



TUGAS AKHIR - QQXXXXXX

PENGEMBANGAN SISTEM REKOMENDASI RUTE WISATA HALAL DI SURABAYA DENGAN METODE HYBRID GENETIC ALGORITHM

RAHMAT RAMADHAN PERMANA

NRP 5026221154

Dosen Pembimbing

Amalia Utamima, S.Kom., MBA., Ph.D.

NIP 198612132015042001

Program Studi S-1 Sistem Informasi

Departemen Sistem Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025



TUGAS AKHIR - QQXXXXXX

PENGEMBANGAN SISTEM REKOMENDASI RUTE WISATA HALAL DI SURABAYA DENGAN METODE HYBRID GENETIC ALGORITHM

RAHMAT RAMADHAN PERMANA

NRP 5026221154

Dosen Pembimbing

Amalia Utamima, S.Kom., MBA., Ph.D.

NIP 19861213 2015042001

Program Studi S-1 Sistem Informasi

Departemen Sistem Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025



FINAL PROJECT - QQXXXXXX

DEVELOPMENT OF HALAL TOURISM ROUTES RECOMMENDATION SYSTEM IN SURABAYA USING HYBRID GENETIC ALGORITHM

RAHMAT RAMADHAN PERMANA

NRP 5026221154

Advisor

Amalia Utamima, S.Kom., MBA., Ph.D.

NIP 198612132015042001

Study Program S-1 Information System

Department of Information System

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2025

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM REKOMENDASI RUTE WISATA HALAL DI SURABAYA DENGAN METODE HYBRID GENETIC ALGORITHM

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar S.Kom. pada

Program Studi S-1 Sistem Informasi

Departemen Sistem Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh : Rahmat Ramadhan Permana

NRP. 5026221154

Disetujui oleh Tim Penguji Proposal Tugas Akhir :

- | | |
|--|------------|
| 1. Amalia Utamima, S.Kom., MBA., Ph.D. | Pembimbing |
| 2. Edwin Riksakomara, S.Kom., M.T. | Penguji |
| 3. Faizal Mahananto, S.Kom., M.Eng., Ph.D. | Penguji |

SURABAYA

Juli, 2025

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Rahmat Ramadhan Permana / 5026221154
Program studi : Sistem Informasi
Dosen Pembimbing / NIP : Amalia Utamima, S.Kom., MBA., Ph.D. /
198612132015042001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Pengembangan Sistem Rekomendasi Rute Wisata Halal di Surabaya dengan Metode Hybrid Genetic Algorithm” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, _____

Mengetahui
Dosen Pembimbing

Mahasiswa

Amalia Utamima, S.Kom., MBA., Ph.D.
NIP. 198612132015042001

Rahmat Ramadhan Permana
NRP. 5026221154

ABSTRAK

PENGEMBANGAN SISTEM REKOMENDASI RUTE WISATA HALAL DI SURABAYA DENGAN METODE HYBRID GENETIC ALGORITHM

Nama Mahasiswa / NRP : **Rahmat Ramadhan Permana / 5026221154**
Departemen : **Sistem Informasi FTEIC - ITS**
Dosen Pembimbing : **Amalia Utamima, S.Kom., MBA., Ph.D.**

Abstrak

Sektor pariwisata, khususnya wisata halal, memiliki peran penting dalam perekonomian Kota Surabaya. Meskipun memiliki potensi besar, masyarakat dan wisatawan masih memiliki pemahaman dan informasi yang minim terkait wisata halal. Penelitian sebelumnya mengenai sistem rekomendasi rute wisata di Surabaya masih memiliki keterbatasan, seperti cakupan destinasi yang terbatas hanya pada kuliner dan sebaran data yang belum merata. Tugas akhir ini bertujuan untuk mengembangkan sistem rekomendasi rute wisata halal di Surabaya yang mampu meminimalkan jarak tempuh dengan menggunakan metode *Hybrid Genetic Algorithm* (HGA). Metode HGA dilakukan dengan menggabungkan keunggulan *Genetic Algorithm* (GA) dalam pencarian solusi global dan algoritma 2-Opt untuk optimasi lokal, sehingga menghasilkan rute yang lebih optimal. Data yang digunakan dalam pengembangan tugas akhir ini didapatkan dari Dinas Kebudayaan, Kepemudaan dan Olahraga serta Pariwisata (Disbudporapar) Kota Surabaya dan *website* WaveHalal. Sistem ini akan menghasilkan rute yang mencakup berbagai kategori destinasi, mulai dari kuliner makanan berat, makanan ringan, wisata non-kuliner, hingga pusat oleh-oleh atau souvenir, dalam durasi perjalanan maksimal satu hari. Luaran dari penelitian ini adalah sebuah *website* yang tidak hanya menyajikan rute wisata halal yang optimal, tetapi juga dilengkapi dengan fitur tambahan chatbot berbasis *Artificial Intelligence* (AI) untuk membantu menjawab pertanyaan pengguna seputar pariwisata di Surabaya. Diharapkan sistem ini dapat memudahkan wisatawan dalam merencanakan perjalanan mereka, memberikan pengalaman wisata yang lebih baik, serta turut meningkatkan daya tarik pariwisata halal di Kota Surabaya.

Kata kunci: *Sistem Rekomendasi, Hybrid Genetic Algorithm, Rute Wisata Halal, Travelling Salesman Problem.*

ABSTRACT

DEVELOPMENT OF HALAL TOURISM ROUTES RECOMMENDATION SYSTEM IN SURABAYA USING HYBRID GENETIC ALGORITHM

Student Name / NRP : **Rahmat Ramadhan Permana / 5026221154**
Department : **Sistem Informasi FTEIC - ITS**
Advisor : **Amalia Utamima, S.Kom., MBA., Ph.D.**

Abstract

The tourism sector, particularly halal tourism, plays a significant role in the economy of Surabaya City. Despite its great potential, the public and tourists still have minimal understanding and information regarding halal tourism. Previous research on tour route recommendation systems in Surabaya has had limitations, such as a destination scope confined only to culinary tourism and an uneven data distribution. This final project aims to develop a decision support system for determining halal tourism route in Surabaya that can minimize travel distance by using the Hybrid Genetic Algorithm (HGA) method. The HGA method is implemented by combining the strengths of the Genetic Algorithm (GA) for global solution search and the 2-Opt algorithm for local optimization, thereby producing a more optimal route. The data used in the development of this final project is obtained from the Surabaya City Office of Culture, Youth, Sports, and Tourism (Disbudporapar) and the WaveHalal website. The system will generate a route that cover various destination categories, from heavy meal culinary spots, snacks, and non-culinary tourism, to souvenir or gift centers, all within a maximum travel duration of one day. The output of this research is a website that not only provides optimal halal tourism route but is also equipped with an additional Artificial Intelligence (AI)-based chatbot feature to help answer user questions about tourism in Surabaya. It is hoped that this system can make it easier for tourists to plan their trips, provide a better tourism experience, and help increase the appeal of halal tourism in Surabaya City.

Keywords: **Route Recommendation System, Hybrid Genetic Algorithm, Halal Tourism, Travelling Salesman Problem.**

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR KODE	ix
DAFTAR SIMBOL	x
BAB 1 PENDAHULUAN	11
1.1 Latar Belakang.....	11
1.2 Rumusan Masalah.....	12
1.3 Batasan Masalah.....	12
1.4 Tujuan.....	13
1.5 Manfaat.....	13
1.6 Relevansi	13
1.7 Asumsi	14
1.8 Luaran.....	14
BAB 2 TINJAUAN PUSTAKA	15
2.1 Hasil Penelitian Terdahulu	15
2.2 Dasar Teori	19
2.2.1 Sistem Rekomendasi Rute	19
2.2.2 <i>Travelling Salesman Problem</i>	20
2.2.3 <i>Genetic Algorithm</i>	21
2.2.4 Algoritma 2-Opt	22
2.2.5 <i>Hybrid Genetic Algorithm</i>	23
2.2.6 Chatbot.....	23
BAB 3 METODOLOGI	25
3.1 Metode yang Digunakan.....	25
3.1.1 Identifikasi Masalah	27
3.1.2 Studi Literatur.....	27
3.1.3 Pengumpulan Data.....	27
3.1.4 Praproses Data	27

