

**IMPLEMENTASI TRAVELLING SALESMAN PROBLEM PADA
PENDISTRIBUSIAN AIR MINUM MENGGUNAKAN METODE NON-
DOMINATED SORTING GENETIC ALGORITHM (NSGA-II)**

SKRIPSI

Oleh :
KEVIN NAUFAL FAHREZI
NIM. 19650109



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**IMPLEMENTASI TRAVELLING SALESMAN PROBLEM PADA
PENDISTRIBUSIAN AIR MINUM MENGGUNAKAN METODE NON-
DOMINATED SORTING GENETIC ALGORITHM (NSGA-II)**

SKRIPSI

Oleh :
KEVIN NAUFAL FAHREZI
NIM. 19650109

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

IMPLEMENTASI *TRAVELLING SALESMAN PROBLEM* PADA PENDISTRIBUSIAN AIR MINUM MENGGUNAKAN METODE *NON- DOMINATED SORTING GENETIC ALGORITHM (NSGA-II)*

SKRIPSI

Oleh :

KEVIN NAUFAL FAHREZI

NIM. 19650109

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 26 Juni 2023

Pembimbing I,



Juniardi Nur Fadila, M.T
NIP. 19920605 201903 1 015

Pembimbing II,



Supriyono, M.Kom
NIP. 19841010 201903 1 012

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

IMPLEMENTASI *TRAVELLING SALESMAN PROBLEM* PADA PENDISTRIBUSIAN AIR MINUM MENGGUNAKAN METODE *NON- DOMINATED SORTING GENETIC ALGORITHM (NSGA-II)*

SKRIPSI

Oleh :

KEVIN NAUFAL FAHREZI

NIM. 19650109

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 18 Oktober 2023

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	()
Anggota Penguji I	: <u>Johan Ericka Wahyu P, M. Kom</u> NIP. 19831213 201903 1 004	()
Anggota Penguji II	: <u>Juniardi Nur Fadilah, M.T</u> NIP. 19920605 201903 1 015	()
Anggota Penguji III	: <u>Supriyono, M. Kom</u> NIP. 19841010 201903 1 012	()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang


Dr. Fachrudin Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Kevin Naufal Fahrezi

NIM : 19650109

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Implementasi *Travelling Salesman Problem* Pada
Pendistribusian Air Minum Menggunakan Metode *Non-Dominated Sorting Genetic Algorithm*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 18 Oktober 2023
Yang membuat pernyataan,



Kevin Naufal Fahrezi
NIM.19650109

MOTTO

... Jangan Pernah Berhenti ...

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamiin seluruh puji bagi Allah subhanahu wa ta'ala, serta shalawat dan salam kepada nabi agung Muhammad SAW, sehingga penulis mendapatkan kesempatan untuk menyelesaikan tugas akhir. Dengan rasa hormat dan terimakasih, penulis mempersembahkan skripsi tugas akhir ini kepada:

1. Kedua orang tua penulis, Bapak Suhartono dan Ibu Nety Latiefah sebagai alasan dibalik perjuangan dan kesuksesan anaknya kelak. Gelar sarjana ini akan selamanya terpatrit pada nama belakang kalian yang tercinta mama dan papa, bukan padaku. Kedua orang yang kurang beruntung untuk dapat merasakan bangku pendidikan sekolah tinggi, namun mampu menyekolahkan anaknya hingga bergelar sarjana.
2. Kakak penulis Dinda Larasati. Sosok musuh yang penulis takuti ketika masa kecil, namun saudara yang paling penulis sayangi ketika dewasa. Seseorang yang menginspirasi penulis untuk dapat terus berjuang dan bertahan dalam kondisi yang sulit. Terima kasih karena sudah menjadi kakak yang baik untuk adik-adiknya. *You're doing more than just enough.*
3. Bapak Juniardi Nur Fadila, M.T dan Bapak Supriyono, M.Kom selaku Dosen Pembimbing yang tulus dalam memberikan bimbingan dalam penyelesaian tugas akhir.
4. Didier Yves Drogba Tébily. Motor kesayangan penulis yang telah menemani sejak 2014 hingga sekarang. Terima kasih telah menjadi saksi bisu terutama dalam masa masa terpuruk ketika menggarap tugas akhir ini

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat serta kesehatan, sehingga penulis mampu menyelesaikan penelitian ini dengan baik. Penulis menyampaikan ucapan terima kasih kepada semua pihak yang pernah terlibat langsung maupun tidak langsung dalam menyelesaikan penelitian ini, bukan hanya karena usaha keras dari penulis sendiri, akan tetapi karena adanya dukungan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan M.MT., IPM selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang
4. Juniardi Nur Fadilah, M.T dan Supriyono, M.Kom selaku Dosen Pembimbing yang tulus dalam memberikan bimbingan dalam penyelesaian tugas akhir.
5. Dr. Cahyo Crysdian selaku penguji I dan Johan Ericka Wahyu P, M.Kom selaku penguji II yang telah meluangkan waktunya untuk menguji dan dengan sabar memberi arahan dan saran dalam menyelesaikan skripsi ini.
6. Segenap civitas akademik Program Studi Teknik Informatika, dan seluruh dosen yang telah memberikan ilmu serta arahan semasa kuliah.

7. Kedua orang tua penulis, Bapak Suhartono dan Ibu Nety Latiefah, serta keluarga besar yang telah memberikan banyak dukungan, doa serta selalu menjadi semangat sehingga penulis mampu menyelesaikan masa studi hingga mencapai gelar sarjana.
8. Saudara kandung penulis, Dinda Larasati, Ellena Ardy Cahyarini dan Evalia Kirana Dewi, yang telah memberikan dukungan dan doa yang selalu menjadi semangat sehingga penulis dapat menyelesaikan masa studi hingga mencapai gelar sarjana.

Skripsi yang telah ditulis ini masih jauh dari kata sempurna, oleh karena itu penulis sangat menghargai dan senang jika terdapat kritik dan saran yang diberikan. Semoga skripsi ini dapat memberikan manfaat.

Wassalamu alaikum, Wr. Wb.

Malang, 18 Oktober 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
ملخص.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian	5
1.6 Hipotesis.....	6
1.7 Sistematika Penulisan	6
BAB II STUDI PUSTAKA	8
2.1 <i>Travelling Salesman Problem</i>	8
2.2 Algoritma Genetika.....	12
2.3 Metode <i>Non-Dominated Sorting</i>	14
2.4 Pendistribusian	15
BAB III DESAIN PENELITIAN.....	17
3.1 Rangkaian Penelitian.....	17
3.2 <i>Data Collection</i>	18
3.3 Desain Arsitektur <i>Travelling Salesman Problem</i> (TSP)	18
3.4 Desain Arsitektur <i>Non-Dominated Sorting Genetic Algorithm-II</i> (NSGA-II)	24
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Implementasi Sistem	34
4.1.1 Alur TSP & NSGA-II Pada Sistem.....	35
4.1.2 Implementasi <i>Design Interface</i>	40
4.2 Uji Coba Sistem	44

4.2.1 Pengujian.....	45
4.2.2 Hasil Uji Coba	46
4.3 Pembahasan.....	47
BAB V KESIMPULAN DAN SARAN	50
5.1 Kesimpulan	50
5.2 Saran.....	51
DAFTAR PUSTAKA	52

DAFTAR GAMBAR

Gambar 3.1 <i>Flowchart</i> alur <i>Travelling Salesman Problem</i>	23
Gambar 3.2 Proses <i>Non-dominated sorting</i> menuju <i>Crowding Distance</i>	31
Gambar 3.3 <i>Flowchart</i> alur NSGA-II	33
Gambar 4.1 Tampilan Utama	42
Gambar 4.2 Visualisasi Grafik	43
Gambar 4.3 Visualisasi Maps	44

DAFTAR TABEL

Tabel 3.1 <i>Dataset</i> Pendistribusian PT. Krakatau Daya Tirta	18
Tabel 4.1 Rute yang dapat dilalui oleh <i>Salesman</i>	46
Tabel 4.2 Pengujian jarak dari rute yang dapat dilalui	46

ABSTRAK

Fahrezi, Kevin Naufal. 2023. **Implementasi *Travelling Salesman Problem* Pada Pendistribusian Air Minum Menggunakan Metode *Non-Dominated Sorting Genetic Algorithm (NSGA-II)***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Juniardi Nur Fadila, M.T (II) Supriyono, M. Kom

Kata Kunci : Travelling Salesman Problem, Pendistribusian, Non-Dominated Sorting Genetic Algorithm (NSGA-II)

Travelling Salesman Problem (TSP) adalah permasalahan optimasi yang mencari rute terpendek yang mengunjungi sejumlah kota dan kembali ke kota awal. Pendistribusian air minum merupakan masalah penting dalam manajemen sumber daya air dan membutuhkan perencanaan rute yang efisien untuk menghemat waktu dan biaya. Metode NSGA-II digunakan untuk mengatasi TSP dalam konteks pendistribusian air minum. NSGA-II adalah algoritma evolusi yang digunakan untuk mencari solusi optimal dalam ruang pencarian yang kompleks. Dalam skripsi ini, NSGA-II diterapkan untuk menghasilkan rute pendistribusian air minum yang efisien.

ABSTRACT

Fahrezi, Kevin Naufal. 2023. **Implementasi Travelling Salesman Problem Pada Pendistribusian Air Minum Menggunakan Metode Non-Dominated Sorting Genetic Algorithm (NSGA-II)**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Juniardi Nur Fadila, M.T (II) Supriyono, M. Kom

Travelling Salesman Problem (TSP) is an optimization problem that seeks the shortest route visiting a number of cities and returning to the starting city. The distribution of drinking water is a critical issue in the management of water resources and requires efficient route planning to save time and costs. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) method is employed to address the TSP in the context of water distribution. NSGA-II is an evolutionary algorithm used to find optimal solutions in complex search spaces. In this thesis, NSGA-II is applied to generate efficient routes for water distribution.

Keywords : Travelling Salesman Problem, Pendistribusian, Non-Dominated Sorting Genetic Algorithm (NSGA-II)

ملخص

فخريزي، كيفين نوفل. ٢٠٢٣. تنفيذ مسألة البائع المتجول في توزيع مياه الشرب باستخدام طريقة خوارزمية الفرز الجينية غير المهيمنة (NSGA-II). البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: جونيادي نور فضيلة، الماجستير. المشرف الثاني: سوفريونو، الماجستير.

الكلمات الرئيسية: مسألة بائع متجول، توزيع، خوارزمية الفرز الجينية غير المهيمنة (NSGA-II).

مسألة البائع المتجول (TSP) هي مشكلة تحسين تبحث عن أقصر طريق يزور عددا من المدن ويعود إلى المدينة الأصلية. يعد توزيع مياه الشرب قضية مهمة في إدارة الموارد المائية ويتطلب تخطيطا فعالا للطريق لأجل توفير الوقت والتكاليف. واستخدمت طريقة خوارزمية الفرز الجينية غير المهيمنة في سياق توزيع مياه الشرب. NSGA-II هي خوارزمية تطورية تستخدم للبحث عن الحلول المثلى في مساحات البحث المعقدة. في هذا البحث الجامعي، يتم تطبيق NSGA-II لإنتاج طرق فعالة لتوزيع مياه الشرب.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendistribusian air minum adalah salah satu masalah yang sering dihadapi oleh perusahaan air minum. Perusahaan harus memastikan bahwa air minum tersedia dan dapat diakses oleh masyarakat dengan efisien (Puteri *et al.*, 2017). Namun, masalah ini seringkali diperparah oleh kendala jarak dan jumlah rute yang harus ditempuh. Pendistribusian air minum seringkali menghadapi masalah optimasi, seperti menentukan rute terpendek, menentukan biaya yang efisien, dan menentukan waktu yang tepat untuk melakukan pendistribusian.

Penelitian ini memiliki tujuan untuk membuat sistem yang dapat membantu proses distribusi dengan biaya minimum dan rute terpendek, permasalahan tersebut disebut dengan *Travelling salesman problem* (TSP). *Travelling salesman problem* (TSP) adalah salah satu masalah optimasi yang berkaitan dengan penentuan rute terpendek untuk melalui sejumlah titik dan kembali ke titik awal (Fauzi & Sulistyono, 2022). *Travelling salesman problem* (TSP) adalah masalah optimasi klasik yang memiliki aplikasi dalam berbagai bidang, salah satunya adalah pendistribusian air minum. TSP mencoba menentukan rute terpendek yang harus dilalui oleh salesmen dalam melakukan kunjungan ke beberapa kota (Kusrini & Istiyanto, 2007).

