan dari total jarak maka dapat dilakukan perhitungan *fitness* dengan menggunakan rumus pada persamaan 2.4.

Parameter yang digunakan yakni total jarak, dimana perhitungan total jarak dimulai dari Gudang menuju Titik Distribusi. Dalam perhitungan manualisasi menggunakan 3 *salesman* untuk melakukan proses pendistribusian beras. Contoh perhitungan pada nilai *fitness* yaitu:

- Kromosom = [1 2 3 4 5 6 7 8 9 10 11 12]
 Jarak Tempuh = $50+46+53+52+14+3,4+6,7+5,6+8+9+49+5,7+44 = 346,4$
- Kromosom = [3 5 8 1 2 4 11 10 7 9 12 6]
 Jarak Tempuh = $56+56+4+49+46+17+46+49+7,3+13+49+45+11 = 448,3$
- Kromosom = [12 7 9 5 10 2 1 4 3 6 8 11]
 Jarak Tempuh = $44+39+13+6,3+5,7+8,4+46+46+52+58+6,1+49+51 = 424,5$

Pada contoh perhitungan manualisasi diatas, perhitungan jarak tempuh diperoleh dari Gudang menuju Titik Distribusi pertama dan selanjutnya sampai Titik Distribusi terakhir dan kembali lagi menuju Gudang awal.

Tabel 4.5 Proses Evaluasi

| Individu | Kromosom | Segmen | Total Jarak | <i>Fitness</i> |
|----------|------------------------------|---------|-------------|----------------|
| P1 | [1 2 3 4 5 6 7 8 9 10 11 12] | [4 4 4] | 346.4 | 0.288 |
| P2 | [3 5 8 1 2 4 11 10 7 9 12 6] | [4 4 4] | 448.3 | 0.223 |
| P3 | [12 7 9 5 10 2 1 4 3 6 8 11] | [4 4 4] | 424.5 | 0.235 |
| C1 | [1 2 3 11 5 6 12 9 7 4 8 10] | [4 4 4] | 350,7 | 0.285 |
| C2 | [3 5 8 1 2 11 7 12 4 10 9 6] | [4 4 4] | 417,4 | 0.239 |
| C3 | [3 5 8 9 2 4 11 10 7 1 12 6] | [4 4 4] | 365,3 | 0.273 |

4.5.6 Seleksi

Seleksi dilakukan untuk memilih individu yang dapat diterima untuk ke generasi selanjutnya. Penggunaan metode *elitism selection* yakni dengan melakukan pengurutan individu dengan nilai *fitness* dari nilai tertinggi ke terendah sebanyak jumlah populasi (*popsiz*e) yang sebelumnya telah ditetapkan. Proses seleksi yaitu:

1. Pengumpulan pada individu *parent* dan *child* serta melakukan pengurutan pada nilai *fitness* tertinggi ke terendah. Pada Tabel 4.6 terjadi proses seleksi untuk pengurutan nilai *Fitness* tertinggi ke terendah.



Tabel 4.6 Proses Seleksi

| Individu | Kromosom | Segmen | Fitness |
|----------|------------------------------|---------|---------|
| P1 | [1 2 3 4 5 6 7 8 9 10 11 12] | [4 4 4] | 0,288 |
| C1 | [1 2 3 11 5 6 12 9 7 4 8 10] | [4 4 4] | 0,285 |
| C3 | [3 5 8 9 2 4 11 10 7 1 12 6] | [4 4 4] | 0,273 |
| C2 | [3 5 8 1 2 11 7 12 4 10 9 6] | [4 4 4] | 0,239 |
| P3 | [12 7 9 5 10 2 1 4 3 6 8 11] | [4 4 4] | 0,235 |
| P2 | [3 5 8 1 2 4 11 10 7 9 12 6] | [4 4 4] | 0,223 |

- Melakukan pemilihan individu terbaik dengan *elitism selection* dan melakukan pengurutan pada nilai *fitness* dan individu terpilih ditetapkan sebagai populasi baru untuk generasi selanjutnya.

Individu terbaik:

| | | | | | | | | | | | | |
|----|---|---|---|---|---|---|----|----|---|---|----|---|
| P1 | 3 | 5 | 8 | 9 | 2 | 4 | 11 | 10 | 7 | 1 | 12 | 6 |
|----|---|---|---|---|---|---|----|----|---|---|----|---|

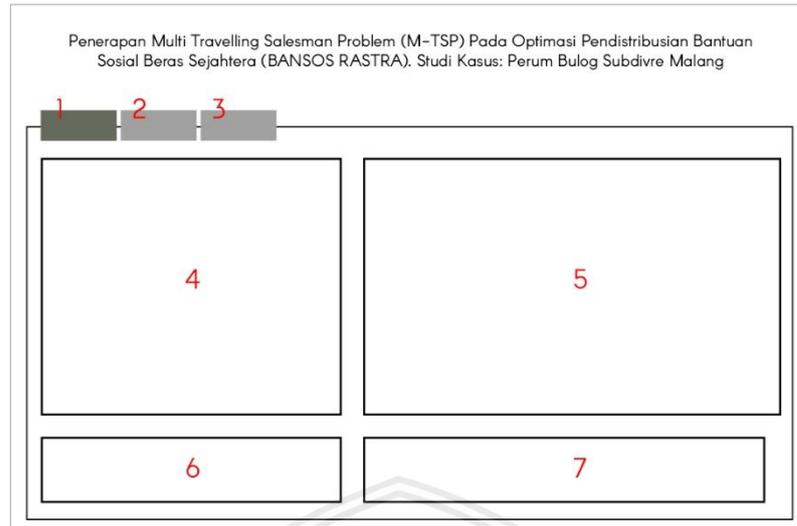
4.6 Perancangan Antarmuka

Perancangan antarmuka dari perangkat lunak untuk permasalahan Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog Subdivre Malang terdiri dari tiga menu yang dibuat dalam tab. Menu pertama berisi halaman *input* data parameter algoritme genetika berupa banyak populasi, banyak jumlah generasi, *crossover rate*, dan *mutation rate*. Perancangan antarmuka diproses seperti yang ditunjukkan pada Gambar 4.11. Pada menu kedua yang ditunjukkan Gambar 4.12 berupa halaman tampilan data jarak yang berisikan jarak yang dibuat dalam table serta menampilkan jarak tersebut melalui *maps* yang diperoleh dari *google maps*. Gambar 4.13 menampilkan menu halaman kromosom terbaik yang terpilih. Menu halaman ini berisikan rincian dari kromosom terbaik mulai dari tabel kromosom terpilih dan perhitungan manual kromosom tersebut.

4.6.1 Perancangan Antarmuka Halaman Utama

Perancangan antarmuka pada halaman utama berisi fungsi utama dari program Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog Subdivre Malang. Fungsi utama tersebut terdiri dari 4 panel yang terdiri dari panel input parameter, panel detail hasil, panel memuat dan memproses data, dan panel untuk menampilkan nilai kromosom dan nilai *fitness*. Pada Gambar 4.9 merupakan rancangan antarmuka halaman utama.





Gambar 4.9 Perancangan Antarmuka Halaman Utama

Keterangan:

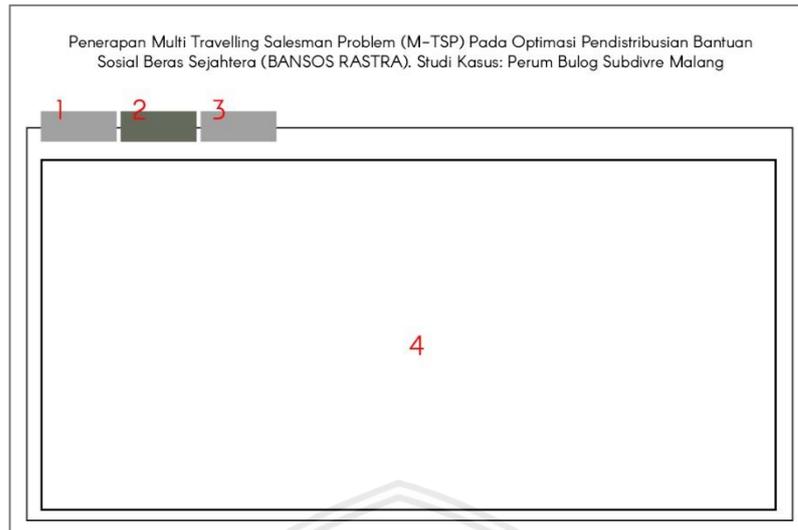
1. Tab menu halaman utama program.
2. Tab menu halaman data jarak pada program.
3. Tab menu halaman detail kromosom terbaik.
4. Kolom *input* data algoritme genetika berupa banyak populasi, jumlah generasi, *crossover rate*, dan *mutation rate*.
5. Kolom detail *output* data hasil proses.
6. Kolom untuk memuat data dan memproses data.
7. Kolom menampilkan nilai kromosom terbaik dan juga nilai *fitness*.