3.1.5	Pengembangan Sistem Rekomendasi Rute.....	28
3.1.6	Pengembangan Website.....	29
3.1.7	Penyusunan Buku Tugas Akhir	29
3.2	Urutan Pengerjaan Tugas Akhir	30
BAB 4	Perancangan	31
4.1	Perancangan Arsitektur <i>Website</i>	31
4.1.1	<i>Frontend</i>	32
4.1.2	<i>Backend</i>	32
4.2	Pengumpulan dan Deskripsi Data.....	34
4.2.1	Matriks Jarak	34
4.3	Perancangan Algoritma Sistem Rekomendasi Rute	36
4.3.1	Perancangan Kromosom.....	36
4.3.2	Perancangan Fungsi <i>Fitness</i>	37
4.3.3	Perancangan Populasi.....	41
4.3.4	Perancangan Operator Seleksi	42
4.3.5	Perancangan Operator <i>Crossover</i>	43
4.3.6	Perancangan Operator Mutasi	46
4.3.7	Perancangan Algoritma 2-Opt.....	47
4.3.8	Perancangan Proses Evolusi	48
4.4	Perancangan Uji Coba Sistem	51
4.4.1	Perancangan Uji Fungsional.....	51
4.4.2	Perancangan Uji Performa Algoritma	52
BAB 5	Implementasi	53
5.1	Pengembangan Algoritma Sistem Rekomendasi Rute	53
5.2	Integrasi Algoritma Sistem Rekomendasi Rute dengan <i>Website</i>	54
BAB 6	Hasil dan Pembahasan	55
DAFTAR PUSTAKA		56
LAMPIRAN		58

DAFTAR GAMBAR

Gambar 1.1 Roadmap Laboratorium RDIB	14
Gambar 3.1 Diagram alir pengembangan tugas akhir	26
Gambar 4.1 Alur Pengguna untuk Menghasilkan Rekomendasi Rute Wisata pada <i>Website</i> ...	31
Gambar 4.2 Tahap pertama dalam <i>crossover</i>	44
Gambar 4.3 Tahap kedua dalam <i>crossover</i>	44
Gambar 4.4 Tahap ketiga dalam <i>crossover</i>	45
Gambar 4.5 Tahap keempat dalam <i>crossover</i>	45
Gambar 4.6 Tahap kelima dalam <i>crossover</i>	46
Gambar 4.7 Diagram alir proses evolusi	48

DAFTAR TABEL

Tabel 1.1 Deskripsi notasi dari urutan rute wisata	13
Tabel 2.1 Literatur 1	16
Tabel 2.2 Literatur 2	16
Tabel 2.3 Literatur 3	16
Tabel 2.4 Literatur 4	17
Tabel 2.5 Literatur 5	17
Tabel 2.6 Literatur 6	18
Tabel 2.7 Literatur 7	18
Tabel 2.8 Literatur 8	19
Tabel 2.9 Literatur 9	19
Tabel 2.10 Deskripsi notasi dari persamaan fungsi tujuan	21
Tabel 3.1 Jadwal penggerjaan tugas akhir.....	30

DAFTAR KODE

Kode 4.1 Contoh <i>Payload Request</i> pada Endpoint /api/generate-routes.....	33
Kode 4.2 Pseudocode inisialisasi populasi HGA	42
Kode 4.3 Pseudocode <i>tournament selection</i>	43
Kode 4.4 Pseudocode <i>crossover</i>	43
Kode 4.5 Pseudocode mutasi.....	47
Kode 4.6 Pseudocode algoritma 2-Opt.....	47
Kode 4.7 Pseudocode proses evolusi.....	50
Kode 4.8 Pseudocode pembuatan generasi baru	51

DAFTAR SIMBOL

BAB 1 PENDAHULUAN

Pada bab ini dijelaskan latar belakang dari tugas akhir ini. Berikutnya dilakukan perumusan masalah berdasarkan latar belakang yang telah dijelaskan. Setelah rumusan masalah ditetapkan, selanjutnya ditetapkan batasan-batasan dan tujuan dalam mengembangkan tugas akhir ini. Selain itu, pada bab ini juga dijelaskan manfaat dan relevansi tugas akhir ini dengan *roadmap* laboratorium Rekayasa Data dan Intelelegensi Bisniss (RDIB) Sistem Informasi ITS.

1.1 Latar Belakang

Sektor pariwisata memegang peranan penting dan strategis dalam pembangunan nasional Indonesia. Sektor ini diposisikan sebagai prioritas nasional ketiga yang tertuang dalam Rencana Pembangunan Jangka Menengah Nasional (RPJMN) 2025–2029 untuk memacu pertumbuhan ekonomi wilayah (Pemerintah Pusat Indonesia, 2025). Sebagai salah satu kota terbesar di Indonesia, Surabaya memiliki potensi pariwisata yang sangat besar. Berdasarkan data dari Dinas Kebudayaan, Kepemudaan dan Olahraga serta Pariwisata (Disbudporapar) Kota Surabaya, sepanjang tahun 2023, terdapat sekitar 17,4 juta wisatawan yang berkunjung ke Kota Surabaya (Pemerintah Kota Surabaya, 2024). Kemudian, sepanjang tahun 2024, pengeluaran rumah tangga yang salah satunya adalah wisata kuliner, menggerakkan 60,24% perekonomian di Surabaya. Hal tersebut menunjukkan bahwa sektor pariwisata juga berperan penting dalam roda ekonomi di Kota Surabaya (Bappedalitbang Kota Surabaya, 2025).

Salah satu kategori dalam sektor pariwisata yang memiliki prospek yang baik dalam aspek ekonomi adalah wisata halal. Inti dari wisata halal adalah penerapan prinsip-prinsip islam dalam semua aspek pariwisata, mulai dari pelayanan yang ramah hingga pemilihan destinasi yang menjunjung nilai-nilai syariat islam (Noviantoro & Zurohman, 2020). Global Muslim Travel Index (GMTI) memproyeksikan bahwa jumlah kedatangan wisatawan Muslim internasional akan mencapai 245 juta pada tahun 2030, dengan total pengeluaran diperkirakan mencapai 230 miliar Dolar AS. Sebagai negara dengan populasi Muslim terbesar di dunia, Indonesia memiliki potensi luar biasa untuk menjadi pusat pariwisata halal global (Pak, 2025). Meskipun demikian, wisata halal di Indonesia masih menghadapi tantangan seperti minimnya pengetahuan masyarakat atau wisatawan terkait dengan destinasi wisata halal serta kurangnya pengembangan infrastruktur dan promosi destinasi wisata halal. Hal tersebut juga merupakan imbas dari minimnya riset yang terkait dengan wisata halal di Indonesia (Noviantoro & Zurohman, 2020).

Saat ini, salah satu upaya yang telah dilakukan untuk meningkatkan potensi wisata halal dan menyediakan informasi untuk wisata halal terutama di Surabaya adalah dengan membuat sistem rekomendasi wisata halal. Terdapat beberapa penelitian dan tugas akhir yang telah dilakukan terkait dengan sistem rekomendasi rute wisata, seperti tugas akhir yang dibuat oleh Arifin (2024), menjelaskan pengembangan sistem rekomendasi rute wisata kuliner halal di Surabaya dengan metode *Genetic Algorithm* (GA). Kemudian, tugas akhir yang dibuat oleh Edwardo (2023) mengembangkan sistem informasi rekomendasi pariwisata di Kota Surabaya menggunakan GA yang kemudian luaran dari model ini ditampilkan melalui sebuah aplikasi berbasis web. Sedangkan, Rakhmawati, et. al. (2024) dalam penelitiannya menggunakan metode linked data untuk memprediksi status halal pada suatu produk.

Upaya yang telah dilakukan tersebut memiliki beberapa kekurangan, salah satunya pada penelitian yang dilakukan oleh Arifin (2024), yakni, cakupan destinasi hanya spesifik pada

usaha kuliner saja. Selain itu, persebaran data destinasi yang digunakan pada penelitian yang dilakukan oleh Edwardo (2023) hanya terpusat di satu kecamatan pada area Surabaya saja sehingga hal tersebut dapat memengaruhi pengalaman pengguna dalam menggunakan layanan rekomendasi rute wisata. Selain tugas akhir yang dibuat oleh Arifin (2023) dan Edwardo (2023), saat ini belum ada lagi sistem web yang bisa merekomendasikan rute wisata kuliner halal yang ada di Surabaya sehingga dengan adanya tugas akhir ini, diharapkan terdapat sebuah sistem web yang bisa membantu masyarakat atau wisatawan yang berwisata di Surabaya untuk menentukan rute restoran halal sekaligus berwisata dengan tambahan fitur bantuan chatbot berbasis *Artificial Intelligence* (AI).