Dalam hal ini, masalah TSP dapat digunakan untuk menyelesaikan masalah pendistribusian air minum (Made *et al.*, 2018). *Travelling salesman problem* (TSP) dapat diselesaikan menggunakan berbagai metode optimisasi. Beberapa metode

optimasi yang dapat digunakan dalam pendistribusian air minum antara lain adalah algoritma genetika (GA), algoritma *particle swarm optimization* (PSO), dan algoritma *tabu search* (TS) dan juga *non-dominated sorting genetic algorithm II* (NSGA-II), yang merupakan salah satu metode yang efektif untuk menyelesaikan permasalahan *multi-objective* seperti TSP (Fatmawati *et al.*, 2015).

Dalam penelitian ini, NSGA-II akan diterapkan pada masalah pendistribusian air minum untuk menentukan rute terpendek, biaya yang efisien, dan waktu yang tepat untuk melakukan pendistribusian. *Non-dominated sorting genetic algorithm* (NSGA-II) adalah salah satu metode optimasi yang dapat digunakan untuk menyelesaikan masalah TSP (Myszkowski *et al.*, 2019). NSGA-II memungkinkan penentuan rute terpendek dengan mempertimbangkan berbagai kriteria seperti jarak, biaya, dan waktu. *Non-dominated sorting genetic algorithm* (NSGA-II) adalah algoritma optimasi *multi-objective* yang menggabungkan teori algoritma genetika dan teori *non-dominated sorting* (Belluano, 2016). NSGA-II dapat digunakan untuk memecahkan masalah optimasi *multi-objective* yang memiliki beberapa kriteria optimasi yang saling bertentangan. Dalam penelitian ini, NSGA-II akan digunakan sebagai metode optimasi untuk mengatasi masalah *travelling salesman problem* (TSP) dalam pendistribusian air minum. NSGA-II akan membantu menentukan rute terpendek, biaya yang efisien, dan waktu yang tepat untuk melakukan pendistribusian air minum.

Data yang digunakan akan berupa lokasi titik pengiriman air minum dan jarak antar titik. Hasil implementasi ini akan membantu dalam menentukan

efektivitas dan efisiensi NSGA-II dalam mengatasi masalah optimasi pendistribusian air minum.

Dalam konteks implementasi *non-dominated sorting genetic algorithm* (NSGA-II) pada *travelling salesman problem* (TSP) dalam pendistribusian air minum, integrasi sains dan Islam dapat diwujudkan melalui penerapan konsep-konsep sains dan teknologi yang selaras dengan prinsip-prinsip Islam dalam pengambilan keputusan yang bijaksana, berkelanjutan, dan berwawasan lingkungan. Dalam surah Al-Asr, Allah Subhanahu wa ta'ala mengajarkan tentang pentingnya waktu dan bekerja dengan tekun untuk mencapai tujuan. Sebagaimana diriwayatkan dalam tafsir Jalalain, kita semua orang-orang yang beriman harus lebih memperhatikan masalah waktu, dan mampu memanfaatkan waktu sebaik-baiknya untuk hal-hal yang terpuji sesuai ajaran Islam. Hal ini sesuai dengan konsep dasar yang digunakan dalam implementasi *travelling salesman problem* dimana algoritma tersebut digunakan untuk mencari solusi terbaik dengan memaksimalkan kriteria efisiensi waktu dalam hal pendistribusian air minum. Dalam ayat tersebut, Allah Subhanahu wa ta'ala berfirman:

وَالْعَصْرِ (١) إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ (٢) إِلَّا الَّذِينَ ءَامَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ (٣)

“*Demi masa, sesungguhnya manusia berada dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal shalih, serta saling menasehati dalam kebenaran dan saling menasehati dalam kesabaran.*” (QS Al-Asr: 1-3)”

Integrasi ayat tersebut dapat diharapkan akan digunakan untuk pengimplementasian tentang pentingnya menggunakan waktu sebaik mungkin dalam mencapai tujuan yang diinginkan, seperti dalam mencari solusi terbaik untuk permasalahan pendistribusian air minum. Sebagaimana riwayat Al-Imam Asy-

Syafi'i rahimahullah dalam kitab Al Jawaabul Kaafi karya Ibnul Qayyim juga disebutkan bahwa:

الْوَقْتُ سَيْفٌ فَإِنْ لَمْ تَقْطَعْهُ قَطَعَكَ، وَنَفْسُكَ إِنْ أَشْغَلْتَهَا بِالْحَقِّ وَالْإِلَّا اشْتَغَلَتْكَ بِالْبَاطِلِ

"Waktu ibarat pedang, jika engkau tidak menebasnya maka ialah yang akan menebasmu. Dan jiwamu jika tidak kau sibukkan di dalam kebenaran maka ia akan menyibukkanmu dalam kebatilan (Kitab Al Jawaabul Kaafi karya Ibnul Qayyim)".

Sebagai pemimpin di bumi, manusia bertanggung jawab untuk memelihara bumi dan memanfaatkan sumber daya alam secara bijak dan bertanggung jawab (Asy'ari, 2019). Penggunaan algoritma genetika dalam optimasi rute pendistribusian air minum dapat membantu mengurangi penggunaan waktu yang digunakan dan mengoptimalkan penggunaan sumber daya secara efisien, sejalan dengan nilai-nilai Islam tentang menjaga lingkungan hidup dan memperoleh keuntungan dari sumber daya alam secara bertanggung jawab. Dengan demikian, integrasi sains dan Islam pada penelitian ini dapat memberikan kontribusi positif bagi pengembangan pendekatan dan solusi yang holistik dan berkelanjutan dalam bidang pendistribusian air minum.

Dengan menentukan hasil implementasi NSGA-II dengan metode optimasi lain, diharapkan dapat diketahui efektivitas dan efisiensi dari NSGA-II dalam mengatasi masalah optimasi pendistribusian air minum. Hal ini akan membantu dalam menentukan metode optimasi terbaik yang dapat digunakan dalam pendistribusian air minum. Dengan demikian, metode NSGA-II digunakan dalam penelitian ini karena mampu mengatasi masalah TSP dengan mempertimbangkan beberapa kriteria sekaligus dan menemukan solusi optimasi yang non-dominan (Belluano, 2016)

1.2 Pernyataan Masalah

Bagaimana implementasi *non-dominated sorting genetic algorithm* (NSGA-II) pada *travelling salesman problem* (TSP) dapat membantu mengukur efektivitas dalam aspek jarak dan waktu pada pendistribusian air minum?

1.3 Tujuan Penelitian

Mengukur efektivitas jarak dan waktu pada metode NSGA-II dalam menentukan rute terpendek untuk pendistribusian air minum.

1.4 Batasan Masalah

Penelitian ini hanya akan memfokuskan pada pendistribusian air minum dalam jangka waktu tertentu serta jumlah rute pendistribusian yang terbatas yaitu rute pendistribusian PT. Krakatau Daya Tirta Cilegon dan tidak mempertimbangkan faktor-faktor eksternal yang mungkin mempengaruhi waktu pendistribusian.

1.5 Manfaat Penelitian

Memberikan solusi optimasi bagi industri pendistribusian air minum dalam hal rute terpendek, biaya yang efisien, dan waktu yang tepat untuk melakukan pendistribusian.

1.6 Hipotesis

Penelitian ini merumuskan hipotesis bahwa implementasi *non-dominated sorting genetic algorithm* (NSGA-II) pada *travelling salesman problem* (TSP) dalam konteks pendistribusian air minum akan memberikan kontribusi positif terhadap efektivitas dan efisiensi operasional. Secara khusus, hipotesis ini merinci bahwa:

1. Hipotesis Nol: Tidak ada perbedaan signifikan dalam jarak tempuh antara rute pendistribusian air minum yang dihasilkan oleh NSGA-II dan metode konvensional
2. Hipotesis Alternatif: Penggunaan NSGA-II akan menghasilkan rute pendistribusian dengan jarak tempuh yang lebih optimal dibandingkan dengan metode konvensional.

Hipotesis-hipotesis ini akan diuji melalui analisis komparatif antara solusi-solusi yang dihasilkan oleh NSGA-II dan metode pendistribusian konvensional.

1.7 Sistematika Penulisan

Gambaran singkat tentang struktur keseluruhan skripsi, membantu pembaca untuk memahami urutan isi dan tujuan setiap bagian. Sistematika penulisan dapat dijelaskan sebagai berikut:

1. Pendahuluan

Dalam bagian pendahuluan, pembaca akan dibawa ke dalam konteks permasalahan pendistribusian air minum yang perlu dioptimalkan. Latar belakang penelitian akan disampaikan bersama dengan tujuan dan manfaat dari penelitian ini, serta pembatasan masalah yang menjadi fokus.

2. Studi Pustaka

Bagian kajian pustaka memberikan wawasan mendalam tentang *travelling salesman problem* (TSP), *non-dominated sorting genetic algorithm* (NSGA-II), dan aplikasinya dalam konteks pendistribusian air minum. Kajian literatur relevan akan dijelaskan untuk memberikan landasan teoretis yang kuat bagi penelitian ini

3. Desain Penelitian

Dalam merinci metode penelitian, pembaca akan diberikan gambaran lengkap tentang pendekatan yang digunakan, langkah-langkah implementasi NSGA-II pada TSP, serta deskripsi data, variabel, dan pengukuran yang relevan. Proses analisis data yang akan dilakukan juga akan dijabarkan secara rinci

4. Hasil dan Pembahasan

Dalam bagian ini, hasil-hasil dari penerapan NSGA-II pada TSP untuk pendistribusian air minum akan dipaparkan secara terperinci. Setiap variabel penelitian dievaluasi, termasuk optimisasi jarak tempuh, reduksi waktu pendistribusian, dan alokasi sumber daya yang efisien

5. Kesimpulan & Saran

Dalam bagian kesimpulan, pembaca akan diajak untuk merangkum hasil penelitian dan interpretasinya. Implikasi kesimpulan terhadap teori dan praktik akan diuraikan, dan saran untuk penelitian selanjutnya juga disampaikan

BAB II

STUDI PUSTAKA

2.1 Travelling Salesman Problem

Sejak tahun 1800, permasalahan TSP telah ditemukan oleh Sir William Rowan Hamilton, seorang matematikawan asal Irlandia, dan Thomas Penyngton Kirkman, matematikawan dari Inggris. Kemudian, para matematikawan seperti Karl Menger di Vienna dan Harvard mempelajari bentuk umum dari TSP pada tahun 1930-an. TSP merupakan permasalahan optimasi klasik yang termasuk dalam kategori *non-deterministic polynomial-time* (NPC) (Vitaloka, 2018). Dalam TSP, tidak ada solusi yang optimal selain dengan mencoba semua kemungkinan solusi. Tujuan utama TSP adalah mencari nilai perjalanan minimum bagi seorang *salesman*. Dalam TSP, tidak ada solusi yang optimal selain dengan mencoba semua kemungkinan solusi (Andri *et al.*, 2013). Meskipun jalur dengan urutan yang berbeda dapat menghasilkan jarak yang sama, jalur dengan urutan yang berbeda tetap dihitung sebagai jalur yang berbeda. Untuk mencari jumlah rute yang mungkin diperoleh pada TSP, dapat digunakan rumus permutasi berikut:

$${}_nP_k = \frac{n!}{(n-k)!} \quad (2.1)$$

Dengan n adalah jumlah seluruh kota dan k adalah jumlah kota yang diseleksi.

Myszkowski *et al.*, (2019) mempresentasikan algoritma genetika turnamen *non-domination sorting* untuk memecahkan permasalahan *multi-objective traveling salesman problem* (MOTSP). Algoritma tersebut dikembangkan dengan mengintegrasikan teknik *tournament selection*, *non-dominated sorting*, dan *elitism*

preservation. Pada setiap generasi, individu terbaik dari turnamen dipilih untuk melakukan operasi *crossover* dan mutasi. Untuk memperbaiki kinerja algoritma, teknik *elitism preservation* digunakan untuk menghindari hilangnya individu terbaik dari generasi sebelumnya. Penelitian ini menunjukkan bahwa algoritma yang diusulkan dapat menghasilkan solusi optimal yang lebih baik daripada beberapa pendekatan *state-of-the-art* lainnya untuk MOTSP. Hal ini lebih memfokuskan pada penggunaan algoritma genetika turnamen *non-dominated sorting* untuk permasalahan *multi-objective* TSP. Selain itu, penelitian tersebut menggunakan pendekatan *elitisme non-dominated sorting*. Data uji yang digunakan dalam penelitian tersebut, menggunakan data uji yang berasal dari sumber publik.