4.6.2 Perancangan Antarmuka Halaman Data Jarak

Perancangan antarmuka pada halaman data jarak berisi fungsi yang memuat tabel data jarak dan menampilkan lokasi jarak dan rute pada program Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog Subdivre Malang. Fungsi antarmuka halaman data jarak terdiri dari 2 panel yang terdiri dari panel tabel data jarak dan panel lokasi rute jarak. Perancangan antarmuka halaman data jarak dapat dilihat pada gambar 4.10.

Keterangan:

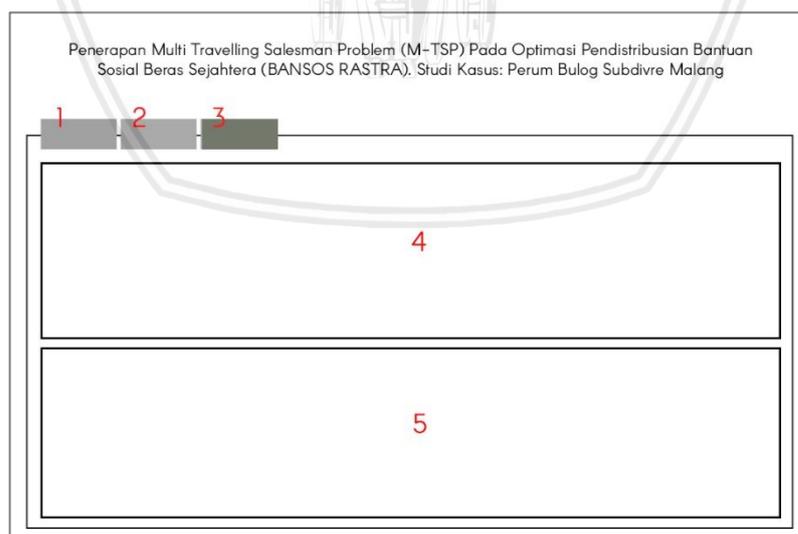
1. Tab menu halaman utama program.
2. Tab menu halaman data jarak pada program.
3. Tab menu halaman detail kromosom terbaik.
4. Kolom berupa tabel yang menampilkan data jarak tiap gudang menuju titik distribusi yang ditempuh.



Gambar 4.10 Halaman Perancangan Antarmuka Data Jarak

4.6.3 Perancangan Antarmuka Halaman Detail Kromosom

Perancangan antarmuka pada halaman detail kromosom berfungsi untuk menampilkan hasil kromosom terbaik dan perhitungan dari kromosom terbaik pada program Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog Subdivre Malang. Fungsi antarmuka halaman detail kromosom terdiri dari 3 panel yang terdiri dari panel hasil kromosom terbaik, panel menampilkan tabel detail dari kromosom terbaik, dan panel tabel perhitungan manual dari kromosom terbaik. Gambar 4.11 merupakan rancangan antarmuka halaman detail kromosom.



Gambar 4.11 Halaman Perancangan Antarmuka Detail Kromosom

Keterangan:

1. Tab menu halaman utama program.
2. Tab menu halaman data jarak pada program.



3. Tab menu halaman detail kromosom terbaik.
4. Kolom untuk menampilkan hasil kromosom terbaik.
5. Kolom berupa tabel untuk menampilkan detail dari kromosom terbaik.

4.7 Perancangan Pengujian dan Evaluasi

Proses perancangan skenario uji coba merupakan evaluasi pada perangkat lunak yang diujikan menggunakan parameter-parameter algoritme genetika. Tujuan pengujian adalah mengetahui peranan pada algoritme genetika terhadap hasil proses optimasi pendistribusian Bansos Rastra. Hasil pengujian yaitu nilai *fitness* terbaik pada suatu individu. Proses pengujian dilakukan pada 4 skenario, setiap skenario melakukan pengujian sebanyak kebutuhan tiap pengujian lalu diperoleh rata-rata nilai *fitness* pada tiap skenario uji. Skenario pengujian diantaranya:

1. Pengujian konvergensi untuk mendapatkan nilai *fitness* terbaik yang didapatkan setelah dilakukan proses pencarian nilai yang ideal dengan besar iterasi maksimum.
2. Pengujian menentukan jumlah populasi optimal dalam penerapan algoritme genetika untuk optimasi distribusi Bansos Rastra.
3. Pengujian kombinasi nilai *mutation rate* dan *crossover rate* untuk penyelesaian permasalahan *m-TSP* pada optimasi distribusi Bansos Rastra.
4. Pengujian Akhir yaitu Pengujian Validasi dimana pada pengujian ini dilakukan pengambilan nilai *fitness* terbaik dari parameter yang telah diujikan. Pengujian validasi dilakukan sebanyak 5 kali pengujian agar benar-benar mendapatkan nilai *fitness* yang optimal.

4.7.1 Perancangan Pengujian Konvergensi

Pengujian konvergensi dilakukan untuk mendapatkan nilai *fitness* terbaik, dimana pengujian ini dilakukan pencarian nilai yang ideal dengan besar generasi atau iterasi maksimum. Jika pengujian telah mendapatkan nilai konvergensi yang berarti selisih nilai *fitness offspring* dan nilai *fitness* induk tidak terlalu jauh maka kondisi berhenti dapat dilakukan. Rancangan pengujian konvergensi dapat dilihat pada Tabel 4.7. Pengujian dilakukan sejumlah 10 kali pengujian dan jumlah generasi dimulai dari 1 hingga 900.

Tabel 4.7 Rancangan Pengujian Konvergensi

| Generasi | Nilai <i>Fitness</i> | | | | | Rata-rata Nilai <i>Fitness</i> |
|----------|----------------------|---|---|---|---|--------------------------------|
| | Pengujian | | | | | |
| | 1 | 2 | 3 | 4 | 5 | |
| 1 | | | | | | |
| 10 | | | | | | |
| 40 | | | | | | |
| 60 | | | | | | |



| | | | | | | | | | | |
|-----|--|--|--|--|--|--|--|--|--|--|
| 80 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| 300 | | | | | | | | | | |
| 500 | | | | | | | | | | |
| 700 | | | | | | | | | | |
| 900 | | | | | | | | | | |

4.7.2 Perancangan Pengujian Parameter Berdasarkan Jumlah Populasi

Rancangan pengujian dengan parameter jumlah populasi (*popsiz*) bertujuan mengetahui ukuran *popsiz* terbaik dan menjadi parameter pada ukuran *popsiz*. Rentang pengujian *popsiz* dimulai dari 10 sampai populasi ke 100 dengan kondisi 10 kali pengujian pada tiap ukuran populasi (*popsiz*). Tabel 4.8 merupakan rancangan uji coba ukuran populasi (*popsiz*).

Tabel 4.8 Rancangan Uji Coba Ukuran *Popsiz*

| <i>Popsiz</i> | Nilai <i>Fitness</i> | | | | | | | | | | Rata-rata Nilai <i>Fitness</i> |
|---------------|----------------------|---|---|---|---|---|---|---|---|----|--------------------------------|
| | Pengujian | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 | | | | | | | | | | | |
| 20 | | | | | | | | | | | |
| 30 | | | | | | | | | | | |
| 40 | | | | | | | | | | | |
| 50 | | | | | | | | | | | |
| 60 | | | | | | | | | | | |
| 70 | | | | | | | | | | | |
| 80 | | | | | | | | | | | |
| 90 | | | | | | | | | | | |
| 100 | | | | | | | | | | | |

4.7.3 Perancangan Pengujian Parameter Berdasarkan Kombinasi Nilai *Cr* dan *Mr*

Rancangan uji coba pada kombinasi nilai *Cr* dan *Mr* memiliki tujuan yaitu menemukan nilai kombinasi *Cr* dan *Mr* yang paling tepat dalam pemenuhan parameter serta pengaruh pada kombinasi nilai *Cr* dan *Mr* terhadap nilai *fitness* yang dihasilkan. Kombinasi *Cr* dan *Mr* merupakan bilangan acak 0 hingga 1.



Pengujian pada kombinasi *Cr* dan *Mr* melakukan 10 kali pengujian. tabel 4.10 merupakan rancangan uji coba kombinasi nilai *Cr* dan *Mr*.