Dari permasalahan yang telah dijelaskan, diperlukan sistem rekomendasi yang merekomendasikan rute wisata halal dengan berbagai kategori destinasi dari wisata kuliner halal, wisata halal non-kuliner, hingga lokasi penjualan oleh-oleh, camilan, atau jajanan yang datanya yang merata di seluruh wilayah Kota Surabaya. Sehingga, dalam tugas akhir ini dibuat sebuah sistem rekomendasi rute wisata halal yang tersebar merata di Kota Surabaya dengan pendekatan *Hybrid Genetic Algorithm* (HGA) dan fitur chatbot berbasis AI untuk membantu pengguna menanyakan hal seputar wisata kuliner di Surabaya. Dengan demikian, pengguna diharapkan bisa mendapatkan pengalaman yang baik dari rekomendasi wisata halal dalam jarak dan waktu tempuh yang mendekati optimal.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan pada sub-bab sebelumnya, maka rumusan masalah yang diselesaikan pada tugas akhir ini adalah sebagai berikut:

- a. Bagaimana cara mengembangkan sistem rekomendasi rute yang dapat meminimalkan jarak tempuh rute destinasi wisata halal di Surabaya dengan menggunakan *Hybrid Genetic Algorithm*?
- b. Bagaimana cara mengimplementasikan sistem rekomendasi rute wisata halal di Surabaya pada sebuah *website*?
- c. Bagaimana mengintegrasikan chatbot berbasis AI sebagai fitur tambahan ke dalam *website* rekomendasi rute wisata halal?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan yang telah dipaparkan sebelumnya, maka batasan permasalahan dalam tugas akhir ini adalah:

- a. Pengembangan sistem rekomendasi rute hanya pada wilayah Kota Surabaya.
- b. Sistem rekomendasi rute yang dikembangkan dalam tugas akhir ini adalah sistem rekomendasi rute lokasi wisata halal.
- c. Dalam satu rute mencakup minimal tiga destinasi wisata kuliner dengan menu makanan berat dan halal, tiga destinasi wisata non-kuliner, dua lokasi penjualan makanan ringan atau cemilan, dan tepat satu lokasi penjualan oleh-oleh atau souvenir.
- d. Rute yang dihasilkan dari sistem rekomendasi rute ini berurut dengan notasi urutan $K_1, C_1, W_1, K_2, W_2, C_2, K_3, O$. Penjelasan setiap notasi dalam urutan rute tersebut dijelaskan pada tabel 1.1.

Tabel 1.1 Deskripsi notasi dari urutan rute wisata

Notasi	Deskripsi
n	Bilangan bulat yang merepresentasikan urutan dari tiap kategori lokasi.
K_n	Destinasi wisata kuliner dengan menu makanan berat dan halal.
C_n	Lokasi penjualan makanan ringan atau cemilan.
W_n	Destinasi wisata non-kuliner.
O	Lokasi penjualan oleh-oleh atau souvenir.

- e. Tugas akhir berfokus pada pengembangan sistem rekomendasi rute dengan batasan jumlah waktu tempuh perjalanan dalam rute maksimal selama lima jam.
- f. Tugas akhir berfokus pada pengembangan sistem rekomendasi rute dengan batasan jumlah jarak tempuh perjalanan dalam rute maksimal sepanjang 20 km.
- g. Data lokasi wisata yang digunakan dalam sistem rekomendasi rute bersifat statis dan didapatkan dari Disbudporapar Kota Surabaya dan website WaveHalal.
- h. Chatbot berbasis AI yang digunakan pada tugas akhir ini hanya sebagai fitur tambahan, tidak berpengaruh pada model sistem rekomendasi, dan digunakan API model AI generatif yang sudah siap digunakan.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, tujuan yang akan dicapai dalam tugas akhir ini adalah:

- a. Mengembangkan sistem rekomendasi rute yang dapat meminimalkan jarak tempuh rute lokasi wisata halal di Surabaya dengan menggunakan *Hybrid Genetic Algorithm*.
- b. Mengimplementasikan algoritma rekomendasi rute untuk merekomendasikan rute destinasi wisata halal di Surabaya ke dalam sebuah *website*.
- c. Mengintegrasikan chatbot berbasis AI sebagai fitur tambahan ke dalam website untuk membantu pengguna menjawab pertanyaan seputar wisata di Surabaya.

1.5 Manfaat

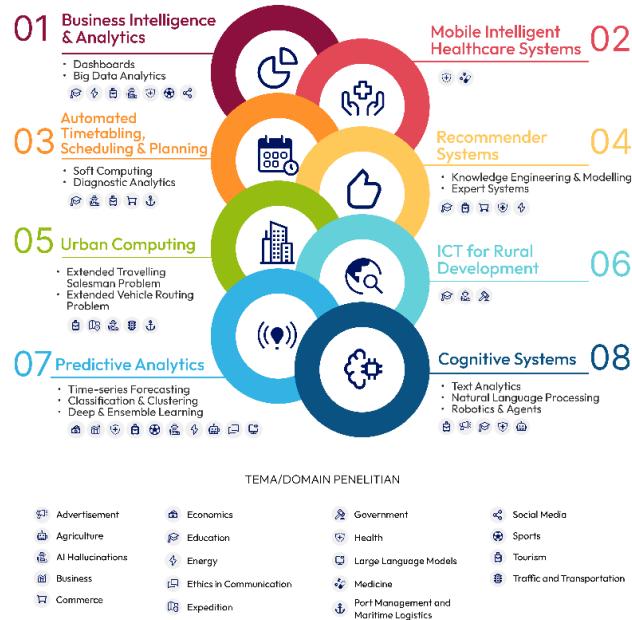
Berikut merupakan manfaat yang diharapkan dapat diperoleh dari tugas akhir ini:

- a. Menambah pengetahuan, pengalaman, dan referensi bagi penulis dalam pengembangan sistem rekomendasi rute dengan *Hybrid Genetic Algorithm* dan mengimplementasikannya dalam sebuah website.
- b. Memudahkan masyarakat terutama wisatawan untuk menjelajahi wisata halal di Surabaya.
- c. Diharapkan dapat meningkatkan daya tarik wisata di Surabaya.

1.6 Relevansi

Tugas akhir ini memiliki relevansi dengan *roadmap* laboratorium Rekayasa Data dan Intelektual Bisnis (RDIB) dengan topik *Urban Computing* yang menyelesaikan *Travelling Salesman Problem* (TSP) seperti yang ditunjukkan oleh gambar 1.1

ROADMAP LABORATORIUM
**REKAYASA DATA &
INTELEGENSI BISNIS**
TAHUN 2025 - 2035



Gambar 1.1 Roadmap Laboratorium RDIB

Pembahasan lebih lanjut terkait TSP dan bagaimana keterkaitannya dengan tugas akhir ini dijelaskan lebih lanjut pada bab 2.

1.7 Asumsi

1. Wisata halal adalah destinasi wisata yang lokasinya bukan tempat ibadah, wilayah adat, atau kebudayaan nonmuslim.
2. Dalam satu rute wisata, pengguna perlu makan makanan berat sebanyak tiga kali, membeli makanan ringan sebagai camilan ketika berwisata, mengunjungi dua destinasi wisata nonkuliner, dan membeli oleh-oleh di akhir rute wisata.

1.8 Luaran

Luaran yang akan dihasilkan dari tugas akhir ini yakni:

- a. Sistem rekomendasi rute wisata halal di Kota Surabaya berbasis *Hybrid Genetic Algorithm*.
- b. Website rekomendasi rute wisata halal di Kota Surabaya.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini dilakukan tinjauan pustaka dengan melakukan studi literatur yang terkait dengan topik yang diangkat pada tugas akhir ini. Tinjauan pustaka dimulai dengan mencari penelitian-penelitian terdahulu yang berkaitan dengan topik tugas akhir ini. Dari literatur-literatur tersebut, berikutnya disusun dasar teori yang dapat mendukung pengembangan tugas akhir ini.