Shuai *et al.*, (2019) membahas tentang penggunaan metode NSGA-II untuk menyelesaikan *multiple traveling salesman problem* (MTSP). MTSP adalah variasi dari TSP, di mana ada beberapa *salesman* yang perlu mengunjungi kumpulan kota tertentu, dan tugasnya adalah menemukan rute optimal yang memenuhi kriteria tertentu. Dalam penelitian tersebut, para penulis mengusulkan metode yang dapat menyelesaikan MTSP dengan mengubahnya menjadi beberapa TSP yang lebih kecil, yang kemudian dapat diselesaikan menggunakan NSGA-II. Hasil eksperimen menunjukkan bahwa metode yang diusulkan dapat memberikan hasil yang baik dalam menyelesaikan MTSP, terutama dalam kasus ketika jumlah kota sangat besar. Penelitian tersebut lebih memfokuskan pada TSP dengan NSGA-II untuk memecahkan masalah *multiple* TSP (MTSP). Pada teknik penyelesaian penelitian tersebut, NSGA-II digunakan untuk memperkenalkan metode efektif untuk memecahkan MTSP dengan menggabungkan TSP dan NSGA-II. Di bagian

evaluasi hasil, penelitian mengevaluasi kinerja algoritma NSGA-II dalam menyelesaikan MTSP dan membandingkannya dengan beberapa algoritma heuristik. Tujuan penelitian tersebut menunjukkan bahwa metode yang diusulkan menghasilkan hasil yang lebih baik daripada metode NSGA-II konvensional dan algoritma heuristik.

Luo *et al.*, (2014) membahas peningkatan performa algoritma NSGA-II untuk menyelesaikan permasalahan *multi-objective traveling salesman problem* (MOTSP). Penelitian tersebut menggunakan dua strategi, yaitu peningkatan pemilihan *parents* dengan mempertimbangkan kedekatan jarak dan perbedaan tingkatan antara individu, dan pengurangan populasi dengan mengeliminasi individu yang mendominasi sebagian besar individu lain dalam populasi. Penulis melakukan eksperimen dengan menguji algoritma yang diusulkan pada beberapa instans tes MOTSP standar, dan membandingkannya dengan NSGA-II dan algoritma *multi-objective* lainnya. Hasilnya menunjukkan bahwa algoritma yang diusulkan memiliki kinerja yang lebih baik dalam mencapai solusi optimal *pareto*. Fokus penelitian tersebut pada TSP sebagai masalah optimasi multi-objektif pada umumnya. Selain itu, penelitian tersebut mencoba meningkatkan performa NSGA-II dengan menggunakan strategi rekombinasi yang berbeda. Namun, kedua penelitian tersebut juga memiliki kesamaan dalam beberapa hal. Keduanya sama-sama menggunakan pendekatan multi-objektif untuk menyelesaikan TSP dan menggunakan NSGA-II sebagai metode optimasi multi-objektif. Selain itu, keduanya juga menunjukkan bahwa pendekatan multi-objektif dan NSGA-II dapat menghasilkan solusi yang lebih baik dan efisien dibandingkan dengan metode

heuristik atau eksak yang lain. Secara keseluruhan, penelitian tersebut memberikan kontribusi dalam pengembangan metode optimasi multi-objektif untuk TSP, meskipun fokus dan strategi yang digunakan sedikit berbeda.

Bolaños *et al.*, (2015) membahas penggunaan NSGA-II untuk menyelesaikan *multiple traveling salesman problem* (MTSP), yaitu perluasan dari TSP yang memerlukan beberapa *salesman* untuk menyelesaikan rute. Algoritma NSGA-II dimodifikasi dengan menggabungkan konsep *guided mutation* dan *elite preservation* untuk meningkatkan performa algoritma dalam menemukan solusi yang optimal. Metode ini diuji pada beberapa instansi dan diuji dengan membandingkan hasilnya dengan metode NSGA-II standar. Hasil pengujian menunjukkan bahwa algoritma yang diusulkan dapat menghasilkan populasi solusi yang lebih baik dengan waktu komputasi yang lebih singkat. Pada perbandingan penelitian, kedua penelitian memiliki tujuan untuk mengoptimalkan rute dalam masalah *travelling salesman problem*, namun masalah yang diselesaikan berbeda. Penelitian ini yang menjadi objek penelitian bertujuan untuk meminimalkan biaya distribusi air minum, sementara penelitian tersebut bertujuan untuk menyelesaikan masalah *multiple traveling salesman problem*. Lalu pada metode, kedua penelitian menggunakan metode yang sama, yaitu *non-dominated sorting genetic algorithm II* (NSGA-II), namun implementasi dan parameter yang digunakan mungkin berbeda. Penelitian ini memiliki objek penelitian hanya memiliki satu objektif yaitu minimasi biaya pengiriman air minum, sedangkan penelitian tersebut memiliki dua objektif yaitu minimasi jarak total dan jumlah kendaraan yang digunakan. Hasil yang diperoleh dari kedua penelitian juga berbeda. Penelitian ini yang menjadi

objek penelitian berhasil menghasilkan rute pengiriman air minum yang optimal dengan biaya yang lebih rendah dibandingkan dengan rute yang digunakan sebelumnya. Sementara itu, penelitian tersebut berhasil menyelesaikan masalah *multiple traveling salesman problem* dan menghasilkan himpunan solusi yang terbaik secara non-dominan. Dataset yang digunakan pada kedua penelitian juga berbeda. Penelitian ini memiliki objek penelitian menggunakan data distribusi air minum di perusahaan PT. Krakatau Daya Tirta, sementara penelitian tersebut menggunakan data dari 13 kota di Kolombia.

Berdasarkan perbandingan penelitian terdahulu, terdapat persamaan dan perbedaan dengan penelitian ini. Penelitian ini merupakan penyelesaian masalah sebuah perusahaan dalam rute yang akan digunakan ketika mendistribusikan produknya. Metode yang digunakan pada penelitian ini yaitu metode NSGA-II.

2.2 Algoritma Genetika

Algoritma genetika merupakan suatu teknik optimasi yang dapat digunakan pada permasalahan dengan ruang pencarian yang luas dan kompleks. Algoritma ini menggunakan pendekatan berbasis populasi sehingga mampu menemukan solusi yang lebih baik daripada nilai maksimum lokal (Karimah *et al.*, 2017). Dalam mencari solusi, algoritma genetika mampu menyelesaikan permasalahan yang kompleks dengan memanfaatkan fungsi probabilitas untuk mendapatkan solusi yang mendekati optimum. Proses penemuan solusi dalam algoritma genetika melibatkan berbagai pilihan titik optimal. Algoritma genetika dapat diterapkan dalam berbagai situasi, termasuk dalam masalah pencarian rute dengan jarak

minimum dari suatu titik awal ke beberapa titik tujuan, yang sering disebut sebagai TSP.

Algoritma genetika memungkinkan digunakan dalam optimasi masalah dengan ruang pencarian yang rumit dan luas karena menggunakan pendekatan berbasis populasi. Dengan demikian, algoritma genetika dapat mencapai solusi yang melampaui nilai solusi maksimum lokal. Implementasi algoritma genetika untuk menyelesaikan suatu masalah dilakukan dengan memperhatikan beberapa aspek penting, sebagai berikut:

1. Inisialisasi

Pada tahap awal, dilakukan inisialisasi populasi dengan menghasilkan sekumpulan kromosom yang merepresentasikan rute yang mungkin dilewati dalam permasalahan TSP. Populasi awal biasanya terdiri dari sejumlah kromosom acak yang dibangkitkan dengan memperhatikan batasan-batasan yang ada dalam permasalahan. Prosesnya yaitu membuat satu set solusi baru yang terdiri dari string kromosom maupun representasi kromosom.

2. *Crossover*

Proses *crossover* merupakan melibatkan proses pindah silang antara dua kromosom yang berperan sebagai *parent*. Tahapan ini membentuk kromosom baru dengan cara menyilangkan dua kromosom yang dipilih secara acak. Dalam proses persilangan tersebut, terbentuklah keturunan kromosom atau individu baru.

3. Evaluasi

Proses evaluasi dimana semua induk (*parent*) dan anak (*child*) digabungkan menjadi satu populasi, kemudian menghitung dan mengurutkan nilai *fitness* setiap individu.

4. Setelah terbentuk himpunan solusi baru dilakukan proses seleksi untuk menentukan solusi terbaik.

2.3 Metode *Non-Dominated Sorting*

Algoritma genetik seleksi non-dominasi adalah algoritma optimisasi multi-objektif pada sektor komputasi evolusioner. *Non-dominated genetic sorting algorithm* merupakan perpanjangan dari algoritma genetika guna mengoptimalkan fungsi *multi-objective* (Belluano, 2016). Ada dua versi algoritma yang banyak digunakan kala ini, NSGA klasik dan jenis terbaru saat ini, NSGA-II. Tujuan dari algoritma NSGA adalah guna memajukan penyesuaian adaptif populasi solusi kandidat ke *front pareto* yang dibatasi dengan aturan fungsi tujuan. Algoritma menerapkan metode evolusi menggunakan operator evolusi alternatif, termasuk seleksi, persilangan genetik, dan mutasi genetik. Populasi diklasifikasikan ke dalam tingkatan sub-populasi berdasarkan dominasi *pareto*. Kemudian, kesesuaian antara anggota dari masing-masing sub-kelompok dinilai di *front pareto*, dan kelompok yang dihasilkan serta ukuran kesetaraan digunakan untuk mempromosikan *front* yang beragam dari solusi yang tidak didominasi.

Deb *et al.*, (2002), mengusulkan pengurutan algoritma genetika tidak dominan dengan strategi elit yang merupakan algoritma evolusi *multi-objective* yang cepat dan elit berdasarkan pendekatan pengurutan tidak dominan. NSGA-II

mengoptimalkan semua tujuan secara simultan dengan memberikan dua atribut pada setiap solusi i . Pertama adalah peringkat pengurutan tidak-dominan (i_{rank}) yang didasarkan pada perilaku pengurutan tidak-dominan di antara solusi populasi F , dan yang lainnya adalah jarak *crowding distance* ($d_i(F_{i_{rank}})$) yang dihitung dengan mengikuti prosedur yang diberikan dan digunakan untuk mempertahankan keragaman populasi. Dengan dua atribut tersebut, operator perbandingan disajikan sebagai:

$$i < j \Leftrightarrow [(i_{rank} < j_{rank}) \quad (2.2)$$

atau

$$(i_{rank} = j_{rank} \text{ dan } d_i(F_{i_{rank}}) > d_j(F_{j_{rank}})) \quad (2.3)$$

Kemudian operator seleksi berdasarkan operator perbandingan dilakukan pada persilangan. Secara khusus, gabungkan populasi induk dan keturunan dan pilih solusi terbaik dengan memperhatikan kebugaran dan penyebaran. Strategi di atas memandu proses seleksi pada berbagai tahap algoritma menuju *front optimal pareto* yang merata. NSGA-II adalah pengoptimal *multi-objective* sederhana namun efisien dengan kompleksitas perhitungan (Li *et al.*, 2015).

2.4 Pendistribusian

Pendistribusian air minum merupakan kegiatan untuk memindahkan air dari sumber air ke tempat konsumen (Karimah *et al.*, 2017). Kegiatan pendistribusian dilakukan untuk memfasilitasi pengiriman barang dari produsen ke pelanggan dengan lancar dan mudah, sehingga barang dapat digunakan sesuai kebutuhan (Puteri *et al.*, 2017). Distribusi air minum yang efektif dan efisien sangat penting dalam memenuhi kebutuhan air minum masyarakat. Hampir semua perusahaan

mengadopsi kegiatan pendistribusian untuk meningkatkan produktivitasnya. Dalam proses pendistribusian, strategi yang tepat sangat penting untuk mengoptimalkan jalur dan memastikan proses pendistribusian berjalan dengan baik sehingga menguntungkan perusahaan. Tujuannya adalah untuk mencari jalur yang optimal agar proses pendistribusian barang menjadi lebih efektif. Proses distribusi di bagi menjadi 2 diantaranya :

1. Distribusi Langsung

Proses pendistribusian yang pada prosesnya berjalan langsung dari produsen ke konsumen tanpa memerlukan perantara lainnya.