Tabel 4.9 Rancangan Uji Coba *Cr* dan *Mr*

| <i>Cr; Mr</i> | Nilai <i>Fitness</i> | | | | | | | | | | Rata-rata Nilai <i>Fitness</i> |
|---------------|----------------------|---|---|---|---|---|---|---|---|----|--------------------------------|
| | Pengujian | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 0,1;0,9 | | | | | | | | | | | |
| 0,2;0,8 | | | | | | | | | | | |
| 0,3;0,7 | | | | | | | | | | | |
| 0,4;0,6 | | | | | | | | | | | |
| 0,5;0,5 | | | | | | | | | | | |
| 0,6;0,4 | | | | | | | | | | | |
| 0,7;0,3 | | | | | | | | | | | |
| 0,8;0,2 | | | | | | | | | | | |
| 0,9;0,1 | | | | | | | | | | | |

4.7.4 Perancangan Pengujian Akhir atau Validasi

Perancangan pengujian akhir atau pengujian validasi dilakukan untuk mendapatkan nilai optimal dari individu terbaik. Perancangan pengujian dilakukan dengan mengambil individu terbaik dari nilai *fitness* dari pengujian parameter yang telah diujikan sebelumnya. Pengujian ini dilakukan sebanyak 5 kali pengujian untuk mendapatkan nilai yang benar-benar optimal.

Tabel 4.10 Rancangan Pengujian Validasi

| Individu Terbaik | Nilai <i>Fitness</i> | | | | | <i>Fitness</i> Terbaik |
|------------------|----------------------|---|---|---|---|------------------------|
| | Pengujian | | | | | |
| | 1 | 2 | 3 | 4 | 5 | |
| 1 | | | | | | |

BAB 5 IMPLEMENTASI

Pada bab implementasi merupakan tahap penerapan sistem, dimana aplikasi siap dijalankan pada keadaan sebenarnya.

5.1 Lingkungan Implementasi

Lingkungan impleementasi merupakan ruang lingkup kebutuhan dalam sistem Algoritme Genetika pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra). Terdapat dua lingkungan yang digunakan dalam membangun sistem aplikasi tersebut, yakni lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam penerapan aplikasi algoritme genetika pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) adalah sebagai berikut:

1. *Processor*: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz.
2. *Memory* (RAM): 16.0 GB.
3. *Harddisk* dengan kapasitas 500 GB.
4. *Monitor* 14 inch.

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam penerapan aplikasi algoritme genetika pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) adalah sebagai berikut:

1. Sistem Operasi yang digunakan adalah Windows 10 Pro 64-bit.
2. *Tools* yang digunakan adalah JDK1.8.
3. *Editor* yang digunakan adalah Netbeans IDE 8.2.

5.2 Batasan Implementasi

Adapun batasan yang dibuat dalam upaya implementasi sistem yaitu:

1. Proses implementasi sistem pada penelitian menggunakan bahasa pemrograman java.
2. Data disimpan dalam format *excel* untuk proses implementasi.
3. Metode yang digunakan dalam Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) adalah *Multi Travelling Salesman Problem* (*m-TSP*) pada Algoritme Genetika.

5.3 Implementasi Algoritme

Implementasi algoritme menjabarkan kode program dari implementasi sistem. Implementasi meliputi inialisasi parameter berupa pembangkitan individu awal, *Crossover rate (Cr)*, dan *Mutation rate (Mr)*.

5.3.1 Implementasi Membangkitkan Individu Awal

Membangkitkan populasi awal dimulai dengan membangkitkan individu. Suatu individu diperoleh dengan cara random sebanyak panjang. Proses membangkitkan individu dilakukan pengecekan pada setiap gen agar tidak terdapat nilai yang sama dalam satu kromosom. Pada kode program 5.1 merupakan proses implementasi membangkitkan populasi awal.

| Source code: Algen.java | |
|-------------------------|---|
| 1 | public void inialisasiKromosom(){ |
| 2 | Random rand = new Random(); |
| 3 | for(int i = 0; i < this.popSize; i++){ |
| 4 | ArrayList<Integer> nilaiRandomTerpakai = new ArrayList<>(); |
| 5 | for(int j = 0; j < this.JUMLAH_KROMOSOM; j++){ |
| | int temp; |
| 6 | if(nilaiRandomTerpakai.isEmpty()){ |
| 7 | temp = 0 + (int) (Math.random() * |
| | (((this.JUMLAH_KROMOSOM - 1) - 0) + 1)); |
| | } else { |
| 8 | temp = randomDenganPengecualian(rand, 0, |
| | this.JUMLAH_KROMOSOM - 1, nilaiRandomTerpakai); |
| | } |
| 9 | nilaiRandomTerpakai.add(temp); |
| 10 | Collections.sort(nilaiRandomTerpakai); |
| | this.p[i][j] = temp; |
| | } |
| 11 | nilaiRandomTerpakai.clear(); |
| | } |
| | } |

Kode Program 5.1 Implementasi Membangkitkan Populasi Awal

Deskripsi pada kode program 5.1, baris ke-:

4. Melakukan inialisasi variable nilaiRandomTerpakai ke dalam arraylist bertipe integer.
5. Perulangan yang berfungsi untuk melakukan pembuatan individu dengan panjang kromosom.
6. Pembuatan kondisi untuk mencegah terdapatnya nilai random yang sama.
7. Menyimpan nilai kromosom dalam variable temp.
8. Variable temp menginisialisasi fungsi randomDenganPengecualian untuk menjaga nilai random pada kromosom.
9. Objek nilaiRandomTerpakai ditambahkan pada variable temp.
10. Mengurutkan objek nilaiRandomTerpakai.
11. Fungsi clear untuk mengosongkan kembali dari stack objek bernama nilaiRandomTerpakai.

5.3.2 Implementasi Proses Crossover

Metode *Crossover* menggunakan *one-cut-point crossover*. Proses pemilihan induk yang akan mengalami *crossover* disesuaikan dengan penentuan nilai *crossover rate (cr)*. Kode program 5.2 merupakan implementasi proses *crossover*.

```
Source code: Algen.java
1 public void crossover(){
    Random rand = new Random();
    int mulaiBaru = 1;
2     int individu1 = 0 + (int) (Math.random() * ((this.popSize
    - 0)));
3     ArrayList<Integer> individuTerpilih = new ArrayList<>();
    individuTerpilih.add(individu1);
4     int individu2 = randomDenganPengecualian(rand, 0, 2,
    individuTerpilih);
5     int titikPotong = 0 + (int) (Math.random() *
    (((this.JUMLAH_KROMOSOM - 1) - 0) + 1));
6     for (int i = 0; i < this.c.length; i++) {
7         System.out.println("Individu1:      "+individu1+"
    individu2:      "+individu2+"          titikPotong:
    "+titikPotong);
            if(i % 2 == 0){
                this.c[i]= crossoverIndividu
                    (Arrays.copyOfRange(this.p[individu1],      0,
                    this.JUMLAH_KROMOSOM),
                    Arrays.copyOfRange(this.p[individu2],      0,
                    this.JUMLAH_KROMOSOM), titikPotong);
            }else{
                this.c[i]= crossoverIndividu
                    (Arrays.copyOfRange(this.p[individu2],      0,
                    this.JUMLAH_KROMOSOM),
                    Arrays.copyOfRange(this.p[individu1],      0,
                    this.JUMLAH_KROMOSOM), titikPotong);
            }
8         if(mulaiBaru == 2){
9             individuTerpilih.clear();
10            mulaiBaru = 1;
11            individu1 = 0 + (int) (Math.random() *
                ((this.popSize - 0)));
12            individuTerpilih.add(individu1);
13            individu2 = randomDenganPengecualian(rand, 0, 2,
                individuTerpilih);
14            titikPotong = 0 + (int) (Math.random() *
                (((this.JUMLAH_KROMOSOM - 1) - 0) + 1));
            }else{
15                mulaiBaru++;
            }
        }
    }
}
```

Kode Program 5.2 Implementasi Proses Crossover

Deskripsi pada kode program 5.2, baris ke-:

2. Memilih individu pertama secara acak.
3. Melakukan inisialisasi variable *individuTerpilih* ke dalam *arraylist* bertipe *integer*.
4. Menambahkan individu pertama ke dalam objek *individuTerpilih* dan memilih individu kedua secara acak tanpa mendapatkan individu pertama lagi.
5. Memilih titik potong yang akan dibuat untuk proses *crossover*.
8. Memulai baru proses *crossover* dengan individu baru.