2.1 Hasil Penelitian Terdahulu

Pada subbab ini akan dipaparkan hasil dari penelitian-penelitian terdahulu yang terkait dengan topik pada tugas akhir ini sehingga bisa digunakan sebagai referensi, dasar, dan pedoman dalam pengembangan tugas akhir ini. Ditemukan beberapa penelitian dan tugas akhir yang relevan dengan topik pengembangan sistem rekomendasi rute wisata halal di Surabaya. Tugas akhir yang dibuat oleh Edwardo (2023) mengembangkan sebuah sistem informasi rekomendasi pariwisata di Kota Surabaya. Studi tersebut menyelesaikan *Orienteering Problems* (OP) dengan menggunakan GA untuk menentukan rute destinasi wisata yang kemudian diimplementasikan ke dalam sebuah website. Berikutnya, tugas akhir yang dibuat oleh Arifin (2024) juga mengembangkan sistem rekomendasi rute wisata, tetapi berfokus pada wisata kuliner halal Surabaya. Kedua studi tersebut memberi gambaran yang jelas tentang bagaimana sebuah sistem rekomendasi rute dikembangkan dengan menggunakan GA sehingga terkait langsung dengan topik yang diangkat oleh penulis dalam tugas akhir ini.

Berikutnya, penelitian yang dilakukan oleh Anshari & Baizal (2023) juga menggunakan GA untuk menghasilkan rute optimal berdasarkan jumlah hari perjalanan wisata dengan *Multi-Attribute Utility Theory* (MAUT) dengan mempertimbangkan kriteria-kriteria dari pengguna seperti *rating*, biaya, dan waktu perjalanan. Penelitian tersebut relevan dengan tugas akhir ini sebagai referensi untuk mengembangkan sistem rekomendasi rute dengan mempertimbangkan variabel-variabel yang terkait untuk menentukan rute optimal.

Terdapat pula penelitian untuk memecahkan masalah serupa seperti yang akan diselesaikan pada tugas akhir ini, yakni, *Travelling Salesman Problem* (TSP) yang dilakukan oleh Ahmed, et. al. (2024) yang mengusulkan metode HGA dengan mengombinasikan GA untuk mencari solusi global dan algoritma 2-opt untuk mencari solusi lokal untuk menghasilkan solusi yang lebih optimal untuk TSP daripada menggunakan GA biasa. Kemudian, Soares, et.al (2024) juga mengusulkan pendekatan HGA untuk meningkatkan pencarian solusi optimal dalam TSP-rd. Penelitian yang dilakukan oleh Mahmoudinazlou & Kwon (2024) juga mengusulkan pendekatan HGA yang mengombinasikan GA dengan *dynamic programming* untuk menyelesaikan min-max mTSP dengan lebih optimal. Suanpang, et. al. (2022) juga mengembangkan HGA yang mengombinasikan GA dengan *simulated annealing algorithm* untuk mengoptimalkan penjadwalan layanan pariwisata. Ketiga penelitian tersebut mengusulkan pendekatan HGA sehingga dapat menjadi referensi bagi penulis dalam mengembangkan HGA dalam tugas akhir ini dengan memahami lebih dalam tentang TSP dan varian dari GA.

Kemudian, terdapat penelitian yang dilakukan oleh Rakhmawati, et. al. (2021) yang mengembangkan aplikasi berbasis web dan aplikasi android untuk mengklasifikasikan status halal suatu produk. Hasil dari penelitian tersebut bisa digunakan oleh penulis sebagai salah satu alat untuk membuat dataset restoran halal yang ada di Surabaya. Lalu, penelitian yang

dilakukan oleh Kurniawan (2024) menjelaskan tentang implementasi aplikasi chatbot berbasis web dengan menggunakan kerangka kerja Next.js. Penelitian tersebut relevan dengan pengembangan chatbot berbasis web dalam tugas akhir ini.

Penjelasan lebih lanjut tentang keterkaitan tugas akhir penulis dengan beberapa tugas akhir dan penelitian sebelumnya ditunjukkan pada tabel 2.1.

Tabel 2.1 Literatur 1

Judul	Pengembangan Sistem Informasi Rekomendasi Pariwisata di Kota Surabaya Menggunakan Algoritme Genetika
Penulis	Djiehaddan Edwardo M.
Tahun	2023
Isi	Ini merupakan tugas akhir yang menyelesaikan <i>Orienteering Problems</i> (OP) yang merupakan masalah gabungan dari <i>Travelling Salesman Problem</i> (TSP) dan <i>Knapsack Problems</i> (KP) dengan menggunakan GA. OP yang diselesaikan pada tugas akhir ini adalah penentuan rute untuk wisatawan agar bisa mengunjungi tempat sebanyak mungkin dalam waktu atau jarak tempuh semimum mungkin. Model dalam tugas akhir ini dikembangkan berdasarkan 51 data lokasi; 50 data destinasi pariwisata dan ratingnya, serta data lokasi pengguna.
Keterkaitan dengan Tugas Akhir	Tugas akhir ini menggunakan GA untuk menyelesaikan OP yang menentukan rute rekomendasi pariwisata di Kota Surabaya.

Tabel 2.2 Literatur 2

Judul	Pengembangan Sistem Rekomendasi Rute Wisata Kuliner Halal Surabaya dengan Metode <i>Genetic Algorithm</i>
Penulis	Ramadhana Rizqy Arifin
Tahun	2024
Isi	Ini merupakan tugas akhir yang mengembangkan sistem rekomendasi rute berbasis GA dengan tujuan optimalisasi pemilihan destinasi kuliner halal. Dalam tugas akhir ini digunakan faktor-faktor seperti lokasi, preferensi, dan batasan waktu untuk menentukan rute. Luaran dari tugas akhir ini adalah sistem rekomendasi rute yang diimplementasikan ke dalam sistem berbasis REST API.
Keterkaitan dengan Tugas Akhir	Tugas akhir ini menggunakan GA untuk menentukan rute dan mengimplementasikannya ke dalam sistem REST API. Cara tersebut dapat dijadikan referensi dalam pelaksanaan tugas akhir ini untuk mengintegrasikan sistem rekomendasi rute restoran halal dengan website.

Tabel 2.3 Literatur 3

Judul	<i>N-Days Tourist Route Recommender System in Yogyakarta Using Genetic Algorithm</i>
Penulis	Muhammad Ridha Anshari dan Z. K. A Baizal
Tahun	2023

Isi Penelitian	Penelitian ini mengusulkan penggunaan GA untuk menghasilkan rencana perjalanan dengan waktu kunjungan tertentu (rute tur n-hari) yang optimal secara otomatis. Pada penelitian ini GA dikombinasikan dengan konsep <i>Multi-Attribute Utility Theory</i> (MAUT) yang digunakan untuk mengakomodasi kebutuhan pengguna berdasarkan kriteria-kriteria seperti <i>rating</i> , biaya, dan waktu. Penelitian ini menunjukkan bahwa GA optimal dalam hal waktu eksekusi dan jumlah objek wisata yang dikunjungi dalam kunjungan n-hari.
Keterkaitan dengan Tugas Akhir	Penelitian menjelaskan penggunaan GA dalam pengembangan sistem rekomendasi rute dengan mempertimbangkan variabel-variabel yang terkait seperti <i>rating</i> destinasi, biaya perjalanan, dan waktu tempuh untuk menentukan rute optimal.

Tabel 2.4 Literatur 4

Judul	Optimization of the Travelling Salesman Problem Using a New Hybrid Genetic Algorithm
Penulis	Zakir Hussain Ahmed, Furat Fahad Altukhaim, Abdul Khader Jilani Saudagar, dan Shakir Khan
Tahun	2024
Isi Penelitian	Penelitian ini mengusulkan HGA baru untuk menyelesaikan TSP. Algoritma dalam penelitian ini menggabungkan beberapa metode, yaitu <i>comprehensive sequential constructive</i> crossover, mutasi adaptif, pencarian 2-opt, dan algoritma pencarian lokal baru serta metode penggantian untuk meningkatkan kinerja. Hasil eksperimen pada serangkaian data uji standar (TSPLIB) menunjukkan bahwa HGA yang diusulkan secara signifikan lebih unggul daripada <i>Simple Genetic Algorithm</i> (SGA) dan juga lebih baik dari algoritma canggih lainnya seperti <i>Ant Colony System Algorithm</i> (ACSA), baik untuk kasus TSP simetris maupun asimetris.
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi utama dalam pengembangan HGA dengan mengombinasikan GA dan algoritma 2-opt dalam menyelesaikan TSP.