2. Distribusi Tidak Langsung

Proses distribusi yang dilakukan oleh distributor yang bertanggung jawab dalam pendistribusian barang ke konsumen sebagai pihak ketiga atau perantara

BAB III

DESAIN PENELITIAN

3.1 Rangkaian Penelitian

Rangkaian penelitian dalam penelitian ini dilakukan dengan beberapa tahapan sebagai berikut:

1. Identifikasi masalah: Masalah yang dihadapi dalam penelitian ini adalah bagaimana mendistribusikan air minum ke wilayah-wilayah tertentu dengan menggunakan metode yang efektif dan efisien.
2. Pengumpulan data: Data yang digunakan dalam penelitian ini meliputi data koordinat lokasi wilayah yang akan didistribusikan air minum dan jarak dari titik awal menuju lokasi tujuan.
3. Pembuatan model: Pembuatan model dilakukan dengan menggabungkan permasalahan *travelling salesman problem* dan penjadwalan pengiriman air minum sehingga didapatkan model yang mampu menyelesaikan masalah pendistribusian air minum secara optimal.
4. Implementasi algoritma: Algoritma yang digunakan dalam penelitian ini adalah *non-dominated sorting genetic algorithm* (NSGA-II) yang telah dimodifikasi agar dapat menyelesaikan permasalahan *multi-objective travelling salesman problem*.
5. Evaluasi hasil: Hasil yang dihasilkan dari implementasi algoritma dievaluasi dengan menggunakan metode analisis kualitatif dan kuantitatif. Analisis kualitatif dilakukan dengan melakukan pengamatan terhadap rute yang

dihasilkan, sedangkan analisis kuantitatif dilakukan dengan menghitung nilai *fitness* dan *non-dominated solution*.

6. Kesimpulan: Pada tahap akhir dilakukan kesimpulan dari hasil penelitian dan saran untuk penelitian selanjutnya.

3.2 Data Collection

Berdasarkan Tabel 3.1 merupakan *dataset* yang digunakan untuk data distribusi data uji yang terdiri dari 5 kota serta jaraknya dalam satuan kilometer.

Tabel 3.1 *Dataset* Pendistribusian PT. Krakatau Daya Tirta

Distributor	Koordinat Longitude	Koordinat Latitude
PT. Krakatau Daya Tirta	105.982468	-5.993835
PT. Bumimerak Terminalindo	106.000384	-5.961781
PT. Purna Baja Harsco	105.996239	-6.006686
PT. Barata Indonesia (Persero)	106.019596	-6.007087
PT. Kostec Prima Baja	105.985254	-6.008665

3.3 Desain Arsitektur *Travelling Salesman Problem* (TSP)

Travelling salesman problem (TSP) merupakan permasalahan yang akan diselesaikan pada penelitian ini. TSP sendiri adalah permasalahan pengoptimasian rute kunjungan yang melibatkan beberapa kota dan titik-titik yang perlu dikunjungi dengan jarak yang berbeda-beda antara kota-kota tersebut. Pada penelitian ini, TSP digunakan untuk mendistribusikan air minum ke beberapa titik yang berbeda. Alur TSP dalam penelitian ini dapat dijelaskan sebagai berikut:

1. Penentuan data jarak antar titik

Pada awal alur TSP, data jarak antar titik harus ditentukan terlebih dahulu. Data jarak antar titik ini digunakan sebagai masukan untuk menghitung jarak antar titik pada setiap iterasi. Data ini dapat dilihat pada Tabel 3.1.

2. Inisialisasi populasi awal

Setelah data jarak antar titik ditentukan, langkah selanjutnya adalah melakukan inisialisasi populasi awal. Populasi awal terdiri dari kumpulan rute kunjungan yang diacak secara acak. Ini dirancang untuk menciptakan populasi awal dengan jumlah individu yang telah ditentukan saat nilai dimasukkan. Inisialisasi populasi awal ini dilakukan agar algoritma NSGA-II memiliki kumpulan solusi awal yang beragam. Berdasarkan data yang diperoleh, berikut ini adalah beberapa populasi awal dari rute yang diperoleh:

(1-2-3-4-5-1)

(1-4-3-5-2-1)

(1-3-4-2-5-1)

Apabila dikatakan *dataset* 1 adalah PT. Krakatau Daya Tirta dan sisanya adalah berurut sesuai dengan Tabel. Rute 1 adalah titik awal dan akan kembali pada titik semula dan setiap *salesman* hanya dapat mengunjungi setiap rute 1 kali.

3. Evaluasi populasi awal

Setelah populasi awal terbentuk, langkah selanjutnya adalah melakukan evaluasi terhadap setiap individu dalam populasi. Evaluasi ini dilakukan dengan menghitung total jarak yang ditempuh pada setiap rute kunjungan pada setiap individu. Lebih jelasnya mengenai perhitungan total jarak yang ditempuh adalah:

Diambil 6 rute secara acak yang akan dilalui seorang *salesman*

Bobot [1] = 1-2-3-4-5-1 = $11+7+1+2+4 = 25$

Bobot [2] = 1-3-4-5-2-1 = $5+1+2+9+11 = 28$

Bobot [3] = 1-4-5-2-3-1 = $5+2+9+7+5 = 28$

$$\text{Bobot [4]} = 1-5-2-3-4-1 = 4+9+7+1+5 = 26$$

$$\text{Bobot [5]} = 1-3-2-5-4-1 = 5+7+9+2+5 = 28$$

4. Seleksi genetik

Setelah populasi awal dievaluasi, langkah selanjutnya adalah melakukan seleksi genetik. Seleksi genetik dilakukan dengan memilih individu-individu terbaik dalam populasi. Individu-individu terbaik ini akan menjadi induk untuk membentuk populasi baru pada generasi selanjutnya. Dalam penentuan seleksi genetik, apabila tujuan penelitian untuk mencari rute dengan biaya terendah maka ditentukan rute dengan *fitness* paling rendah. Dalam hal ini contohnya adalah pada perhitungan evaluasi populasi awal diatas dimana bobot rute 1 memiliki *fitness* terendah.

5. Penerapan operator genetik

Setelah seleksi genetik, langkah selanjutnya adalah menerapkan operator genetik pada individu-individu terpilih. Operator genetik yang diterapkan meliputi *crossover* dan mutasi. *Crossover* digunakan untuk menghasilkan individu baru dengan cara menggabungkan materi genetik dari dua orang induk. Sedangkan mutasi digunakan untuk memperkenalkan variasi pada materi genetik individu.

Untuk contoh, diberikan dua kromosom sebagai berikut:

Individu 1: 1 -> 2 -> 3 -> 4 -> 5

Individu 2: 2 -> 4 -> 3 -> 5 -> 1

Maka kita bisa melakukan *crossover* dengan memilih sebuah titik potong secara acak seperti berikut:

Titik pemotongan 1: antara 2 dan 4

Titik pemotongan 2: antara 3 dan 5

Crossover antara dua individu:

Individu 1: 1 -> 2 -> 4 -> 3 -> 5

Individu 2: 2 -> 4 -> 3 -> 5 -> 1

Setelah proses *crossover*, kita dapat menerapkan operasi mutasi untuk memperkenalkan variasi genetik ke dalam populasi. Misalkan kita menggunakan operator mutasi *swap*, di mana dua kota dalam rute akan ditukar. Misalkan kita memilih individu kedua (setelah *crossover*) untuk dimutasi

Individu Baru: 2 -> 4 -> 3 -> 5 -> 1

Kemudian, jika kita menerapkan operasi mutasi dengan menggunakan operator mutasi *swap*, dua posisi dalam rute akan ditukar secara acak. Mari kita pilih dua posisi secara acak:

Posisi 1: Antara B dan D

Posisi 2: Antara C dan E

Setelah dilakukan mutasi, kita dapat menukar posisi B dengan posisi D, dan posisi C dengan posisi E:

Individu 2 (setelah mutasi): D -> B -> E -> C -> A

Dengan demikian, setelah proses mutasi, individu 2 berubah menjadi:

D -> B -> E -> C -> A

6. Evaluasi populasi baru

Setelah operator genetik diterapkan, langkah selanjutnya adalah melakukan evaluasi terhadap setiap individu dalam populasi baru. Evaluasi ini dilakukan

dengan menghitung total jarak yang ditempuh pada setiap rute kunjungan pada setiap individu.

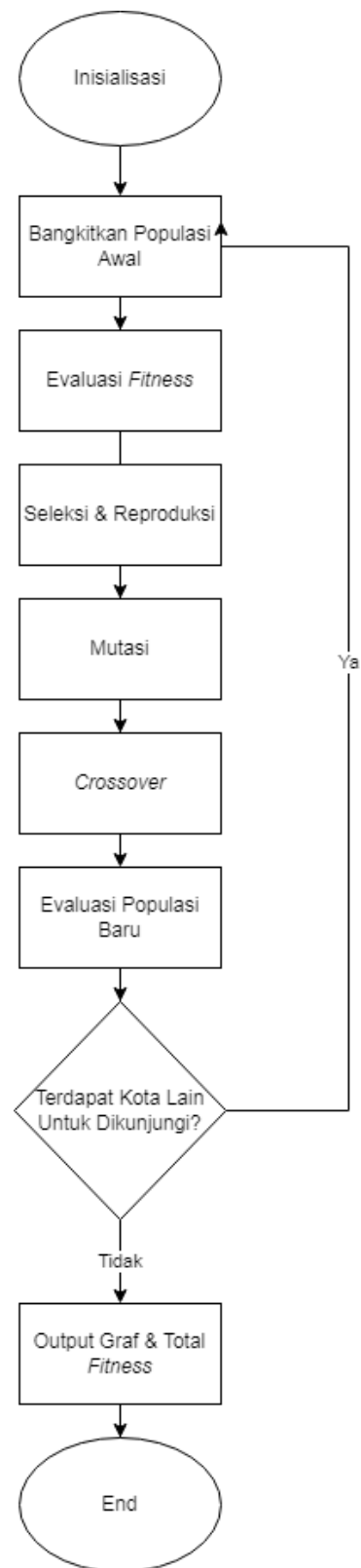
7. Seleksi elit

Setelah populasi baru dievaluasi, langkah selanjutnya adalah melakukan seleksi elit. Seleksi elit dilakukan dengan memilih individu-individu terbaik dalam populasi baru. Individu-individu terbaik ini akan disimpan sebagai solusi terbaik pada generasi tersebut.

8. Iterasi

Langkah-langkah di atas akan diulang dari langkah 4 hingga 7 sebanyak beberapa iterasi. Setiap iterasi akan menghasilkan populasi baru dan solusi terbaik pada generasi tersebut. Setelah semua kota dikunjungi oleh *salesman*, maka rute yang terbentuk akan dianalisis untuk mengevaluasi kinerja algoritma NSGA-II. Evaluasi dilakukan dengan menghitung nilai *fitness* rute, yaitu total jarak yang ditempuh oleh *salesman* dan banyaknya rute yang digunakan untuk pendistribusian air minum. Semakin kecil nilai *fitness*, maka semakin optimal rute yang terbentuk.

Selanjutnya, hasil evaluasi akan digunakan untuk menghasilkan populasi baru pada generasi berikutnya. Hal ini dilakukan dengan cara melakukan pemilihan individu-individu terbaik pada populasi saat ini berdasarkan nilai *fitness* dan melakukan operasi *crossover* dan mutasi pada individu-individu tersebut untuk menghasilkan individu baru pada generasi berikutnya. Algoritma NSGA-II akan diulang sebanyak iterasi yang ditentukan hingga didapatkan hasil yang optimal. Setelah mencapai iterasi terakhir, maka rute terbaik akan dipilih sebagai solusi optimal dari permasalahan TSP pada pendistribusian air minum.



Gambar 3.1 Flowchart alur Travelling Salesman Problem

3.4 Desain Arsitektur *Non-Dominated Sorting Genetic Algorithm-II* (NSGA-II)

Non-dominated sorting genetic algorithm-II (NSGA-II) adalah salah satu metode optimasi heuristik yang digunakan dalam penyelesaian masalah TSP *multi-objective*. Alur NSGA-II pada implementasi TSP dapat dijelaskan sebagai berikut:

1. Membangun populasi awal

Pada tahap ini, dibangun populasi awal dengan cara menginisialisasi kromosom secara acak. Setiap kromosom mewakili suatu jalur tur yang melintasi setiap kota tepat satu kali. Berikut adalah langkah-langkah dalam perhitungan untuk bangkitkan populasi awal secara acak dalam algoritma NSGA-II:

- a. Tentukan ukuran populasi awal yang diinginkan.
- b. Tentukan jumlah kota dalam permasalahan TSP.
- c. Acak urutan kota-kota tersebut untuk membentuk solusi potensial.
- d. Pastikan solusi potensial yang dihasilkan adalah solusi valid yang mencakup semua kota tanpa ada pengulangan kota.
- e. Setiap solusi potensial merepresentasikan sejumlah tur rute yang melintasi semua kota.
- f. Contohnya, jika terdapat 5 kota dengan label A, B, C, D, E, maka salah satu solusi potensial bisa menjadi [A, C, D, B, E, A].