5.3.3 Implementasi Proses Mutasi

Metode mutasi menggunakan *reciprocal exchange mutation*. Penentuan induk dilakukan sebelum proses mutasi dilakukan yang selanjutnya mengalami proses mutasi sesuai dengan nilai *mutation rate*. Kode program 5.3 merupakan implementasi proses mutasi.

| Source code: Algen.java | |
|-------------------------|--|
| 1 | public void mutasi(){ |
| 2 | Random rand = new Random(); |
| 3 | for (int i = 0; i < this.m.length; i++) { |
| | int individu = 0 + (int) (Math.random() * ((this.popSize - 0))); |
| 4 | int gen1 = 0 + (int) (Math.random() * ((this.JUMLAH_KROMOSOM - 1) - 0) + 1)); |
| 5 | ArrayList<Integer> genTerpilih = new ArrayList<>(); |
| 6 | genTerpilih.add(gen1); |
| 7 | int gen2 = randomDenganPengecualian(rand, 0, this.JUMLAH_KROMOSOM - 1, genTerpilih); |
| 8 | this.m[i] = Arrays.copyOfRange(this.p[individu], 0, this.p[individu].length); |
| | double temp = this.m[i][gen1]; |
| 9 | this.m[i][gen1] = this.m[i][gen2]; |
| 10 | this.m[i][gen2] = temp; |
| | } |
| | } |

Kode Program 5.3 Implementasi Proses Mutasi

Deskripsi pada kode program 5.3, baris ke-:

3. Melakukan perulangan dengan kondisi yang digunakan untuk memilih individu secara acak.
4. Memilih gen pertama secara acak.
5. Inisialisasi objek genTerpilih yang disimpan pada arraylist.
6. Memasukkan gen pertama pada objek genTerpilih.
7. Memilih gen kedua secara acak dengan syarat kecuali gen pertama.
8. Menyalin individu terpilih.
9. Proses menukar gen pertama dan gen kedua.

5.3.4 Implementasi Perhitungan Nilai *Fitness*

Nilai terbaik atau nilai *fitness* didapatkan dari konstanta dibagi dengan total jarak kesetiap tujuan yang dimulai dari kantor pusat dalam hal ini gudang hingga tujuan akhir dan kembali lagi ke gudang. Perhitungan nilai *fitness* dilakukan untuk tiap proses yang bekerja dalam kromosom. Proses perhitungan *fitness* dapat dilihat pada kode program 5.4 dan kode program 5.5.

| Source code: Algen.java | |
|-------------------------|---|
| 1 | public double hitungFitness(double arr[]) { |
| 2 | double totalJarak = hitungJarak(arr); |
| 3 | return 100 / totalJarak; |
| | } |

Kode Program 5.4 Implementasi Perhitungan Nilai *Fitness*

Deskripsi pada kode program 5.4, baris ke-:

2. Deklarasi variabel totalJarak dengan memanggil fungsi hitungJarak.

3. Mengembalikan fungsi hitungFitness dengan rumus $100/\text{totalJarak}$.

```
Source code: Algen.java
1 public double hitungJarak(double arr[]) {
2     double totalJarak = this.dataGBBGadang[0][((int) arr[0]) + 1];
3     for (int i = 0; i < this.JUMLAH_TITIK - 2; i++) {
4         totalJarak += this.dataGBBGadang[((int) arr[i]) + 1][((int)
5             arr[i + 1]) + 1];
6     }
7     totalJarak += this.dataGBBGadang[((int) arr[this.JUMLAH_TITIK -
8         2]) + 1][0];
9     return totalJarak;
10 }
```

Kode Program 5.5 Implementasi Perhitungan Total Jarak

Deskripsi pada kode program 5.5, baris ke-:

2. Menghitung nilai variabel totalJarak yang didapat dari dataset.
3. Melakukan perulangan dengan kondisi pada variable JUMLAH_TITIK dikurangi 2 dan melakukan penjumlahan pada variable totalJarak.

5.3.5 Implementasi Proses Evaluasi

Evaluasi dilakukan untuk mendapatkan nilai *fitness* pada individu dalam populasi. Kode program 5.6 merupakan implementasi proses evaluasi.

```
Source code: Algen.java
1 public void evaluasi(){
2     for (int i = 0; i < this.p.length; i++) {
3         for (int j = 0; j < this.p[i].length; j++) {
4             this.ps[i][j] = this.p[i][j];
5         }
6     }
7     for (int i = 0; i < this.c.length; i++) {
8         for (int j = 0; j < this.p[i].length; j++) {
9             this.ps[i + this.p.length][j] = this.c[i][j];
10        }
11    }
12    for (int i = 0; i < this.m.length; i++) {
13        for (int j = 0; j < this.p[i].length; j++) {
14            this.ps[i + this.p.length + this.c.length][j] =
15                this.m[i][j];
16        }
17    }
18    for (int i = 0; i < this.ps.length; i++) {
19        double fitness = hitungFitness(this.ps[i]);
20        this.ps[i][this.JUMLAH_KROMOSOM] = fitness;
21    }
22 }
```

Kode Program 5.6 Implementasi Proses Evaluasi

Deskripsi pada kode program 5.6, baris ke-:

2. Menggabungkan individu *parent* dengan *child*.
11. Menghitung fitness yang memanggil fungsi hitungFitness.

5.3.6 Implementasi Proses Seleksi

Implementasi proses seleksi digunakan dalam hal pemilihan individu terbaik untuk generasi berikutnya menggunakan metode seleksi *elitism*. Pada seleksi ini,

pemilihan individu yang merupakan individu dengan nilai *fitness* terbaik 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. Implementasi proses seleksi dapat dilihat pada kode program 5.7.

| Source code: Algen.java | |
|-------------------------|--|
| 1 | public void seleksi(){ |
| 2 | for (int g = 0; g < this.JUMLAH_GUDANG; g++) { |
| 3 | double tempSeleksi[][] = new |
| | double[this.ps[g].length][this.ps[g][0].length]; |
| 4 | double hasilSeleksi[][] = new |
| | double[this.popSize][this.ps[g][0].length]; |
| 5 | System.arraycopy(this.ps[g], 0, tempSeleksi, 0, |
| | this.ps[g].length); |
| 6 | tempSeleksi = bubbleSort(tempSeleksi); |
| 7 | this.ps[g] = tempSeleksi; |
| 8 | System.arraycopy(tempSeleksi, 0, hasilSeleksi, 0, |
| | this.popSize); |
| | this.p[g] = hasilSeleksi; |
| | } |
| 9 | double a = this.ps[0][0][this.JUMLAH_KROMOSOM]; |
| 10 | for (int i = 0; i < this.ps[0].length; i++) { |
| | this.fitnessTotal[i] = this.ps[0][i][this.JUMLAH_KROMOSOM] + |
| | this.ps[1][i][this.JUMLAH_KROMOSOM] + |
| | this.ps[2][i][this.JUMLAH_KROMOSOM] + |
| | this.ps[3][i][this.JUMLAH_KROMOSOM]; |
| | } |
| | } |

Kode Program 5.7 Implementasi Proses Seleksi

Deskripsi pada kode program 5.7, baris ke-

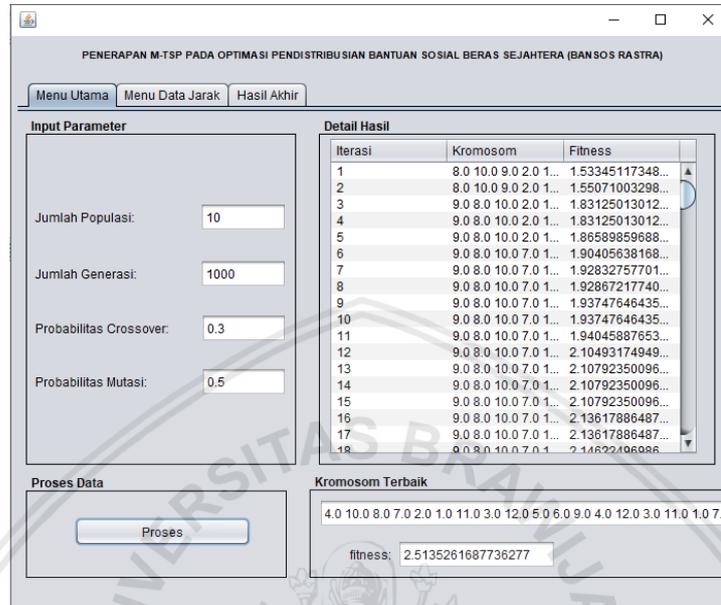
2. Perulangan yang dibuat untuk menampilkan data pada JUMLAH_GUDANG.
3. Melakukan inisialisasi variabel array multidimensi tempSeleksi yang berfungsi untuk menampung populasi.
4. Melakukan inisialisasi variabel array multidimensi hasilSeleksi yang berfungsi untuk menampung *popsize*.
5. Menyalin variabel tempSeleksi.
6. Melakukan seleksi menggunakan metode *bubble sort* pada variabel tempSeleksi dengan memanggil fungsi bubbleSort.
8. Menyalin variabel pada tempSeleksi dan hasilSeleksi.
9. Menyimpan nilai JUMLAH_KROMOSOM pada variabel a.
10. Melakukan proses pengulangan untuk perhitungan *fitness* pada pengujian.

5.4 Implementasi Antarmuka (*User Interface*)

Implementasi antarmuka (*User Interface*) dari sistem Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) terdiri dari tiga antarmuka halaman yakni halaman utama, halaman data jarak, dan halaman hasil akhir.

5.4.1 Implementasi Antarmuka Halaman Utama

Halaman utama berisi input data parameter algoritme genetika yang berfungsi melakukan perhitungan. Gambar 5.1 merupakan implementasi antarmuka halaman utama.