Tabel 2.5 Literatur 5

Judul	An Efficient Hybrid Genetic Algorithm for The Travelling Salesman Problem with Release Date
Penulis	Gabriel Soares, Teobaldo Bulhões, dan Bruno Bruck
Tahun	2024
Isi Penelitian	Penelitian ini menyelesaikan TSP yang setiap pelanggannya terasosiasi dengan tanggal rilis yang didefinisikan sebagai waktu saat produk yang diinginkan tersedia di depot. Untuk menyelesaikan masalah tersebut, penelitian ini mengusulkan HGA yang menggabungkan mekanisme yang lebih canggih untuk mengevaluasi individu dan memastikan keragaman populasi. Mekanisme yang digunakan adalah dengan menambahkan fase pendidikan untuk menyempurnakan setiap solusi (individu) baru yang dihasilkan dari fase <i>crossover</i> . Digunakan prosedur pencarian lokal dalam

	fase pendidikan untuk memperbaiki solusi secara iteratif. Kemudian, digunakan juga algoritma <i>split</i> yang merupakan algoritma <i>dynamic programming</i> yang dikembangkan khusus untuk menyelesaikan masalah pada penelitian ini untuk mengevaluasi <i>fitness</i> setiap individu dalam GA.
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi dalam pengembangan HGA sehingga dapat ditentukan bagaimana peran GA dan algoritma lainnya dalam menyelesaikan TSP dengan HGA.

Tabel 2.6 Literatur 6

Judul	<i>A Hybrid Genetic Algorithm for the Min-Max Multiple Travelling Salesman Problem</i>
Penulis	Sasan Mahmoudinazlou dan Changhyun Kwon
Tahun	2024
Isi Penelitian	Penelitian ini mengusulkan metode <i>Hybrid GA</i> untuk menyelesaikan <i>Multiple Travelling Salesman Problem</i> (mTSP) dengan tujuan untuk meminimalkan panjang perjalanan terpanjang. Dalam penelitian ini GA dikombinasikan dengan <i>dynamic programming</i> untuk menghasilkan solusi optimal. Hasil penelitian ini secara rata-rata mengungguli semua algoritma yang ada, dengan ambang batas waktu pemotongan yang sama, ketika diuji terhadap beberapa set tolok ukur yang ditemukan dalam literatur.
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi dalam pengembangan HGA sehingga dapat ditentukan bagaimana peran GA dan algoritma lainnya dalam menyelesaikan TSP dengan HGA.

Tabel 2.7 Literatur 7

Judul	<i>Tourism Service Scheduling in Smart City Based on Hybrid Genetic Algorithm Simulated Annealing Algorithm</i>
Penulis	Panee Suanpong, Pitchaya Jamjuntr, Kittisak Jermisittiparsert, dan Phuripoj Kaewyong
Tahun	2022
Isi Penelitian	Penelitian ini mengembangkan <i>hybrid GA</i> untuk menyelesaikan masalah penjadwalan yang kompleks pada layanan pariwisata di Thailand. Algoritma ini diciptakan dengan mengombinasikan GA, algoritma <i>simulated annealing</i> , dan metode <i>gradient search</i> . Inovasi utama dalam algoritma ini adalah penggunaan operator berbasis pengetahuan untuk memanfaatkan waktu henti mesin serta strategi pencarian lokal berbasis <i>simulated annealing</i> untuk menyempurnakan solusi. Berdasarkan hasil perbandingan dengan algoritma genetika tradisional menggunakan studi kasus dari Departemen Pariwisata Thailand, algoritma <i>hybrid</i> yang diusulkan terbukti menunjukkan kinerja yang lebih baik dan lebih efektif dalam menghasilkan jadwal layanan pariwisata yang memuaskan.
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi dalam pengembangan HGA sehingga dapat ditentukan bagaimana peran GA dan algoritma lainnya dalam menyelesaikan TSP dengan HGA.

Tabel 2.8 Literatur 8

Judul	<i>Linked Open Data for Halal Food Products</i>
Penulis	Nur Aini Rakhmawati, Jauhar Fatawi, Ahmad Choirun Najib, dan Azmi Adi Firmansyah
Tahun	2021
Isi Penelitian	Penelitian ini mengatasi masalah keterbatasan informasi produk makanan halal di Indonesia dengan mengusulkan sistem bernama LODHalal yang berbasis <i>Linked Open Data</i> . Sistem ini mengintegrasikan data dari berbagai sumber, termasuk LPPOM MUI, E-Number, dan Open Food Facts, lalu mengubahnya menjadi format RDF dan menghubungkannya dengan basis data besar lainnya seperti DBpedia dan PubChem. Sebagai hasil akhir, penelitian ini mengembangkan aplikasi web dan Android yang dapat memberikan rekomendasi kepada pengguna mengenai status kehalalan dan kandungan gizi suatu produk makanan secara lebih komprehensif.
Keterkaitan dengan Tugas Akhir	Hasil dari penelitian ini dapat digunakan sebagai salah satu alat untuk mengumpulkan data pada penelitian ini.

Tabel 2.9 Literatur 9

Judul	Pengembangan Desain Antarmuka Pengguna dan Implementasi Front End Aplikasi Chatbot Tanya.In Menggunakan Pendekatan User-Centered Design
Penulis	Mochamad Revanza Kurniawan
Tahun	2024
Isi Penelitian	Penelitian ini mengembangkan chatbot dan mengimplementasikannya ke dalam sebuah website yang dikembangkan dengan kerangka kerja Next.js. Antarmuka website dikembangkan dengan metode <i>User-Centered Design</i> (UCD).
Keterkaitan dengan Tugas Akhir	Penelitian ini dapat dijadikan referensi dalam integrasi chatbot berbasis AI dengan website yang dikembangkan dalam tugas akhir ini yang juga dikembangkan menggunakan kerangka kerja Next.js

Penelitian-penelitian sebelumnya yang telah dijelaskan tersebut berikutnya dijadikan landasan sebagai dasar teori pada tugas akhir ini.

2.2 Dasar Teori

Subbab ini menjelaskan teori-teori yang dapat mendukung pengembangan tugas akhir ini berdasarkan penelitian-penelitian sebelumnya yang berkaitan dengan topik tugas akhir ini. Beberapa teori yang digunakan untuk mendukung tugas akhir ini yakni tentang sistem rekomendasi rute, *Travelling Salesman Problem*, *Genetic Algorithm*, algoritma 2-Opt, *Hybrid Genetic Algorithm*, dan *chatbot*.

2.2.1 Sistem Rekomendasi Rute

Sistem rekomendasi rute bertujuan untuk menyediakan rute terbaik bagi pengguna untuk mencapai lokasi-lokasi yang ditujunya. Rute yang optimal dapat ditentukan oleh sistem rekomendasi berdasarkan kalkulasi yang dilakukan oleh model pemilihan rute. Model

pemilihan rute mencoba untuk mencari jalur terbaik dari berbagai pilihan jalur yang akan dilewati oleh pengguna dari lokasi asal menuju lokasi tujuannya (Nasution, et. al., 2021).

Sistem rekomendasi rute merepresentasikan lalu lintas ke dalam bentuk graf yang memiliki *nodes*, *edges*, dan bobot. Dengan demikian, penentuan rute optimal berdasarkan kriteria atau preferensi dari pengguna seperti *rating* lokasi dan jarak tempuh dapat diimplementasikan pada sistem rekomendasi rute melalui representasi bobot pada graf di sistem rekomendasi rute (Nasution, et. al., 2021).

Sistem rekomendasi rute dapat membantu penggunanya untuk melakukan perjalanan ke berbagai tempat dengan batasan seperti waktu dan jarak dengan rute yang optimal. Oleh karena itu, sistem rekomendasi rute cocok digunakan untuk aktivitas berwisata karena dapat membantu wisatawan untuk menemukan rute optimal untuk perjalanan mereka. Terdapat beberapa algoritma yang dapat digunakan dalam pengembangan sebuah sistem rekomendasi rute, seperti Tabu Search Algorithm, Simulated Annealing Algorithm, dan Genetic Algorithm (Anshari & Baizal, 2023).