Dengan melakukan langkah-langkah di atas untuk setiap individu dalam populasi, kita dapat menghasilkan populasi awal yang terdiri dari sejumlah solusi potensial yang diacak secara acak. Populasi awal ini kemudian akan digunakan sebagai titik awal dalam algoritma NSGA-II untuk mencari solusi-solusi yang lebih baik melalui proses seleksi, rekombinasi, dan mutasi.

2. Evaluasi fungsi objektif

Setiap kromosom pada populasi dievaluasi dengan fungsi objektif, yaitu jarak total yang harus ditempuh untuk menyelesaikan tur pada jalur tertentu. Terdapat beberapa kriteria atau objektif yang ingin dioptimalkan dalam permasalahan yang sedang diselesaikan. Misalnya, dalam permasalahan TSP, objektif utama adalah mencari rute dengan jarak tempuh yang minimal. Untuk setiap individu dalam populasi, lakukan langkah-langkah berikut:

- a. Hitung jarak tempuh atau nilai objektif yang ingin dioptimalkan berdasarkan solusi yang direpresentasikan oleh individu tersebut. Misalnya, jika jarak antara pasangan kota adalah $AC = 10$ km, $CD = 15$ km, $DB = 12$ km, $BE = 8$ km, dan $EA = 13$ km, maka total jarak atau nilai objektif yang ingin dioptimalkan adalah $10 + 15 + 12 + 8 + 13 = 58$ km
- b. Dalam kasus TSP, perhitungan jarak antara kota-kota biasanya menggunakan rumus *haversine* atau metode lainnya yang sesuai dengan permasalahan yang sedang dihadapi. Rumus *haversine* digunakan untuk menghitung jarak antara dua titik koordinat berdasarkan garis lintang dan bujur.

Rumus *haversine* digunakan untuk menghitung jarak antara dua titik pada permukaan bumi menggunakan garis lintang dan garis bujur. Rumus ini berguna dalam navigasi atau pemetaan lokasi berdasarkan koordinat geografis. Berikut adalah rumus *haversine*:

$$\text{haversin}(\theta) = \sin^2(\theta/2) = (1 - \cos(\theta))/2 \quad (2.4)$$

dengan:

- θ : sudut di antara dua titik dalam radian.
- sin: fungsi sinus.
- cos: fungsi kosinus.

Rumus Haversine yang lebih lengkap untuk menghitung jarak antara dua titik A (latA, lonA) dan B (latB, lonB) adalah sebagai berikut:

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{latA}) * \cos(\text{latB}) * \sin^2(\Delta\text{lon}/2)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

dengan:

- $\Delta\text{lat} = \text{latB} - \text{latA}$: selisih garis lintang antara dua titik dalam radian.
- $\Delta\text{lon} = \text{lonB} - \text{lonA}$: selisih garis bujur antara dua titik dalam radian.
- R: radius bumi (dalam satuan yang sama dengan jarak yang ingin dihasilkan, misalnya kilometer atau mil).

Hasil perhitungan d adalah jarak antara titik A dan B dalam satuan yang sama dengan R.

- c. Jumlahkan semua jarak antara pasangan kota untuk mendapatkan nilai total jarak atau objektif yang ingin dioptimalkan.
- d. Misalnya, jika terdapat solusi potensial [A, C, D, B, E, A] dengan jarak antara kota-kota sebagai berikut: AC = 10 km, CD = 15 km, DB = 12 km, BE = 8 km, dan EA = 13 km, maka total jaraknya adalah $10 + 15 + 12 + 8 + 13 = 58$ km.

Setelah melakukan perhitungan evaluasi *fitness* untuk setiap individu dalam populasi, nilai *fitness* individu-individu tersebut akan digunakan dalam langkah-langkah seleksi, rekombinasi, dan mutasi pada algoritma NSGA-II.

3. *Non-dominated sorting*

Populasi awal kemudian diurutkan menjadi beberapa *front* berdasarkan konsep *non-dominated sorting*. *Front-front* ini terdiri dari individu-individu yang tidak dapat saling mengalahkan satu sama lain dalam hal kemampuan untuk memenuhi kriteria optimasi. Setiap kromosom diberikan ranking, dan nilai *crowding distance* (jarak ke individu terdekat di dalam *front*) dihitung untuk masing-masing kromosom.

Contoh proses *non-dominated sorting* pada NSGA-II dapat dijelaskan sebagai berikut:

Populasi awal:

Individu 1: Kota A - Kota B - Kota C - Kota D - Kota E (Jarak: 100 km)

Individu 2: Kota B - Kota A - Kota D - Kota E - Kota C (Jarak: 120 km)

Individu 3: Kota C - Kota D - Kota A - Kota B - Kota E (Jarak: 90 km)

Individu 4: Kota D - Kota B - Kota C - Kota A - Kota E (Jarak: 110 km)

Individu 5: Kota E - Kota C - Kota A - Kota B - Kota D (Jarak: 130 km)

Langkah-langkah ranking dan penyortiran *non-dominated*:

1. Evaluasi dominasi:

- a. Membandingkan setiap individu dengan individu lainnya berdasarkan nilai kriteria.

- b. Jika ada individu lain yang memiliki nilai kriteria yang lebih baik atau setidaknya sama baik dalam semua kriteria, maka individu tersebut didominasi.

2. Pembentukan lapisan *pareto*:

- a. Membentuk lapisan *pareto* berdasarkan dominasi.
- b. Individu yang tidak didominasi langsung ditempatkan pada lapisan *pareto* pertama (lapisan teratas).
- c. Individu yang didominasi oleh satu atau lebih individu ditempatkan pada lapisan *pareto* berikutnya.

Contoh hasil ranking dan penyortiran *non-dominated*:

Lapisan *pareto* 1:

- a. Individu 1: Kota A - Kota B - Kota C - Kota D - Kota E (Jarak: 100 km)
- b. Individu 3: Kota C - Kota D - Kota A - Kota B - Kota E (Jarak: 90 km)

Lapisan *pareto* 2:

- a. Individu 2: Kota B - Kota A - Kota D - Kota E - Kota C (Jarak: 120 km)
- b. Individu 4: Kota D - Kota B - Kota C - Kota A - Kota E (Jarak: 110 km)

Lapisan *pareto* 3:

- a. Individu 5: Kota E - Kota C - Kota A - Kota B - Kota D (Jarak: 130 km)

Dalam contoh ini, Individu 1 dan Individu 3 berada pada lapisan *pareto* pertama karena tidak didominasi oleh individu lain. Individu 2 dan Individu 4 didominasi oleh Individu 1 dan Individu 3, sehingga berada pada lapisan *pareto* kedua. Individu 5 didominasi oleh semua individu lainnya, sehingga berada pada lapisan *pareto* ketiga.

Ranking dan penyortiran *non-dominated* membantu mengidentifikasi individu yang mendekati solusi optimal dan memiliki trade-off yang berbeda antara kriteria jarak distribusi.

4. Seleksi orang tua (*parent selection*)

Pada tahap ini, kromosom dipilih sebagai orang tua untuk generasi berikutnya menggunakan teknik *tournament selection*. Kromosom-kromosom dengan peringkat *front* yang lebih rendah dan nilai *crowding distance* yang lebih tinggi memiliki peluang lebih besar untuk dipilih.

Contoh langkah-langkah pada tahap seleksi individu dapat dijelaskan di bawah ini. Misalkan terdapat tiga lapisan *pareto* hasil dari tahap ranking dan penyortiran *non-dominated*. Setelah perhitungan *crowding distance*, didapatkan nilai *crowding distance* untuk setiap individu dalam lapisan *pareto* seperti berikut:

Lapisan *Pareto* 1:

- a. Individu 1: Kota A - Kota B - Kota C - Kota D - Kota E (Jarak: 100 km)
- b. Individu 3: Kota C - Kota D - Kota A - Kota B - Kota E (Jarak: 90 km)

Lapisan *Pareto* 2:

- a. Individu 2: Kota B - Kota A - Kota D - Kota E - Kota C (Jarak: 120 km)
- b. Individu 4: Kota D - Kota B - Kota C - Kota A - Kota E (Jarak: 110 km)

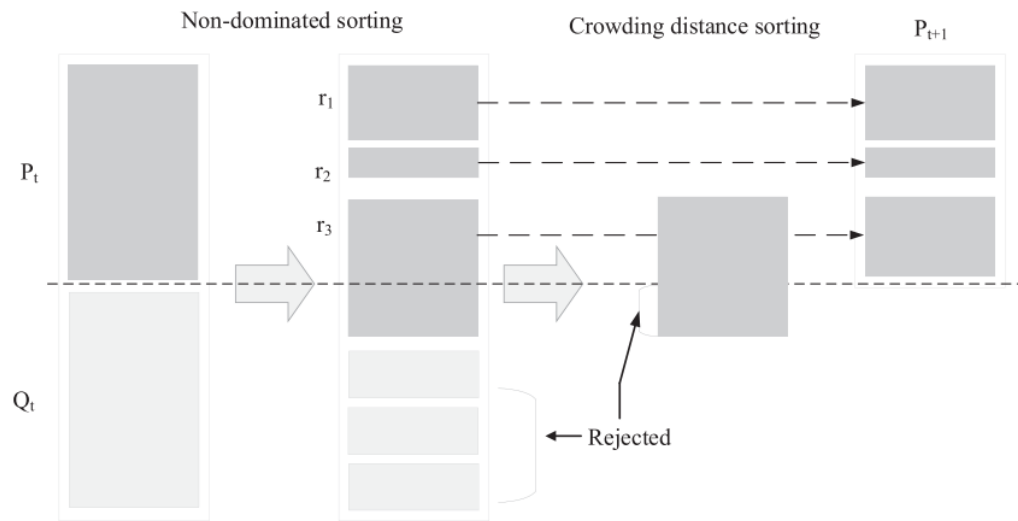
Lapisan *Pareto* 3:

- a. Individu 5: Kota E - Kota C - Kota A - Kota B - Kota D (Jarak: 130 km)

Selanjutnya, untuk memilih individu-individu yang akan menjadi bagian dari generasi berikutnya, langkah-langkah yang dapat diikuti adalah:

- a. Pertama, individu dengan ranking yang lebih tinggi dipilih terlebih dahulu. Misalkan dalam contoh ini, individu-individu di lapisan *pareto* 1 memiliki ranking yang lebih tinggi daripada lapisan *pareto* 2 dan 3.
- b. Kemudian, jika ada kapasitas tersisa dalam populasi baru, individu dengan *crowding distance* yang lebih tinggi juga dipilih untuk memberikan variasi dalam populasi dan potensi solusi alternatif.
- c. Dalam contoh ini, misalkan hanya ingin mempertahankan 4 individu dalam generasi berikutnya.
 - Individu 3 (lapisan *pareto* 1) dipilih karena memiliki ranking tertinggi dan *crowding distance* yang tinggi.
 - Individu 1 (lapisan *pareto* 1) dipilih karena memiliki ranking tinggi dan *crowding distance* yang lebih rendah dibandingkan dengan individu lain di lapisan *pareto* 1.
 - Individu 4 (lapisan *pareto* 2) dipilih karena *crowding distance* yang tinggi.
 - Individu 5 (lapisan *pareto* 3) dipilih karena *crowding distance* yang tinggi.

Keempat individu tersebut akan menjadi bagian dari generasi berikutnya dan melanjutkan ke tahap operasi *crossover* dan mutasi pada iterasi berikutnya.