Gambar 5.1 Implementasi Antarmuka Halaman Utama

Pada halaman utama sistem setelah user memasukkan data input parameter, selanjutnya menekan tombol proses pada panel Proses Data agar inputan parameter selanjutnya dilakukan komputasi untuk mendapatkan hasil perhitungan yang ditampilkan pada panel Detail Hasil. Panel Kromosom Terbaik menampilkan individu terbaik dengan nilai *fitness* tertinggi.

5.4.2 Implementasi Antarmuka Halaman Data Jarak

Pada halaman menu data jarak berisi jarak rute dari tiap gudang menuju titik distribusi yang ditempuh. Pada sistem Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) mengambil empat lokasi gudang, yaitu Gudang Beras Bulog (GBB) Gadang, GBB Kebonagung, GBB Gadingrejo, dan GBB Kejapanan. Adapun titik distribusi yang ditempuh yaitu 12 titik tiap gudangnya. Gambar 5.2 merupakan implementasi antarmuka halaman data jarak.

| Nama | GBB... | Banj... | Bulu... | Don... | Jati... | Jam... | Kas... | Kre... | Se... | Taji... | Tam... | Tem... | Tlog... |
|---------|--------|---------|---------|--------|---------|--------|--------|--------|-------|---------|--------|--------|---------|
| GBB... | 0.0 | 50.0 | 12.0 | 56.0 | 9.4 | 9.2 | 11.0 | 11.0 | 5.9 | 8.1 | 4.9 | 51.0 | 44.0 |
| Banj... | 50.0 | 0.0 | 46.0 | 17.0 | 46.0 | 50.0 | 52.0 | 45.0 | 49.0 | 56.0 | 49.0 | 11.0 | 10.0 |
| Bulu... | 12.0 | 46.0 | 0.0 | 53.0 | 17.0 | 6.1 | 5.3 | 2.8 | 6.8 | 11.0 | 8.4 | 47.0 | 40.0 |
| Don... | 56.0 | 17.0 | 53.0 | 0.0 | 52.0 | 56.0 | 58.0 | 52.0 | 55.0 | 62.0 | 55.0 | 7.1 | 11.0 |
| Jati... | 9.4 | 46.0 | 17.0 | 52.0 | 0.0 | 14.0 | 16.0 | 16.0 | 11.0 | 17.0 | 9.8 | 46.0 | 39.0 |
| Jam... | 9.2 | 50.0 | 6.1 | 56.0 | 14.0 | 0.0 | 3.4 | 6.6 | 4.0 | 6.3 | 5.7 | 50.0 | 43.0 |
| Kas... | 11.0 | 52.0 | 5.3 | 58.0 | 16.0 | 3.4 | 0.0 | 6.7 | 6.1 | 5.6 | 7.7 | 52.0 | 45.0 |
| Kre... | 11.0 | 45.0 | 2.8 | 52.0 | 16.0 | 6.6 | 6.7 | 0.0 | 5.6 | 13.0 | 7.3 | 46.0 | 39.0 |
| Se... | 5.9 | 49.0 | 6.8 | 55.0 | 11.0 | 4.0 | 6.1 | 5.6 | 0.0 | 8.0 | 2.3 | 49.0 | 42.0 |
| Taji... | 8.1 | 56.0 | 11.0 | 62.0 | 17.0 | 6.3 | 5.6 | 13.0 | 8.0 | 0.0 | 9.0 | 56.0 | 49.0 |
| Tam... | 4.9 | 49.0 | 8.4 | 55.0 | 9.8 | 5.7 | 7.7 | 7.3 | 2.3 | 9.0 | 0.0 | 49.0 | 42.0 |
| Tem... | 51.0 | 11.0 | 47.0 | 7.1 | 46.0 | 50.0 | 52.0 | 46.0 | 49.0 | 56.0 | 49.0 | 0.0 | 5.7 |
| Tlog... | 44.0 | 10.0 | 40.0 | 11.0 | 39.0 | 43.0 | 45.0 | 39.0 | 42.0 | 49.0 | 42.0 | 5.7 | 0.0 |

Gambar 5.2 Implementasi Halaman Antarmuka Data Jarak

5.4.3 Implementasi Antarmuka Halaman Data Hasil Akhir

Dalam antarmuka halaman data hasil akhir pada sistem Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) berisi hasil optimasi rute terbaik pada tiap gudangnya serta menampilkan selisih jarak yang diperoleh dari jarak prediksi program dengan jarak aktual. Implementasi halaman data hasil akhir dapat dilihat pada Gambar 5.3.

Rute Terbaik Tiap Gudang

Rute terbaik gudang GBB Gadang:
 KP GBB Gadang → Jatisari → Tambaksari → Sempalwadak → Krebet → Bululawang → Banjarejo → Tempursari

Rute terbaik gudang GBB Kebonagung:
 KP GBB Kebonagung → Segaran → Kidangbang → Sumberejo → Patokpilis → Gedangan → Karangkalas → Tum

Rute terbaik gudang GBB Gadingrejo:
 KP GBB Gadingrejo → Baturetno → Langlang → Randuagung → Losari → Candirenggo → Watugede → Gunungrejo

Rute terbaik gudang GBB Kejapanan:
 KP GBB Kejapanan → Sngading → Wonorejo → Bedali → Sidoluhur → Ketindan → Mulyoarjo → Turirejo → Lawang

| No | Gudang | Jarak Aktual | Jarak Program | Selisih |
|----|----------------|--------------|----------------|---------------|
| 1 | GBB Gadang | 437.1 | 165.1 | 272.0 |
| 2 | GBB Kebona... | 456.7 | 157.4 | 299.299999... |
| 3 | GBB Gadingr... | 171.4 | 160.8 | 10.5999999... |
| 4 | GBB Kejapa... | 180.8 | 153.7000000... | 27.0999999... |

Gambar 5.3 Implementasi Antarmuka Halaman Data Hasil Akhir

BAB 6 PENGUJIAN DAN ANALISIS

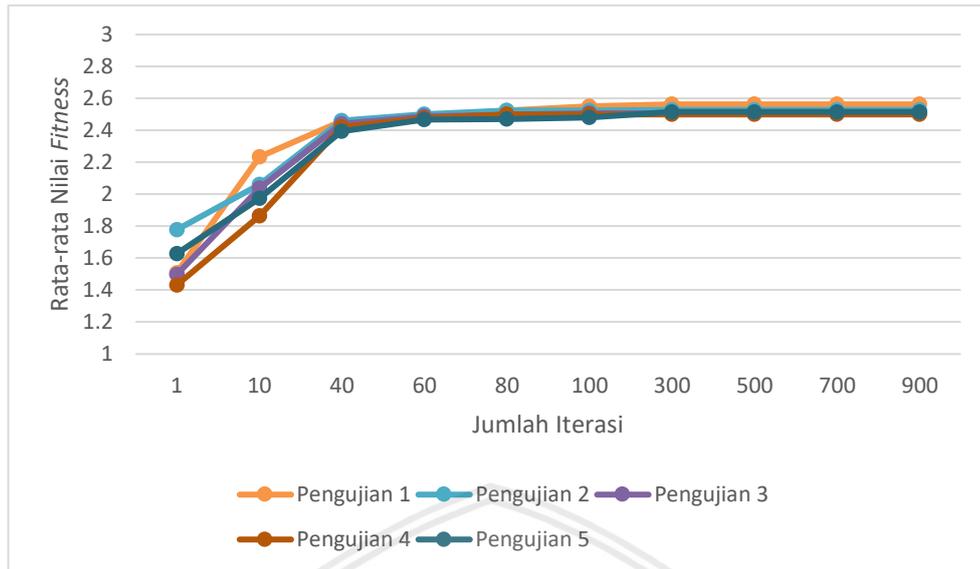
Pembahasan pada pengujian dan analisis yakni terkait proses-proses pengujian pada sistem “Penerapan Multi Travelling Salesman Problem Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Malang. Proses pengujian melakukan pengujian dari perancangan pengujian yang dibuat. Pengujian dilakukan untuk memperoleh hasil dari evaluasi dari sistem berdasarkan implementasi metode yang diterapkan.