2.2.2 Travelling Salesman Problem

Travelling Salesman Problem (TSP) adalah sebuah jenis masalah di mana sebuah objek (*salesman*) perlu mengunjungi berbagai lokasi dengan jarak optimal (terpendek) dan masing-masing lokasi dikunjungi tepat satu kali. TSP termasuk dalam masalah optimasi kombinatorial dan dikategorikan sebagai masalah NP-hard, yang berarti bahwa solusi pastinya tidak praktis untuk dikomputasi ketika berhadapan dengan dataset yang besar. Oleh karena itu, diperlukan algoritma metaheuristik untuk menyelesaikan TSP dengan tujuan mendapatkan solusi mendekati optimal dalam jangka waktu yang wajar (Mahmoudinazlou & Kwon, 2024).

TSP dapat didefinisikan sebagai sebuah masalah di mana terdapat sebuah jaringan yang terdiri dari n kota (simpul) dengan kantor pusat berada di simpul pertama. Untuk setiap pasang kota (i, j) , tersedia sebuah matriks jarak atau bisa juga berupa waktu atau biaya perjalanan, $D = [d_{ij}]$. Tujuan dari TSP adalah untuk menemukan sebuah siklus Hamiltonian (rute yang mengunjungi setiap kota tepat satu kali dan kembali ke kota awal) dengan total jarak, biaya, atau waktu yang minimum. Terdapat dua jenis utama dari TSP, yakni, TSP simetris yang terjadi jika jarak antara dua kota sama, tidak peduli arah perjalanannya ($d_{ij} = d_{ji}$) dan TSP asimetris yang terjadi jika jarak antara dua kota berbeda tergantung pada arah perjalanannya ($d_{ij} \neq d_{ji}$) (Ahmed, et. al., 2024). Model matematis untuk fungsi tujuan dari TSP ditunjukkan oleh persamaan (2.1).

$$\text{Min } Z = \sum_{i \in V} \sum_{j \in V, i \neq j} d_{ij} x_{ij} \quad (2.1)$$

Adapun himpunan yang digunakan dalam model TSP ditunjukkan oleh persamaan (2.2) dan (2.3).

$$N = \{1, 2, \dots, n\} \quad (2.2)$$

$$V = N \cup \{0\} \quad (2.3)$$

Setiap notasi yang digunakan dalam persamaan (2.1), (2.2), dan (2.3) dijelaskan pada tabel 2.10 (Dhouib, 2021).

Tabel 2.10 Deskripsi notasi dari persamaan fungsi tujuan

Notasi	Deskripsi
$\text{Min } Z$	Total jarak yang tempuh rute yang perlu diminimalkan
d_{ij}	Jarak tempuh dari simpul i ke simpul j , untuk setiap $i, j \in V, i \neq j$.
x_{ij}	Variabel biner yang bernilai 1 jika rute langsung dari simpul i ke simpul j dipilih sebagai bagian dari rute, dan 0 jika tidak, untuk $i, j \in V, i \neq j$.
N	Himpunan n lokasi potensial yang dapat dikunjungi.
V	Himpunan semua simpul (titik) dalam jaringan, di mana simpul 0 merepresentasikan lokasi pengguna.

Terdapat beberapa metode yang dapat digunakan untuk menyelesaikan TSP, seperti menggunakan algoritma *Tabu Search*, *Simulated Annealing*, *Ant Colony*, *Artificial Bee Colony*, *Genetic Algorithm*, dan lainnya. Masing-masing metode tersebut menawarkan mekanisme pencarian yang berbeda-beda untuk menjelajahi ruang solusi yang sangat besar secara efisien. Tujuannya adalah menemukan solusi terbaik atau mendekati optimal dalam waktu yang wajar, terutama ketika metode eksak tidak lagi praktis untuk digunakan pada masalah berskala besar (Anshari & Baizal, 2023).

2.2.3 Genetic Algorithm

Genetic Algorithm (GA) adalah algoritma yang mengikuti konsep seleksi alam dan evolusi dalam ilmu biologi. GA bekerja dengan cara menentukan populasi dari individu, di mana setiap individu merepresentasikan solusi yang berbeda-beda dari masalah yang diatasi, dan kemudian akan dilakukan proses untuk menemukan nilai terbaik yang diukur berdasarkan nilai *fitness* tiap individu yang merepresentasikan kualitas masing-masing individu. GA dapat digunakan untuk mengatasi masalah pencarian nilai optimal (Anshari & Baizal, 2023).

Terdapat beberapa istilah yang digunakan dalam GA, yakni, kromosom, *fitness function*, dan operator-operator GA. Dalam konteks pengembangan sistem rekomendasi rute, kromosom merepresentasikan rute-rute yang potensial untuk dilewati. Kemudian, *fitness function* merupakan fungsi untuk membandingkan jarak pada setiap rute yang dihasilkan dengan tujuan untuk mengoptimalkan jarak rute. Lalu, terdapat juga operator-operator GA yang bertujuan untuk melakukan proses pencarian solusi terbaik. Operator-operator tersebut terdiri dari seleksi, *crossover*, dan mutasi (Nozari, et. al., 2025). GA memanfaatkan keacakan melalui operator-operator tersebut yang mendorong keberagaman dan kemampuan beradaptasi untuk menentukan solusi optimal dan menghindari solusi lokal optimum. Dengan demikian, GA unggul dalam lingkungan yang lebih kompleks dan dinamis. GA juga ideal untuk menyelesaikan masalah dengan lanskap solusi non-linier, *discontinuous*, dan sulit untuk dipahami, karena GA bisa menjelajahi ruang solusi yang lebih besar dan berevolusi seiring berjalannya waktu sehingga dapat menghasilkan model yang lebih adaptif terhadap berbagai skenario (Wahab, et. al., 2024).

Secara umum, proses untuk menentukan solusi optimal dalam GA dijelaskan dengan langkah-langkah sebagai berikut (Anshari & Baizal, 2023).

1. Inisialisasi Populasi

Inisialisasi populasi merupakan proses untuk menghasilkan sejumlah kromosom berdasarkan jumlah populasi yang ditentukan (Wahab, 2024). Langkah awal yang dilakukan dalam GA adalah menghasilkan populasi. Dalam konteks pengembangan sistem rekomendasi rute, populasi merepresentasikan koordinat-koordinat lokasi yang digunakan dalam sistem rekomendasi rute.

2. Seleksi

Seleksi adalah proses untuk memilih kromosom untuk setiap generasinya berdasarkan *fitness function*. Metode seleksi yang efektif dapat meningkatkan performa GA untuk mencari solusi optimal dengan lebih cepat. Selain itu, terdapat juga strategi elitism yang dapat digunakan untuk mempertahankan kromosom dengan nilai *fitness* terbaik pada tiap generasinya (Suanpang, et. al., 2022). Terdapat berbagai metode yang dapat digunakan untuk menghitung nilai *fitness*, seperti Sum of the Squared Error, COSEC, Silhouette Coefficient, dan Davis-Bouldin Index (Dhana, 2024).

3. Crossover

Crossover merupakan langkah untuk menghasilkan sebuah kromosom dari sepasang kromosom yang disebut sebagai parent. Crossover dilakukan dengan saling menukar gen dari parent sehingga dapat menghasilkan keturunan (kromosom) baru.

4. Mutasi

Pada tahap mutasi dilakukan perubahan satu atau lebih gen secara acak dengan probabilitas tertentu pada kromosom. Pada tahap ini, terdapat berbagai pendekatan untuk mengubah kromosom, seperti *perturbation*, *insertion*, *movement*, *division and absorption*, dan *deletion* (Dhana, 2024).

Lebih lanjut lagi, metode GA yang bersifat paralel memungkinkan eksplorasi secara simultan pada berbagai kemungkinan solusi yang dapat meningkatkan efisiensi komputasi. Secara keseluruhan, GA merupakan pilihan yang sangat baik untuk perencanaan rute dalam keadaan yang rumit karena kemampuannya untuk melakukan eksplorasi solusi mendekati optimal dengan kombinasi yang unik dan kemampuan beradaptasi dalam lingkungan yang *noise*.