Gambar 3.2 Proses *Non-dominated sorting* menuju *Crowding Distance sorting*

5. Pencampuran (*crossover*) dan mutasi (*mutation*)

Setelah seleksi orang tua dilakukan, pencampuran (*crossover*) dan mutasi (*mutation*) diterapkan pada kromosom orang tua untuk menghasilkan kromosom anak. Teknik *crossover* yang digunakan pada penelitian ini adalah *order crossover* (OX). Berikut adalah penyelesaian *crossover*:

Misalkan terdapat dua *parent* yaitu *parent A* dan *parent B*, dengan kromosom mereka adalah sebagai berikut:

Parent A: 1-2-3-4-5-6-7-8-9-10

Parent B: 10-9-8-7-6-5-4-3-2-1

a. Pemilihan titik *crossover*

Titik *crossover* harus dipilih secara acak. Misalkan titik *crossover* yang terpilih adalah 3.

b. Proses *crossover*

Proses *crossover* menghasilkan dua *offspring*:

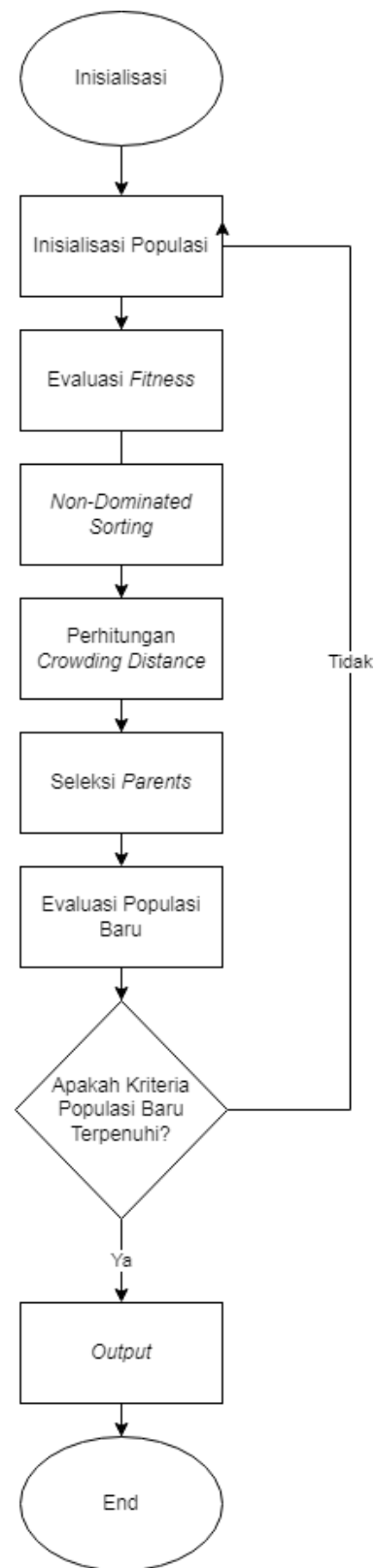
- *Offspring* 1: Ambil bagian genetik dari *parent* A sebelum titik pemotongan dan sisa genetik dari *parent* B setelah titik pemotongan.
- *Offspring* 1: 1-2-3-7-6-5-4-3-2-1
- *Offspring* 2: Ambil bagian genetik dari *parent* B sebelum titik pemotongan dan sisa genetik dari *parent* A setelah titik pemotongan.
- *Offspring* 2: 10-9-8-4-5-6-7-8-9-10

c. Proses mutasi

Setelah tahap *crossover*, langkah selanjutnya adalah melakukan operasi mutasi pada *offspring* yang dihasilkan. Mutasi bertujuan untuk memperkenalkan variasi genetik tambahan ke dalam populasi. Pada langkah ini, gen-gen individu *offspring* dipilih secara acak dan diubah nilainya.

Contoh alur mutasi:

- Misalkan kita mengambil *offspring* 1 yang telah dihasilkan sebelumnya: 1-2-3-7-6-5-4-3-2-1
- Pilih gen secara acak untuk dimutasi. Misalnya, pilih gen pada posisi 4.
- Ganti nilai gen tersebut dengan nilai acak yang valid dalam rentang genetik yang diperbolehkan. Misalnya, ganti nilai gen pada posisi 4 dengan nilai 8.
- *Offspring* 1 setelah proses mutasi: 1-2-3-8-6-5-4-3-2-1
- Setelah proses mutasi, individu-*offspring* yang telah dimutasi tersebut dapat dimasukkan ke dalam populasi dan digunakan untuk iterasi berikutnya. Proses akan diulang dalam setiap generasi hingga mencapai kondisi terminasi.



Gambar 3.3 Flowchart alur NSGA-II

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Berikut adalah beberapa langkah uji coba yang dapat dilakukan untuk mengevaluasi performa dari sistem yang diimplementasikan:

- a. Persiapan data: Persiapkan data yang akan digunakan dalam pengujian, termasuk data koordinat lokasi, dan data jarak lokasi.
- b. Pembuatan rute: Buat rute yang dapat ditempuh oleh kendaraan dengan menggunakan algoritma TSP atau NSGA-II. Simpan hasil rute pada file eksternal.
- c. Uji coba kecepatan: Uji coba kecepatan sistem dengan mengukur waktu yang diperlukan untuk menghasilkan rute terbaik. Uji coba ini dapat dilakukan dengan membandingkan waktu yang diperlukan untuk menghasilkan rute dengan menggunakan TSP dan NSGA-II.
- d. Uji coba akurasi: Uji coba akurasi sistem dengan membandingkan hasil rute yang dihasilkan dengan rute optimal yang telah diketahui sebelumnya.
- e. Uji coba kehandalan: Uji coba kehandalan sistem dengan memeriksa apakah sistem dapat menghasilkan rute yang konsisten dan dapat diandalkan dalam berbagai kondisi pengujian yang berbeda.
- f. Uji coba skalabilitas: Uji coba skalabilitas sistem dengan mengukur performa sistem pada *dataset* yang lebih besar. Uji coba ini dapat dilakukan dengan menguji sistem pada *dataset* yang memiliki jumlah titik yang lebih banyak dari dataset yang telah digunakan sebelumnya.

- g. Analisis hasil: Analisis hasil dari uji coba untuk mengevaluasi kekuatan dan kelemahan dari sistem yang telah diimplementasikan. Identifikasi faktor-faktor yang dapat mempengaruhi performa sistem dan ajukan saran untuk perbaikan dan pengembangan sistem di masa depan.

4.1.1 Alur TSP & NSGA-II Pada Sistem

Alur NSGA-II pada sistem terdiri dari beberapa tahapan utama yang dilakukan untuk mendapatkan solusi terbaik pada permasalahan TSP dalam pendistribusian air minum. Berikut adalah penjelasan mengenai alur NSGA-II pada sistem:

a. Inisialisasi Populasi

Pada tahap awal, dilakukan inisialisasi populasi dengan menghasilkan sekumpulan kromosom yang merepresentasikan rute yang mungkin dilewati dalam permasalahan TSP. Populasi awal terdiri dari sejumlah kromosom acak yang dibangkitkan dengan memperhatikan batasan-batasan yang ada dalam permasalahan. Berikut adalah penjelasan mengenai *pseudocode* pada inisialisasi populasi:

```

1 def bangkitkan_populasi_awal(jumlah_kota,
2   ukuran_populasi):
3     populasi_awal = []
4     for _ in range(ukuran_populasi):
5         rute = list(range(1, jumlah_kota + 1))
6         random.shuffle(rute)
7         populasi_awal.append(rute)
8     return populasi_awal

```

Metode ini digunakan untuk menghasilkan rute perjalanan individu secara acak. Menggunakan fungsi *random.shuffle()* untuk memilih sejumlah kota dari *cities* sebanyak jumlah kota yang ada.

b. Evaluasi *Fitness*

Selepas populasi awal terwujud, dievaluasi nilai *fitness* dari setiap kromosom dalam populasi yaitu seberapa baik dari kromosom tersebut dalam memecahkan persoalan TSP. Semakin kecil nilai *fitness* maka semakin efektif kromosom tersebut. Berikut adalah *pseudocode* dari fungsi evaluasi *fitness*.

```

1     def evaluasi_fitness(populasi,
2     koordinat_kota):
3         fitness_populasi = []
4         for individu in populasi:
5             panjang_rute =
6             hitung_panjang_rute(individu, koordinat_kota)
7             fitness_populasi.append((individu,
8             panjang_rute))
9         return fitness_populasi

```

Metode ini digunakan untuk menghitung nilai kecocokan (*fitness*) individu. Menggunakan variabel *distance* yang diinisialisasi dengan 0. Melakukan iterasi melalui rute perjalanan individu menggunakan *loop for*, dan untuk setiap kota dalam rute, menghitung jarak ke kota berikutnya menggunakan metode *distance_to* dari objek *city*. Jarak tersebut ditambahkan ke variabel *distance*. Setelah itu, jarak dari kota terakhir ke kota pertama juga ditambahkan ke *distance*. Nilai *distance* inilah yang akan menjadi *fitness* individu.

c. *Non-Dominated Sorting*

Pada tahap ini, setiap kromosom dalam populasi dikelompokkan ke dalam beberapa *front* non-dominan, yaitu kelompok kromosom-kromosom yang saling tidak dominan satu sama lain. *Front* pertama terdiri dari kromosom-kromosom yang tidak didominasi oleh kromosom lain dalam populasi, *front* kedua terdiri dari kromosom-kromosom yang hanya didominasi oleh kromosom di *front* pertama, dan seterusnya. Kromosom-kromosom yang berada di dalam *front* yang sama memiliki nilai *fitness* yang sebanding. Berikut adalah penjelasan mengenai *pseudocode* alur tersebut:

```

1      def tsp_nsga2(koordinat_kota,
2      ukuran_populasi, probabilitas_mutasi,
3      jumlah_generasi):
4          jumlah_kota = len(koordinat_kota)
5          populasi =
6      bangkitkan_populasi_awal(jumlah_kota,
7      ukuran_populasi)
8
9      for _ in range(jumlah_generasi):
10         populasi_baru = []
11         elit = seleksi_elit(populasi,
12 koordinat_kota, ukuran_populasi // 2)
13
14         while len(populasi_baru) <
15 ukuran_populasi:
16             parent1 = random.choice(elit)
17             parent2 = random.choice(elit)
18             anak = crossover(parent1, parent2)
19             anak_mutasi = mutasi(anak,
20 probabilitas_mutasi)
21             populasi_baru.append(anak_mutasi)
22
23         populasi = populasi_baru

```

1. $\text{Jumlah_kota} = \text{len}(\text{koordinat_kota})$

Menghitung jumlah kota berdasarkan koordinat kota yang diberikan.

2. Inisialisasi $\text{populasi} = \text{bangkitkan_populasi_awal}(\text{jumlah_kota}, \text{ukuran_populasi})$

Membangkitkan populasi awal secara acak dengan ukuran sebanyak ukuran_populasi . Setiap individu dalam populasi merepresentasikan solusi potensial dalam bentuk rute.

3. *Loop* generasi

Selanjutnya, dilakukan *loop* sebanyak '*generations*' untuk melakukan iterasi melalui generasi.

4. Sortir *non-dominated*

Setelah selesai melakukan *looping* generasi, melakukan evaluasi *fitness* dan perhitungan *crowding distance* terakhir pada populasi.

5. Penggantian populasi

Mengembalikan populasi, *fitness* populasi, dan *crowding distances* sebagai *output* dari fungsi.

6. *Return* hasil terbaik

Pada akhir algoritma, diambil rute terbaik dan nilai kecocokan (*fitness*) terbaik dari populasi.

d. Crowding Distance

Setiap kromosom dalam setiap *front* kemudian diberikan nilai *crowding distance*, yaitu seberapa rapat kromosom tersebut dengan kromosom lain dalam *front* yang sama. Kromosom dengan *crowding distance* yang tinggi menunjukkan

bahwa kromosom tersebut terletak pada daerah yang padat dalam ruang solusi, sehingga penting untuk dilestarikan. Berikut adalah penjelasan mengenai *pseudocode* alur tersebut:

```

1      def
2  hitung_crowding_distance(fitness_populasi):
3      n = len(fitness_populasi)
4      crowding_distances = [0] * n
5      num_objectives = len(fitness_populasi[0][1])
6
7      for obj_idx in range(num_objectives):
8          fitness_populasi.sort(key=lambda x:
9  x[1][obj_idx])
10         crowding_distances[0] =
11 crowding_distances[n - 1] = float('inf')
12
13         min_fitness =
14 fitness_populasi[0][1][obj_idx]
15         max_fitness = fitness_populasi[n -
16 1][1][obj_idx]
17         if max_fitness == min_fitness:
18             continue
19
20         for i in range(1, n - 1):
21             crowding_distances[i] +=
22 (fitness_populasi[i+1][1][obj_idx] -
23 fitness_populasi[i-1][1][obj_idx]) / (max_fitness
24 - min_fitness)
25
26         return crowding_distances

```

Pada fungsi *crowding_distance_assignment*, kita menghitung jarak penyebaran (*crowding distance*) setiap individu pada *front* dengan membandingkan posisi individu dalam urutan pemotongan di setiap dimensi dari rute mereka. Semakin besar jarak penyebaran individu, semakin baik individu tersebut dalam hal kekerapan. Individu dengan jarak penyebaran yang lebih besar akan lebih mungkin dipertahankan dalam seleksi generasi berikutnya.

e. Reproduksi

Kromosom-kromosom terbaik yang terpilih pada tahap seleksi kemudian dipergunakan untuk melakukan operasi reproduksi, yaitu *crossover* dan mutasi, untuk menghasilkan kromosom-kromosom baru yang akan membentuk populasi generasi selanjutnya. Pada operasi *crossover*, dua kromosom dipilih secara acak dan digabungkan untuk membentuk kromosom-kromosom baru. Sedangkan pada operasi mutasi, kromosom-kromosom yang terpilih mengalami perubahan kecil pada beberapa bagian gen-nya. Berikut adalah penjelasan mengenai *pseudocode* alur tersebut:

```

1      populasi = populasi_baru
2
3      fitness_populasi = evaluasi_fitness(populasi,
4      koordinat_kota)
5      crowding_distances =
6      hitung_crowding_distance(fitness_populasi)
7
8      # Mengembalikan populasi, fitness populasi,
9      dan crowding distances
10     return populasi, fitness_populasi,
11     crowding_distances

```

4.1.2 Implementasi *Design Interface*

Pada penelitian ini, antarmuka dibangun dengan menggunakan bahasa pemrograman Python dan modul *streamlit*. Antarmuka dibuat agar pengguna dapat memasukkan parameter yang diperlukan untuk perhitungan optimasi, seperti jumlah pelanggan, dan data jarak lokasi. Selain itu, antarmuka juga menampilkan hasil dari perhitungan optimasi yang dilakukan oleh algoritma NSGA-II, seperti nilai *fitness*, jarak tempuh, dan visualisasi perhitungan.