6.1 Pengujian Konvergensi

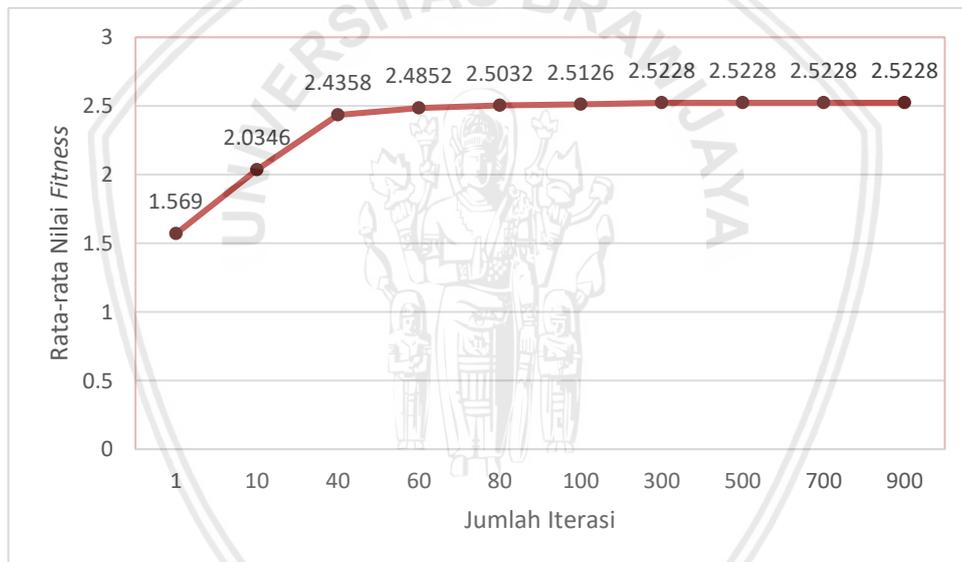
Pengujian konvergensi dilakukan untuk mengetahui nilai *fitness* yang paling optimal pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra). Pengujian ini dilakukan dengan menjalankan program dengan besar generasi atau iterasi maksimum. Jika nilai *fitness offspring* memiliki selisih nilai tidak terlalu jauh dengan nilai *fitness* induk maka dapat dikatakan telah mendapatkan nilai yang konvergen. Proses pengujian menggunakan metode *elitism*. Pengujian ini dilakukan sebanyak 5 kali percobaan dengan jumlah generasi yang dimulai dari 1 hingga 900 generasi. Adapun parameter yang diujikan yakni 10 populasi dengan nilai Cr 0,3 dan nilai Mr 0,5. Pengujian konvergensi dapat dilakukan pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Konvergensi

| Generasi | Nilai <i>Fitness</i> | | | | | Rata-rata Nilai <i>Fitness</i> |
|----------|----------------------|-------|-------|-------|-------|--------------------------------|
| | Pengujian | | | | | |
| | 1 | 2 | 3 | 4 | 5 | |
| 1 | 1,509 | 1,778 | 1,497 | 1,432 | 1,628 | 1,569 |
| 10 | 2,234 | 2,062 | 2,038 | 1,865 | 1,974 | 2,035 |
| 40 | 2,458 | 2,462 | 2,442 | 2,422 | 2,395 | 2,436 |
| 60 | 2,489 | 2,502 | 2,488 | 2,478 | 2,469 | 2,485 |
| 80 | 2,523 | 2,525 | 2,498 | 2,500 | 2,470 | 2,503 |
| 100 | 2,551 | 2,525 | 2,507 | 2,500 | 2,480 | 2,513 |
| 300 | 2,563 | 2,528 | 2,507 | 2,500 | 2,516 | 2,523 |
| 500 | 2,563 | 2,528 | 2,507 | 2,500 | 2,516 | 2,523 |
| 700 | 2,563 | 2,528 | 2,507 | 2,500 | 2,516 | 2,523 |
| 900 | 2,563 | 2,528 | 2,507 | 2,500 | 2,516 | 2,523 |



Gambar 6.1 Grafik Hasil Pengujian Konvergensi



Gambar 6.2 Grafik Hasil Rata-rata Nilai *Fitness* Pengujian Konvergensi

Pada Gambar 6.2 nilai *fitness* menyatakan tidak mengalami perbedaan secara signifikan mulai generasi ke-60. Perbedaan warna pada Gambar 6.1 menunjukkan bahwa pengujian konvergensi dilakukan sebanyak 5 kali pengujian. Sedangkan nilai rata-rata *fitness* ditampilkan pada Gambar 6.2.

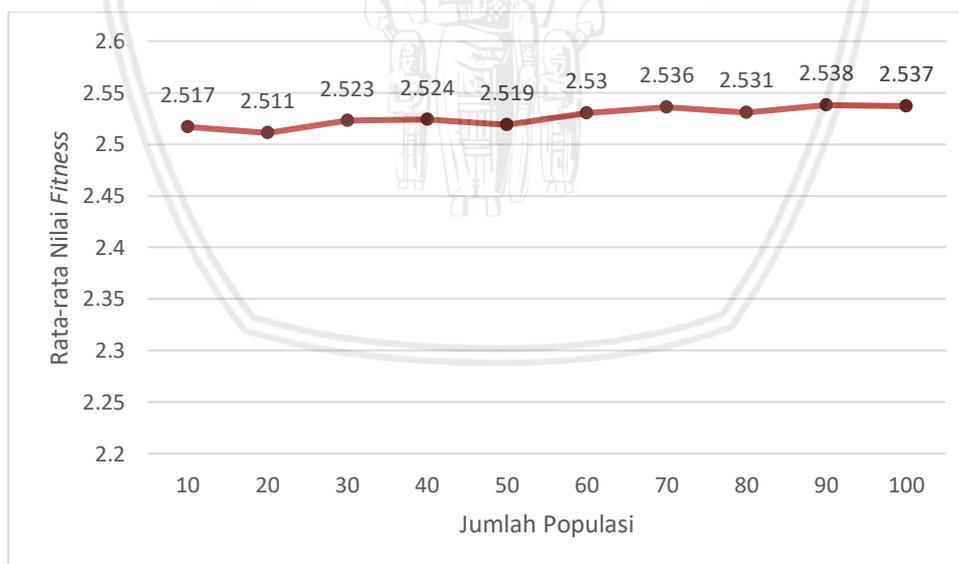
6.2 Pengujian Parameter Berdasarkan Jumlah Populasi

Pengujian pada parameter jumlah populasi dilakukan dalam upaya menemukan banyak populasi dan solusi terbaik untuk Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra). Pengujian dilakukan sebanyak 10 kali pengujian dengan mengambil generasi terbaik pada pengujian konvergensi yaitu pada iterasi ke-300. Nilai kombinasi C_r yakni 0,3 dan M_r 0,5. Ukuran populasi yang diujikan kelipatan

10, yaitu mulai 10 hingga populasi ke-100. Pengujian dengan parameter jumlah populasi menggunakan metode *elitism* sehingga memperoleh nilai rata-rata *fitness* agar mengetahui solusi terbaik dan ukuran populasi yang optimal. Tabel 6.2 merupakan tabel hasil percobaan ukuran populasi.

Tabel 6.2 Hasil Uji Coba Jumlah Populasi

| Popsize | Nilai <i>Fitness</i> | | | | | | | | | | Rata-rata nilai <i>Fitness</i> |
|---------|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------------------------------|
| | Pengujian ke | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 | 2.524 | 2.522 | 2.532 | 2.524 | 2.553 | 2.494 | 2.467 | 2.515 | 2.532 | 2.501 | 2.517 |
| 20 | 2.544 | 2.505 | 2.490 | 2.493 | 2.521 | 2.503 | 2.560 | 2.468 | 2.535 | 2.487 | 2.511 |
| 30 | 2.480 | 2.488 | 2.514 | 2.545 | 2.514 | 2.543 | 2.557 | 2.558 | 2.523 | 2.504 | 2.523 |
| 40 | 2.537 | 2.572 | 2.508 | 2.514 | 2.470 | 2.517 | 2.517 | 2.546 | 2.514 | 2.544 | 2.524 |
| 50 | 2.554 | 2.501 | 2.514 | 2.425 | 2.517 | 2.530 | 2.501 | 2.552 | 2.535 | 2.563 | 2.519 |
| 60 | 2.561 | 2.549 | 2.529 | 2.484 | 2.523 | 2.535 | 2.532 | 2.546 | 2.537 | 2.503 | 2.530 |
| 70 | 2.565 | 2.545 | 2.513 | 2.527 | 2.533 | 2.550 | 2.544 | 2.546 | 2.529 | 2.512 | 2.536 |
| 80 | 2.602 | 2.543 | 2.504 | 2.554 | 2.484 | 2.512 | 2.514 | 2.520 | 2.547 | 2.529 | 2.531 |
| 90 | 2.479 | 2.539 | 2.516 | 2.534 | 2.560 | 2.570 | 2.514 | 2.559 | 2.548 | 2.559 | 2.538 |
| 100 | 2.563 | 2.540 | 2.558 | 2.530 | 2.504 | 2.514 | 2.505 | 2.573 | 2.549 | 2.538 | 2.537 |