2.2.4 Algoritma 2-Opt

Algoritma 2-opt adalah salah satu metode pencarian lokal (*local search*) klasik dan sederhana, tetapi terbukti efektif untuk menyelesaikan masalah optimasi seperti TSP dan *Vehicle Routing Problem* (VRP). Algoritma ini tidak bertujuan untuk membangun solusi dari awal, melainkan sebagai heuristik perbaikan yang bekerja secara iteratif untuk menyempurnakan sebuah hasil atau solusi yang sudah ada (Ahmed, et. al., 2024)

Algoritma 2-opt dapat digunakan dalam HGA untuk melakukan pencarian lokal. Dalam konteks penyelesaian TSP, ide utama dari algoritma ini adalah dengan menghapus dua sisi (*edge*) dari sebuah rute, yang akan membaginya menjadi dua jalur terpisah, kemudian menyambungkan kembali kedua jalur tersebut dengan cara yang berbeda, misalnya dengan membalik arah rutennya untuk membentuk sebuah rute baru yang valid. Mekanisme kerja 2-opt dimulai dengan sebuah rute yang sudah terbentuk. Algoritma kemudian memilih dua sisi yang tidak bersebelahan (*non-adjacent*) dalam rute tersebut. Sebagai contoh sederhana, misalnya terdapat sebuah rute A-B-C-D. Lalu, algoritma akan memilih dua sisi (*edge*), yakni, A-B dan C-D untuk diputus. Kemudian, rute yang masih saling terhubung, yakni, B-C dibalik menjadi

C-B. Berikutnya, seluruh rute dihubungkan kembali sehingga terbentuk rute baru, yakni, A-C-B-D (Ahmed, et. al., 2024).

2.2.5 Hybrid Genetic Algorithm

Konsep dasar dari *Hybrid Genetic Algorithm* (HGA) adalah menggabungkan mekanisme pencarian global dari GA dengan kemampuan pencarian lokal yang lebih terfokus dari algoritma lain. Ide utamanya adalah memanfaatkan GA untuk menjelajahi ruang solusi secara luas dan mengidentifikasi daerah-daerah yang menjanjikan, lalu menggunakan algoritma pencarian lokal untuk mengeksplorasi daerah-daerah tersebut secara intensif guna menemukan solusi optimal lokal atau bahkan global (Ahmed, et. al., 2024).

Peran utama dari pencarian lokal dalam HGA adalah untuk memperbaiki solusi-solusi yang dihasilkan oleh operator GA (seleksi, *crossover*, dan mutasi). Setelah individu-individu baru (keturunan) terbentuk, algoritma pencarian lokal diterapkan pada sebagian atau seluruh individu tersebut untuk meningkatkan kualitasnya sebelum dimasukkan kembali ke dalam populasi. Proses ini sering disebut sebagai fase edukasi atau pematangan individu. Dengan melakukan eksplorasi lokal, HGA dapat mencapai konvergensi ke solusi berkualitas tinggi dengan lebih cepat dan menghindari terjebak dalam lokal optimum yang kurang baik, yang merupakan salah satu kelemahan GA murni (Mahmoudinazlou & Kwon, 2024).

Terdapat beberapa algoritma yang dapat diintegrasikan ke dalam HGA untuk menyelesaikan TSP, seperti *dynamic programming* (Soares, et. al., 2024), algoritma 2-opt (Mahmoudinazlou & Kwon, 2024) dan *Simulated Annealing* (SA) (Suanpang, et. al., 2024). Integrasi tersebut tidak hanya bersifat komplementer tetapi juga bisa bersifat sinergis. GA menyediakan kerangka kerja untuk pencarian global dan pemeliharaan keragaman populasi, sementara pencarian lokal menyempurnakan solusi dan mempercepat konvergensi menuju solusi optimal.

2.2.6 Chatbot

Chatbot adalah program komputer yang menggunakan model AI untuk dapat menghasilkan berbagai aktivitas, seperti berkomunikasi dengan pengguna saat diminta (Akpan, et. al., 2025). Chatbot dapat diimplementasikan di website, aplikasi mobile, atau media lainnya yang menyediakan informasi kepada pengguna melalui teks, gambar, video, audio, tautan, atau lainnya. Chatbot dapat melakukan berbagai tugas seperti menjawab pertanyaan yang sering diajukan atau pertanyaan seputar layanan, serta mengumpulkan informasi pelanggan. Fungsionalitas setiap chatbot berbeda-beda, bergantung pada kebutuhan bisnis, tempat penerapannya, dan dengan siapa chatbot tersebut berkomunikasi, apakah dengan klien, mitra, atau karyawan. Chatbot AI berbasis web dapat menjadi titik kontak pertama bagi pelanggan dan menjawab FAQ (Stoilova, 2021).

Artificial Intelligence (AI) bukanlah teknologi disruptif yang baru, tetapi evolusi terkininya yang didorong oleh transformasi teknologi, analisis big data, dan komputasi kuantum, telah menghasilkan *Conversational and Generative AI* (CGAI/GenAI) dan chatbot yang mirip seperti sedang berkomunikasi dengan manusia. Chatbot kini dapat menghasilkan respons yang lebih dinamis dan kontekstual. Generative AI (GenAI) adalah kategori AI yang mampu menghasilkan konten baru, seperti teks, gambar, audio, atau video, yang kadang sulit dibedakan dari konten yang dibuat oleh manusia. GenAI memungkinkan chatbot untuk menghasilkan respons yang tidak hanya *retrieved* (diambil dari basis data) tetapi juga

generated (dibuat secara *on-the-fly*). Hal tersebut memungkinkan percakapan yang lebih fleksibel, alami, dan tidak terbatas pada skrip kode yang kaku (Akpan, et. al., 2025).

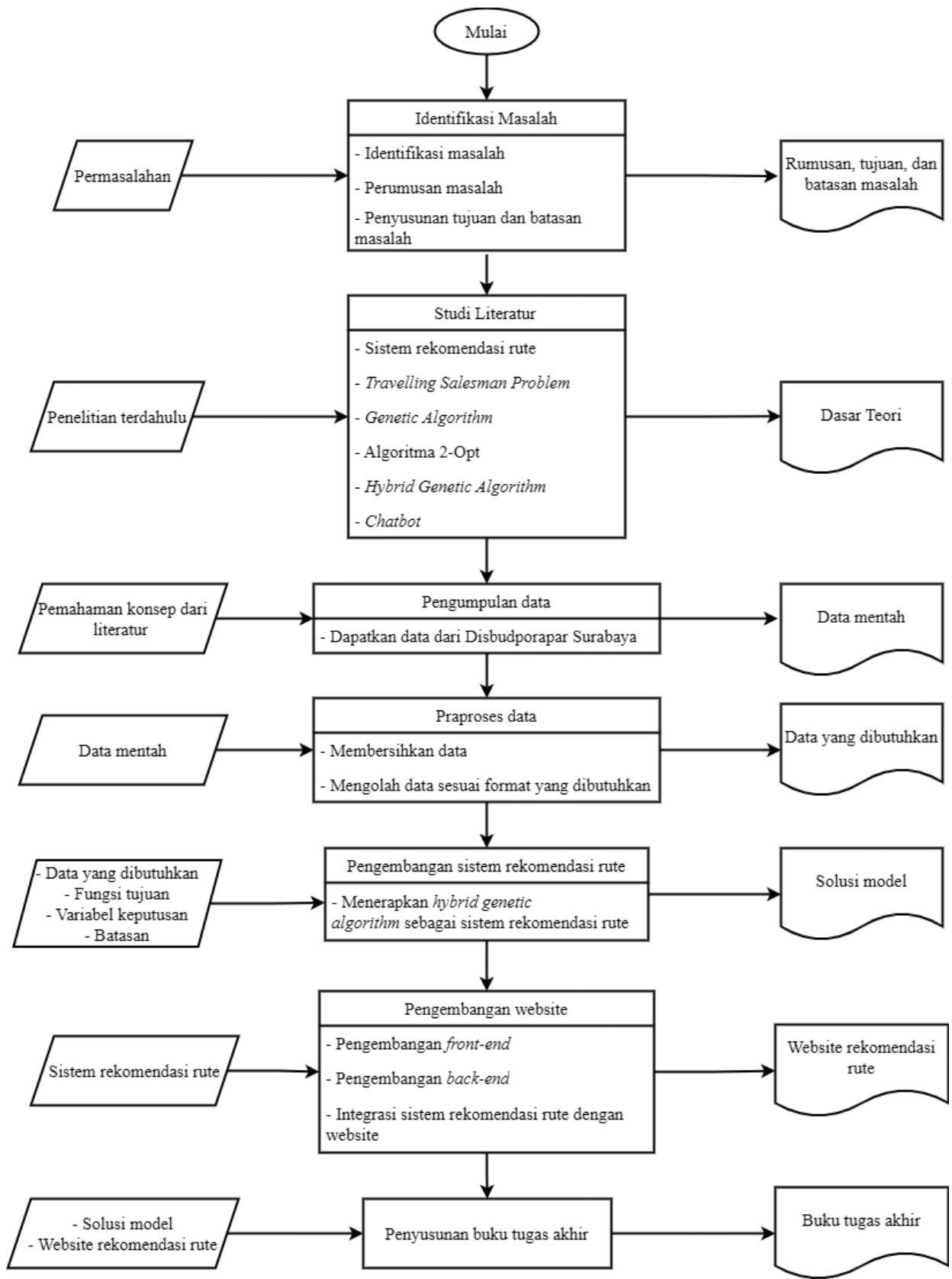
Salah satu model GenAI yang tersedia dalam bentuk API adalah API Gemini milik Google. API Gemini menawarkan berbagai model yang dioptimalkan untuk kasus penggunaan tertentu. Salah satu model yang dapat digunakan dalam pengembangan chatbot adalah Gemini 2.5 Flash yang dapat menerima *input* teks, gambar, video, dan audio. Selain itu, model tersebut juga memiliki fitur berpikir yang membuat chatbot bisa menjawab pertanyaan dari pengguna dengan lebih rasional. Google GenAI menyediakan *Software Development Kit* (SDK) untuk Javascript sehingga Google GenAI dapat dengan mudah diintegrasikan di sebuah website. Google Gen AI SDK dapat diterapkan di Typescript atau Javascript dengan menginstallnya terlebih dahulu melalui *Node Package Manager* (NPM) dengan perintah `npm install @google/genai` (Google AI for Developers, 2025).