Antarmuka pada penelitian ini dirancang dengan sederhana namun informatif. Hal ini dilakukan agar pengguna dapat dengan mudah memahami dan menggunakan antarmuka tersebut. Antarmuka juga dirancang agar dapat menampilkan hasil perhitungan secara *real-time*, sehingga pengguna dapat langsung melihat hasil dari setiap perubahan parameter yang dimasukkan.

Dalam proses implementasi *Design Interface*, dilakukan pengujian terhadap antarmuka yang telah dibangun untuk memastikan antarmuka tersebut dapat berfungsi dengan baik dan menampilkan hasil yang akurat dari perhitungan optimasi yang dilakukan oleh algoritma NSGA-II. Berikut adalah *design interface* pada sistem:

a. Tampilan Utama

Tampilan utama merupakan tempat untuk melakukan proses pemasukan data ke dalam sistem. Dalam penginputan data ini, terdapat menu yang harus diisi yaitu Jumlah kota, Nama kota dan *button* “Lakukan Perhitungan”

Travelling Salesman Problem

with NSGA-II Algorithm

Masukkan jumlah kota:

5

Nama kota 1: PT. Krakatau Daya Tirta

Nama kota 2: PT. Kostec Prima Baja

Nama kota 3: PT. Purna Baja Harsco

Nama kota 4: PT. Barata Indonesia

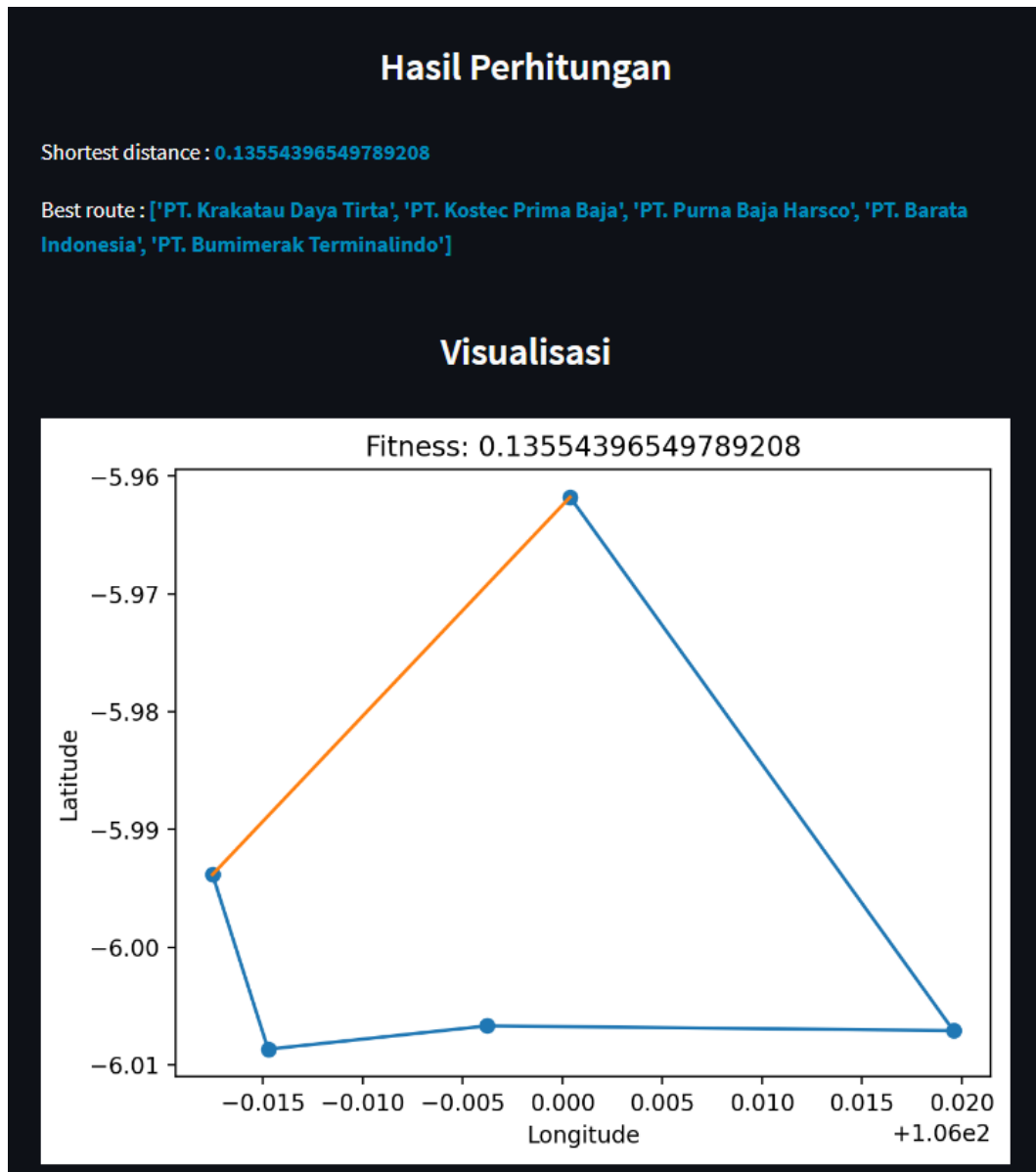
Nama kota 5: PT. Bumimerak Terminalindo

Lakukan Perhitungan

Gambar 4.1 Tampilan Utama

b. Visualisasi Grafik

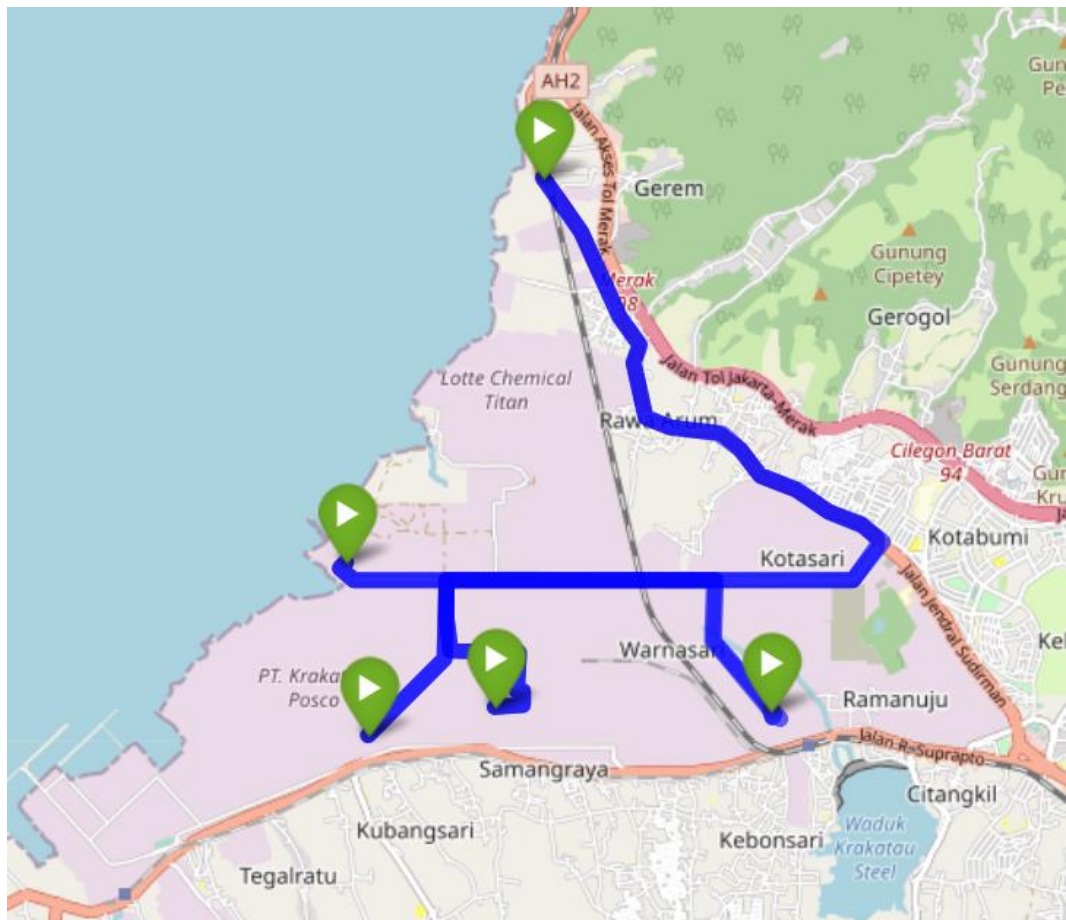
Ditampilkan hasil visualisasi grafik yang didapatkan dari hasil simulasi pendistribusian air minum menggunakan metode *non-dominated sorting genetic algorithm* (NSGA-II) dalam penyelesaian *travelling salesman problem* (TSP). Hasil visualisasi grafik tersebut meliputi grafik antara *fitness* value dan garis *polyline* antar titik.



Gambar 4.2 Visualisasi Grafik

c. Visualisasi Maps

Pada visualisasi peta ini, akan ditampilkan teknik-teknik visualisasi yang digunakan dalam penelitian ini, seperti pemberian garis *polyline* pada rute yang dilalui, pemberian *marker* pada titik-titik lokasi, dan lain sebagainya.



Gambar 4.3 Visualisasi Maps

4.2 Uji Coba Sistem

Setelah tahap implementasi selesai dilakukan, tahap selanjutnya adalah melakukan uji coba sistem. Uji coba sistem bertujuan untuk memastikan bahwa sistem yang telah dibangun berjalan dengan baik dan menghasilkan hasil yang akurat sesuai dengan yang diharapkan. Pada penelitian ini, uji coba sistem dilakukan dengan menggunakan data sampel yang telah disiapkan. Data sampel tersebut digunakan untuk melakukan perhitungan optimasi dengan algoritma NSGA-II. Hal ini dilakukan untuk memastikan bahwa algoritma NSGA-II dapat menghasilkan solusi yang lebih baik dan efisien dalam menyelesaikan

permasalahan pendistribusian air minum dengan menggunakan TSP sebagai dasar penghitungan rute distribusi.

Selama proses uji coba sistem, dilakukan pengujian terhadap antarmuka yang telah dibangun untuk memastikan antarmuka tersebut dapat berfungsi dengan baik dan menampilkan hasil yang akurat dari perhitungan optimasi yang dilakukan oleh algoritma NSGA-II. Selain itu, dilakukan pengujian terhadap kecepatan perhitungan dan kinerja sistem untuk memastikan bahwa sistem dapat menghasilkan hasil perhitungan dengan waktu yang efisien dan cepat. Dari hasil uji coba sistem yang dilakukan, diharapkan dapat terbukti bahwa sistem yang telah dibangun dapat berjalan dengan baik dan menghasilkan solusi yang lebih baik dan efisien dalam menyelesaikan permasalahan pendistribusian air minum dengan menggunakan TSP sebagai dasar penghitungan rute distribusi.