Gambar 6.3 Grafik Hasil Uji Coba Banyak Populasi

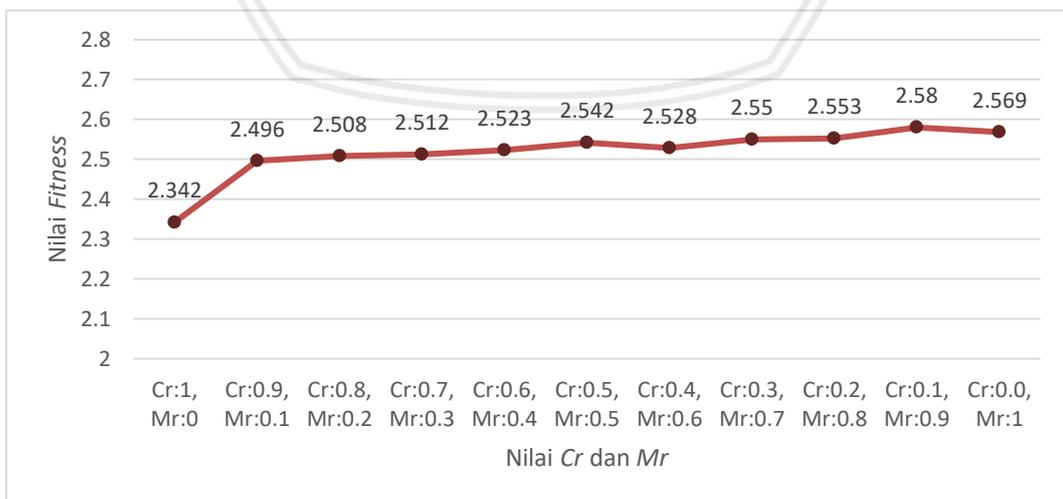
Pada Gambar 6.3 berisi grafik hasil pengujian untuk jumlah populasi pada 10 kali percobaan. Setiap percobaan, nilai *fitness* tidak mengalami nilai yang naik turun dalam arti nilai *fitness* pada uji coba jumlah populasi ini konvergen. Dari hasil grafik permasalahan pada Gambar 6.3 menunjukkan bahwa jumlah populasi optimal ditunjukkan pada populasi ke 90.

6.3 Pengujian Parameter Berdasarkan Kombinasi Nilai Cr dan Mr

Pengujian parameter kombinasi nilai Cr dan Mr dilakukan untuk mengukur nilai kombinasi yang paling tepat pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra). Jumlah Pengujian yakni 10 kali pengujian untuk nilai generasi 300 dan ukuran populasi terbaik pada populasi 90. Pengujian menerapkan metode *elitism* dimana akan mendapatkan nilai rata-rata *fitness* untuk mengetahui nilai optimal dari tiap kombinasi. Tabel 6.4 merupakan tabel hasil percobaan kombinasi Cr dan Mr .

Tabel 6.3 Hasil Uji Coba Kombinasi Cr dan Mr

| $Cr; Mr$ | Nilai <i>Fitness</i> | | | | | | | | | | Rata-rata nilai <i>Fitness</i> |
|----------|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------------------------------|
| | Pengujian ke- | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1.0;0.0 | 2.344 | 2.321 | 2.349 | 2.321 | 2.380 | 2.334 | 2.313 | 2.404 | 2.285 | 2.372 | 2.342 |
| 0.9;0.1 | 2.460 | 2.500 | 2.479 | 2.493 | 2.488 | 2.506 | 2.502 | 2.495 | 2.539 | 2.499 | 2.496 |
| 0.8;0.2 | 2.503 | 2.540 | 2.527 | 2.530 | 2.517 | 2.454 | 2.535 | 2.466 | 2.469 | 2.543 | 2.508 |
| 0.7;0.3 | 2.484 | 2.480 | 2.488 | 2.505 | 2.521 | 2.542 | 2.505 | 2.529 | 2.543 | 2.522 | 2.512 |
| 0.6;0.4 | 2.589 | 2.547 | 2.494 | 2.520 | 2.479 | 2.533 | 2.508 | 2.529 | 2.534 | 2.497 | 2.523 |
| 0.5;0.5 | 2.530 | 2.514 | 2.539 | 2.536 | 2.492 | 2.573 | 2.562 | 2.549 | 2.602 | 2.526 | 2.542 |
| 0.4;0.6 | 2.513 | 2.575 | 2.517 | 2.523 | 2.522 | 2.523 | 2.504 | 2.526 | 2.541 | 2.533 | 2.528 |
| 0.3;0.7 | 2.491 | 2.522 | 2.576 | 2.560 | 2.576 | 2.572 | 2.587 | 2.517 | 2.535 | 2.563 | 2.550 |
| 0.2;0.8 | 2.517 | 2.559 | 2.531 | 2.546 | 2.560 | 2.516 | 2.550 | 2.549 | 2.602 | 2.602 | 2.553 |
| 0.1;0.9 | 2.575 | 2.589 | 2.570 | 2.589 | 2.556 | 2.589 | 2.589 | 2.575 | 2.581 | 2.589 | 2.580 |
| 0.0;1.0 | 2.564 | 2.574 | 2.592 | 2.571 | 2.573 | 2.540 | 2.571 | 2.575 | 2.558 | 2.569 | 2.569 |



Gambar 6.4 Grafik Hasil Uji Coba Kombinasi Cr dan Mr

Gambar 6.4 menjelaskan grafik hasil pengujian kombinasi nilai *Crossover rate* dan *Mutation rate* dalam 10 kali pengujian. Terjadi kenaikan pada nilai *fitness* dari kombinasi nilai *Cr* 1 dan *Mr* 0,9 ke kombinasi nilai *Cr* 0,9 dan *Mr* 1. Sementara itu kombinasi selanjutnya tidak mengalami perubahan nilai *fitness* secara signifikan. Tidak ada ketetapan nilai *Cr* maupun *Mr* yang digunakan untuk mendapatkan solusi optimal. Solusi optimal diperoleh pada kombinasi nilai *Crossover rate* 0,1 dan *Mutation rate* 0,9 dengan nilai rata-rata *fitness* sebesar 2,580.

6.4 Pengujian Akhir atau Validasi

Perancangan pengujian akhir atau pengujian validasi dilakukan untuk mendapatkan nilai optimal dari individu terbaik. Perancangan pengujian dilakukan dengan mengambil individu terbaik dari nilai *fitness* dari pengujian parameter yang telah diujikan sebelumnya. Pengujian dilakukan menggunakan nilai generasi terbaik pada generasi ke-300, nilai populasi ke-90 serta kombinasi nilai *Cr;Mr* sebesar 0.1;0.9. Pengujian dilakukan sebanyak 5 kali pengujian agar memperoleh solusi optimal.

Tabel 6.4 Hasil Pengujian Akhir atau Validasi

| Individu Terbaik | Pengujian | | | | | Rata-rata Nilai <i>Fitness</i> |
|------------------|-----------|-------|-------|-------|-------|--------------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| 1 | 2.602 | 2.589 | 2.589 | 2.561 | 2.572 | 2.583 |

Pada Tabel 6.4 yang merupakan hasil pengujian akhir atau validasi dapat dilihat nilai *fitness* pada tiap pengujian tidak memiliki nilai yang terlalu jauh sehingga diperoleh nilai rata-rata *fitness* sebesar 2,583.

6.5 Evaluasi Sistem

Evaluasi sistem melakukan perbandingan antara data aktual dengan data prediksi pada program. Evaluasi ini dilakukan agar hasil prediksi pada program dapat memberikan nilai yang optimal. Dalam hal ini rute awal yang terdapat pada data aktual dibandingkan dengan rute rekomendasi dari program. Perhitungan selisih absolut pada data jarak rute tiap gudang dapat dihitung dengan persamaan 2.5. Tabel 6.5 merupakan hasil perhitungan evaluasi selisih jarak pada tiap gudang.

Tabel 6.5 Hasil Evaluasi Selisih Jarak

| No | Gudang | Jarak Aktual (km) | Jarak Program (km) | Selisih (km) |
|----|----------------|-------------------|--------------------|--------------|
| 1 | GBB Gadang | 437,1 | 153,7 | 283,4 |
| 2 | GBB Kebonagung | 456,7 | 160,8 | 295,9 |
| 3 | GBB Gadingrejo | 171,4 | 164,9 | 6,5 |
| 4 | GBB Kejapanan | 180,8 | 160,1 | 20,7 |

Dari perhitungan evaluasi selisih pada Tabel 6.5 dinyatakan bahwa perhitungan jarak rute pada program memiliki nilai yang lebih optimal sehingga proses penyaluran dapat lebih efisien. Hasil rute optimal pada tiap gudang yaitu menampilkan trayek angkutan terbaik.