4.2.1 Pengujian

Proses pengujian sistem dilakukan pada proses ketika *input data*, data-data yang digunakan untuk rute pendistribusian dimulai dengan titik awal dari PT. Krakatau Daya Tirta. Berikut beberapa tujuan pendistribusian yang divisualisasi dengan angka adalah sebagai berikut:

1. PT. Krakatau Daya Tirta
2. PT. Bumimerak Terminalindo
3. PT. Purna Baja Harsco
4. PT. Barata Indonesia
5. PT. Kostec Prima Baja

Berikut adalah contoh data lengkap dari beberapa rute pelanggan yang dapat dilalui dan diinput ke dalam sistem yang tersajikan pada Tabel 4.1

Tabel 4.1 Rute yang dapat dilalui oleh *Salesman*

No	Rute	No	Rute	No	Rute
1	1-2-3-4-5-1	10	1-3-4-5-2-1	19	1-5-2-3-4-1
2	1-2-3-5-4-1	11	1-3-5-2-4-1	20	1-5-2-4-3-1
3	1-2-4-3-5-1	12	1-3-5-4-2-1	21	1-5-3-2-4-1
4	1-2-4-5-3-1	13	1-4-2-3-5-1	22	1-5-3-4-2-1
5	1-2-5-3-4-1	14	1-4-2-5-3-1	23	1-5-4-2-3-1
6	1-2-5-4-3-1	15	1-4-3-2-5-1	24	1-5-4-3-2-1
7	1-3-2-4-5-1	16	1-4-3-5-2-1		
8	1-3-2-5-4-1	17	1-4-5-2-3-1		
9	1-3-4-2-5-1	18	1-4-5-3-2-1		

4.2.2 Hasil Uji Coba

Setelah melakukan pengujian terhadap sistem, diperoleh beberapa hasil yang dapat dijadikan evaluasi terhadap sistem yang telah diimplementasikan. Berikut adalah hasil uji coba yang telah dilakukan

Tabel 4.2 Pengujian jarak dari rute yang dapat dilalui

No	Rute	Jarak (Km)	No	Rute	Jarak (Km)	No	Jarak (Km)	Jarak (Km)
1	1-2-3-4-5-1	35	10	1-3-4-5-2-1	40,1	19	1-5-2-3-4-1	35
2	1-2-3-5-4-1	38,4	11	1-3-5-2-4-1	37,4	20	1-5-2-4-3-1	34
3	1-2-4-3-5-1	30,8	12	1-3-5-4-2-1	37,5	21	1-5-3-2-4-1	32,4
4	1-2-4-5-3-1	37,3	13	1-4-2-3-5-1	32,4	22	1-5-3-4-2-1	30,8
5	1-2-5-3-4-1	36,7	14	1-4-2-5-3-1	37,4	23	1-5-4-2-3-1	35,7
6	1-2-5-4-3-1	40	15	1-4-3-2-5-1	35	24	1-5-4-3-2-1	35
7	1-3-2-4-5-1	35,7	16	1-4-3-5-2-1	36,7			
8	1-3-2-5-4-1	41,7	17	1-4-5-2-3-1	41,7			
9	1-3-4-2-5-1	34,1	18	1-4-5-3-2-1	38,4			

4.3 Pembahasan

Dari hasil pengujian yang telah dilakukan, didapatkan rute terbaik yang dapat meminimalkan jarak tempuh untuk mengunjungi seluruh kota tujuan yaitu dengan melalui rute PT. Krakatau Daya Tirta - PT. Kostec Prima Baja - PT. Purna Baja Harsco - PT. Barata Indonesia - PT. Bumimerak Terminalindo dengan nilai *fitness* 0.13554396549789208 dari total jarak 30,8 km. Untuk menghitung perbandingan efektifitas, dapat dilakukan dengan perbandingan nilai *fitness* pada penelitian yang digagaskan oleh Karimah *et al.*, (2017) menggunakan algoritma genetika. Dalam penelitian tersebut, perhitungan TSP menggunakan algoritma genetika menghasilkan selisih total jarak sebanyak 89.3 km dan selisih nilai *fitness* 10.656578. Untuk menghitung efektivitas masing-masing penelitian, kita dapat menggunakan perbandingan nilai *fitness* terhadap total jarak yang ditempuh. Efektivitas dapat diukur dengan menghitung nilai *fitness* per kilometer.

1. Metode Penelitian 1 (Algoritma Genetika):

$$\text{Efektivitas} = \frac{\text{Nilai Fitness}}{\text{Total Jarak}}$$

$$\text{Efektivitas Penelitian 1} = \frac{10,656578}{89,3} = 0,1193$$

2. Metode Penelitian 2 (NSGA-II):

$$\text{Efektivitas} = \frac{\text{Nilai Fitness}}{\text{Total Jarak}}$$

$$\text{Efektivitas Penelitian 2} = \frac{0,1355439654978208}{30,8} = 0,0044$$

3. Persentase Efektivitas

Efektivitas Penelitian 1 – Efektivitas Penelitian 2

$$0,1193 - 0,0044 = 0,1149$$

Untuk menghitung persentase dari suatu nilai terhadap nilai lainnya, dapat menggunakan rumus berikut:

$$\text{Persentase} = \left(\frac{\text{Nilai persentase}}{\text{Nilai total}} \right) \times 100$$

$$\text{Persentase} = \left(\frac{0,1149}{0,1193} \right) \times 100$$

$$\text{Persentase} = 96,17 \%$$

Jadi, efektivitas penelitian pertama dengan menggunakan algoritma genetika adalah sekitar 0.119529 *fitness* per kilometer sementara efektivitas penelitian kedua dengan menggunakan NSGA-II adalah sekitar 0.0044 *fitness* per kilometer. Dengan cara ini, dapat dihasilkan perbandingan efektivitas kedua penelitian dalam konteks perbandingan nilai *fitness* per kilometer yang ditempuh. Semakin kecil nilai ini, semakin baik efektivitasnya. Lalu dapat disimpulkan dari perbandingan nilai tersebut, metode NSGA-II menghasilkan efektivitas yang lebih baik sebesar 96.17%. Dalam perbandingan kedua metode tersebut, diketahui juga dengan kecepatan rata-rata 28 km per jam, penelitian menggunakan metode Algoritma Genetika memperoleh waktu 3 jam 10 menit dalam total jarak 89,3 km. Sementara itu penelitian ini yang menggunakan metode NSGA-II dengan rata-rata kecepatan yang sama yaitu 28 km per jam, dihasilkan waktu 1 jam 6 menit dalam total jarak 30,8 km.

Dalam implementasi NSGA-II, terdapat beberapa parameter yang dapat diatur seperti ukuran populasi, jumlah generasi, dan ukuran turnamen dalam seleksi

individu. Pengujian dilakukan dengan variasi nilai-nilai parameter tersebut untuk mencari konfigurasi terbaik yang dapat menghasilkan rute optimal dengan cepat. Hasil pengujian menunjukkan bahwa NSGA-II mampu memberikan rute terbaik dengan jarak tempuh yang lebih efisien dibandingkan metode lain yang digunakan sebelumnya. Selain itu, hasil pengujian juga menunjukkan bahwa NSGA-II memiliki performa yang relatif stabil dan konsisten dalam mencari solusi optimal pada berbagai variasi parameter. Hal ini menunjukkan bahwa NSGA-II adalah metode yang dapat diandalkan dalam menyelesaikan permasalahan TSP pada pendistribusian air minum dengan mempertimbangkan banyaknya kota yang harus dikunjungi dan jarak tempuh antar kota.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dalam penelitian ini, implementasi NSGA-II dapat menghasilkan solusi-solusi yang optimal atau mendekati optimal untuk TSP pada pendistribusian air minum. Metode NSGA-II mampu menghasilkan populasi individu yang terdiri dari rute-rute distribusi air minum yang berbeda dan kemudian mengaplikasikan operasi genetika seperti seleksi turnamen, *crossover*, dan mutasi untuk memperbaiki populasi dari generasi ke generasi. Penggunaan mekanisme *non-dominated sorting* dan *crowding distance assignment* dalam NSGA-II memungkinkan adanya pemilihan individu yang saling mengimbangi antara kriteria-kriteria yang ingin dioptimalkan.

Hasil penelitian menunjukkan bahwa implementasi NSGA-II mampu membantu menganalisis efektivitas dan efisiensi pendistribusian air minum dengan menghasilkan rute-rute distribusi air minum yang efisien dan optimal. Berdasarkan hasil uji yang diterapkan tersebut, dihasilkan rute terbaik yaitu PT. Krakatau - Daya Tirta - PT. Kostec Prima Baja - PT. Purna Baja Harsco - PT. Barata Indonesia - PT. Bumimerak Terminalindo dengan jarak total 30,8 km dengan waktu yang ditempuh adalah 1 jam 6 menit dengan kecepatan rata-rata 28 km per jam. Dalam uji tersebut, NSGA-II mampu mencapai efektivitas dengan metode lain sebesar 96,17% yang dapat memberikan pilihan kepada pengambil keputusan dalam memilih rute distribusi yang paling sesuai dengan preferensi mereka. Selain itu, NSGA-II juga menunjukkan kemampuan untuk mengatasi permasalahan kompleks seperti TSP.

5.2 Saran

Dengan mempertimbangkan faktor-faktor seperti jarak dan waktu, NSGA-II mampu memberikan solusi yang optimal. Dalam penelitian mendatang, diharapkan adanya upaya pengembangan lebih lanjut dalam hal pemilihan parameter dan studi perbandingan dengan metode lainnya guna meningkatkan efisiensi dan efektivitas pendistribusian air minum.

DAFTAR PUSTAKA

- Andri, A., Suyandi, S., & Win, W. (2013). Aplikasi Travelling Salesman Problem dengan Metode Artificial Bee Colony. *Jurnal SIFO Mikroskil*, 14(1), 59–68. <https://doi.org/10.55601/jsm.v14i1.92>
- Asy'ari, M. (2019). *Konsep Menjemput Rezeki dalam al Qur'an (Studi Aplikatif pada Usaha Nasi Goreng Kebuli Pak Manshur)*. 61–115.
- Belluano, P. L. L. (2016). *Optimalisasi Solusi Terbaik Dengan Penerapan Non-Dominated Sorting II Algorithm*. 8(April), 29–36.
- Bolaños, R. I., Echeverry, M. G., & Escobar, J. W. (2015). A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the multiple traveling salesman problem. *Decision Science Letters*, 4(4), 559–568. <https://doi.org/10.5267/j.dsl.2015.5.003>
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Fatmawati, Prihandono, B., & Noviani, E. (2015). Penyelesaian Travelling Salesman Problem Dengan Metode Tabu Search. *Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster)*, 04 no. 1(1), 17–24.
- Fauzi, A. A., & Sulistyono, E. (2022). Traveling Salesman Problem Dalam Menyelesaikan Rute Optimal Pengiriman Air Minum Isi Ulang. *Jurnal Sintak*, 1(1), 31–38. <https://doi.org/>
- 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). ... *Teknologi Informasi Dan Ilmu ...*, 1(9), 849–858. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/268>
- Kusrini, & Istiyanto, J. E. (2007). Penyelesaian travelling salesman problem dengan algoritma cheapest insertion heuristics dan basis data. *Jurnal Informatika*, 8(2), 109–114. <http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/16775>
- Li, J., Chen, J., Xin, B., & Dou, L. (2015). Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGAII and adaptive MOEA/D: A comparison study. *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, May 2017*, 3132–3139. <https://doi.org/10.1109/CEC.2015.7257280>
- Luo, Y., Liu, M., Hao, Z., & Liu, D. (2014). An Improved NSGA-II Algorithm for Multi-objective Traveling Salesman Problem. *TELKOMNIKA Indonesian*

Journal of Electrical Engineering, 12(6), 4413–4418.
<https://doi.org/10.11591/telkomnika.v12i6.5476>

- Made, P., Raditya, 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*, 2(10), 3560–3568. <http://j-ptiik.ub.ac.id>
- Myszkowski, P. B., Laszczyk, M., & Dziadek, K. (2019). Non-dominated sorting tournament genetic algorithm for multi-objective travelling salesman problem. *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, 18, 67–76.
<https://doi.org/10.15439/2019F192>
- Puteri, R. N., Widodo, A. W., & Cholissodin, I. (2017). Optimasi Multiple Travelling Salesman Problem Pada Pendistribusian Air Minum Menggunakan Algoritme Particle Swarm Optimization (Studi Kasus : UD. Tosa Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, 1(9), 842–848.
- Shuai, Y., Yunfeng, S., & Kai, Z. (2019). An effective method for solving multiple travelling salesman problem based on NSGA-II. *Systems Science and Control Engineering*, 7(2), 121–129. <https://doi.org/10.1080/21642583.2019.1674220>
- Vitaloka, A. R. D. (2018). Penerapan Cat Swarm Optimization Algorithm (CSO) dan Particle Swarm Optimization Algorithm (PSO) Dalam Penyelesaian Travelling Salesman Problem (TSP). *Digital Repository Universitas Jember*.