Trayek angkutan terbaik gudang GBB Gadang:

KP GBB Gadang → Tambaksari → Sempalwadak → Jambearjo → Tajinan → Kasembon → Bululawang → Kregbet → Banjarejo → Tempursari → Donomulyo → Tlogosari → Jatisari → KP GBB Gadang

Trayek angkutan terbaik gudang GBB Kebonagung:

KP GBB Kebonagung → Wajak → Jatiguwi → Sumberpucung → Blayu → Ngebruk → Karangates → Tumpakrejo → Gedangan → Patokpilis → Sumberejo → Kidangbang → Segaran → KP GBB Kebonagung

Trayek angkutan terbaik gudang GBB Gadingrejo:

KP GBB Gadingrejo → Losari → Candirenggo → Watugede → Gunungrejo → Singosari → Tunjungtirto → Tamanharjo → Langlang → Baturetno → Randuagung → Banjararum → Dengkol → KP GBB Gadingrejo

Trayek angkutan terbaik gudang GBB Kejapanan:

KP GBB Kejapanan → Sidoluhur → Ketindan → Bedali → Wonorejo → Srigading → Sumberngepoh → Sidodadi → Sumberporong → Kalirejo → Lawang → Mulyoarjo → Turirejo → KP GBB Kejapanan

BAB 7 PENUTUP

Bab penutup merupakan ulasan mengenai kesimpulan yang diperoleh pada saat hasil penelitian telah dilakukan. Penutup juga meliputi saran sebagai bahan pengembangan atau kelanjutan dari penelitian.

7.1 Kesimpulan

Kesimpulan yang didapat pada Penerapan *Multi Travelling Salesman Problem* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera Studi Kasus: Perum Bulog Subdivre Malang adalah sebagai berikut:

1. Dalam penerapan *Multi Travelling Salesman Problem (m-TSP)* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog, hasil pengujian nilai-nilai parameter algoritme genetika, yakni Ukuran Populasi (*popSize*), Persilangan (*Crossover*), Mutasi (*Mutation*), Seleksi menggunakan *Elitism*, dan Evaluasi memberikan pengaruh yang signifikan, terlihat dari nilai *fitness* yang semakin baik. Dari penggunaan parameter juga menghasilkan data jarak prediksi program yang lebih optimal dibandingkan jarak pada data aktual sebelumnya. Pada parameter ukuran populasi didapatkan populasi terbaik di populasi ke-90. Generasi terbaik yang didapatkan yaitu pada generasi ke-300, serta kombinasi terbaik dari nilai *Crossover rate (Cr)* dan *Mutation rate (Mr)* yaitu 0.1 dan 0.9.
2. Pada penelitian ini menghasilkan hasil evaluasi sebagai berikut: penggunaan parameter terbaik yakni nilai parameter populasi ke-90, nilai generasi ke-300, dan kombinasi nilai *Crossover rate (Cr)* dan *Mutation rate (Mr)* yaitu 0.1 dan 0.9 menghasilkan nilai rata-rata *fitness* sebesar 2.583. Dengan hasil berikut dapat disimpulkan bahwa penggunaan nilai parameter yang tepat dapat meningkatkan performa pada sistem. Disamping itu nilai selisih yang didapatkan pada data prediksi program lebih efisien jika dibandingkan dengan data aktual sehingga proses penyaluran dapat dikatakan lebih efisien.

7.2 Saran

Penelitian ini telah didapatkan kesimpulan sehingga ada saran yang dapat diberikan untuk kelanjutan pada penelitian penerapan *Multi Travelling Salesman Problem (m-TSP)* Pada Optimasi Pendistribusian Bantuan Sosial Beras Sejahtera (Bansos Rastra) Studi Kasus: Perum Bulog, saran yang dapat diberikan pada poin-poin dibawah ini:

1. Pada penelitian yang telah dilakukan, permasalahan *Multi Travelling Salesman Problem (m-TSP)* dapat digunakan untuk pencarian rute terbaik. Diharapkan pada penelitian selanjutnya dapat menambahkan metode seputar pencarian rute terbaik atau optimasi agar data yang dihasilkan dapat lebih optimal.

2. Penelitian ini menggunakan parameter algoritme genetika yang terdiri dari Ukuran Populasi (*popSize*), Persilangan (*Crossover*) menggunakan *one-cut-point*, Mutasi (*Mutation*) menggunakan *Reciprocal Exchange*, Seleksi menggunakan *elitism*, dan Evaluasi. Diharapkan untuk penelitian selanjutnya menggunakan metode yang berbeda dari penelitian ini untuk mendapatkan hasil yang lebih bervariasi dan meningkatkan performa yang dihasilkan.



DAFTAR REFERENSI

- Aditya, C. & Mahmudy, W. F., 2016. Optimasi Persediaan Baju Menggunakan Algoritma Genetika. *Prosiding Seminar Nasional Teknologi dan Rekayasa Informasi Tahun 2016*, p. 78.
- Balaipustaka, 2009. *Balai Pustaka's weblog*. [Online] Available at: <https://balaipustaka.wordpress.com/2009/03/15/pengertian-distribusi/> [Accessed 25 September 2018].
- Bulog, 2018. *Sekilas Perum Bulog*. [Online] Available at: <http://www.bulog.co.id/sekilas.php> [Accessed 25 September 2018].
- Eroy, A. R., 2010. *Slideshare*. [Online] Available at: <https://www.slideshare.net/aryarifqu/strategi-distribusi-pemasaran> [Accessed 30 Juli 2018].
- Gen, M. & Cheng, R., 1997. *Genetic Algorithm and Engineering Design*. New York: John Wiley & Sons, Inc.
- Gen, M. & Cheng, R., 2000. *Genetic Algorithms And Engineering Design*. New York: Willy & Sons, Inc..
- Karimah, S., Widodo, A. W. & Cholissodin, I., 2017. Optimasi Multiple Travelling Salesman Problem Pada Pendistribusian Air Minum Menggunakan Algoritme Genetika (Studi Kasus: UD. Tosa Malang)). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume I, pp. 849-858.
- Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Mahmudy, W. F., 2015. *Dasar-Dasar Algoritma Evolusi*, Malang: Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- Mahmudy, W. F. & Mawaddah, N. K., 2006. Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika. *Kursor*, 2(2), pp. 1-8.
- Ponraj, R. & Amalanathan, G., 2014. Optimizing Multiple Travelling Salesman Problem Considering The Road Capacity. *Journal of Computer Science* 10, Volume IV, pp. 680-688.
- Putri, F. B., Mahmudy, W. F. & Ratnawati, D. E., 2015. Penerapan Algoritma Genetika Untuk Vehicle Routing Problem. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, Volume V, pp. 1-9.
- Raditya, P. M. R. & Dewi, C., 2018. Optimasi Multiple Travelling Salesman Problem (M-TSP) Pada Penentuan Rute Optimal Penjemputan Penumpang Travel Menggunakan Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume II, pp. 3560-3568.

- Reduction, T. N. T. F. T. A. o. P., 2018. *Materi Sosialiasi Bantuan Sosial Beras Sejahtera (BANSOS RASTRA)*, s.l.: The National Team For The Acceleration of Poverty Reduction.
- Santosa, B. & Willy, P., 2011. *Metoda Metaheuristik: Konsep dan Implementasi*. In: *Metoda Metaheuristik: Konsep dan Implementasi*. s.l.:Guna Widya, pp. 8-13.
- Saptaningtyas, F. Y., 2012. MULTI TRAVELING SALESMAN PROBLEM (MTSP) DENGAN ALGORITMA GENETIKA UNTUK MENENTUKAN RUTE LOPER KORAN DI AGEN SURAT KABAR. *Pythagoras*, VII(2), pp. 55-64.
- Setiawan, K., 2003. *Paradigma Sistem Cerdas*. Surabaya: Bayumedia.
- Suyanto, 2005. *Algoritma Genetik Dalam Matlab*. Yogyakarta: Andi.
- Suyanto, 2011. *Artificial Intelligence (Searching, Reasoning, Planning, and Learning)*. Bandung: Informatika.
- Talbi, E.-G., 2009. *Metaheuristics: From Design to Implementation*. s.l.:s.n.
- Widodo, A. W. & Mahmudy, W. F., 2010. Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner. *Jurnal Ilmiah Kursor*, Volume V, pp. 205-211.
- Widodo, T. S., 2012. *Komputasi Evolusioner*. Yogyakarta: Graha Ilmu.
- Wijyaningrum, V. N. & Mahmudy, W. F., 2016. Optimization of Ship's Route Scheduling Using Genetic Algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, Volume II, pp. 180 - 186.
- Yang, X. S., 2010. Engineering Optimization. In: *Engineering Optimization: An Introduction with Metaheuristic Applications*. s.l.:Taylor & Francis, pp. 7-8.
- Yang, X. S., 2012. *Nature-Inspired Metaheuristic Algorithms: Success and New Challenges*. s.l.:s.n.
- Zbigniew, M., 1999. *Genetic Algorithm + Data Structures = Evolution Programs*. New York: Springer Verlag Berlin Heidelberg.