BAB 3 METODOLOGI

Pada bab ini dijelaskan bagaimana tugas akhir ini akan dikembangkan. Bab ini dimulai dengan penjelasan metode yang digunakan dalam pengembangan tugas akhir ini. Seluruh alur pengembangan tugas akhir ini dijelaskan dalam bab ini. Selain itu, dijelaskan pula urutan pelaksanaan tugas akhir yang ditunjukkan dengan tabel jadwal pelaksanaan tugas akhir ini.

3.1 Metode yang Digunakan

Pada subbab ini dijelaskan metode yang digunakan dan proses-proses yang dilakukan dalam pengembangan sistem rekomendasi rute pada tugas akhir ini. Diagram alir pengembangan tugas akhir ini ditunjukkan pada gambar 3.1 sebagai gambaran umum proses pelaksanaan tugas akhir.



Gambar 3.1 Diagram alir pengembangan tugas akhir

Subsubbab berikutnya akan dijelaskan setiap proses dari diagram alir pengembangan tugas akhir. Penjelasan dimulai dari identifikasi masalah, studi literatur, pengumpulan data,

praproses data, pengembangan sistem rekomendasi rute, pengembangan website, hingga penyusunan tugas akhir buku tugas akhir.

3.1.1 Identifikasi Masalah

Pengerjaan tugas akhir dimulai dengan mengidentifikasi masalah pada topik yang diangkat pada tugas akhir ini. Setelah itu, dirumuskan masalah apa saja yang akan diselesaikan dalam tugas akhir ini. Dalam tugas akhir ini diusulkan sebuah sistem rekomendasi rute wisata halal di Surabaya yang diharapkan dapat meningkatkan kunjungan wisatawan ke Surabaya karena kemudahan dalam menjelajahi wisata halal ditambah dengan adanya *chatbot* sebagai asisten virtual.

Sistem rekomendasi rute yang dikembangkan pada tugas akhir ini merekomendasikan wisata halal yang ada di Surabaya, dengan tujuan meminimalkan jarak rute dari seluruh lokasi yang dipilih pengguna sebagai destinasi tujuan. Selain itu, terdapat batasan yang ditentukan dalam sistem rekomendasi rute ini yakni, seluruh destinasi di rute harus ditempuh dalam waktu maksimal satu hari.

3.1.2 Studi Literatur

Dari masalah yang telah diidentifikasi, berikutnya dilakukan studi literatur untuk mencari informasi yang digunakan oleh penulis untuk menyelesaikan masalah pada tugas akhir ini. Masukan pada tahap ini adalah penelitian-penelitian terdahulu yang terkait dengan sistem rekomendasi rute, *travelling salesman problem*, *genetic algorithm*, algoritma 2-opt, *hybrid genetic algorithm*, dan *chatbot*. Luaran dari tahap studi literatur pada tugas akhir ini adalah dasar teori yang kemudian digunakan sebagai landasan pengetahuan dalam pengembangan sistem rekomendasi rute di tugas akhir ini.

3.1.3 Pengumpulan Data

Setelah dasar teori dihasilkan dari tahap studi literatur, berikutnya pemahaman dari dasar teori tersebut digunakan sebagai masukan pada tahap pengumpulan data untuk menentukan bagaimana data yang dibutuhkan untuk pengembangan sistem rekomendasi rute. Dalam tugas akhir ini, data wisata Surabaya didapatkan dari data wisata yang diterbitkan oleh Disbudporapar Surabaya dan mencari secara manual melalui Google Maps. Kemudian, data restoran halal di Surabaya didapatkan dari *website* WaveHalal yang berisi daftar lokasi gerai makanan halal di Surabaya. Luaran yang dihasilkan dari tahap pengumpulan data adalah data mentah yang akan diolah pada tahap praproses data.

3.1.4 Praproses Data

Pada tahap praproses data digunakan data mentah yang didapatkan dari tahap pengumpulan data sebagai masukan. Pada tahap ini data mentah dibersihkan dan diolah untuk menghasilkan data yang sesuai dengan kebutuhan sistem rekomendasi rute pada tugas akhir ini. Proses pembersihan dilakukan untuk menangani nilai yang kosong (*null*) atau tidak valid. Sedangkan proses pengolahan data dilakukan dengan menyaring data wisata dan gerai makanan yang sudah mendapatkan sertifikasi halal dengan mencarinya di *website* halalmui.org dari Majelis Ulama Indonesia (MUI) atau *website* wave.halal.go.id (Rakhmawati, 2024). Kemudian, berikutnya data wisata dan gerai makanan halal diolah untuk mendapatkan nama destinasi wisata, gerai makanan halal dan titik koordinatnya (garis lintang, garis bujur) sehingga dihasilkan luaran data yang dibutuhkan dalam pengembangan sistem rekomendasi rute pada tugas akhir ini.

3.1.5 Pengembangan Sistem Rekomendasi Rute

Tahap ini adalah tahap sistem rekomendasi rute dikembangkan dengan menggunakan HGA. Masukan yang dibutuhkan pada tahap ini adalah data yang telah diolah dari tahap praproses data, fungsi tujuan, variabel keputusan, dan batasan-batasan yang ditetapkan berdasarkan latar belakang, rumusan masalah dan penetapan tujuan, serta tinjauan pustaka yang telah dilakukan di tahap-tahap sebelumnya.

Sistem rekomendasi rute dikembangkan dengan HGA yang mengombinasikan GA dengan algoritma 2-opt. Adapun tahap-tahap pada HGA dalam sistem rekomendasi ini yakni:

1. Representasi genetik

Pada tahap ini ditentukan cara solusi direpresentasikan sebagai kromosom berdasarkan data yang telah dihasilkan dari tahap praproses data. Misalnya, setiap kromosom merepresentasikan sebuah rute wisata dan tiap destinasi wisata direpresentasikan oleh gen-gen pada kromosom tersebut.

2. Inisialisasi populasi

Pada tahap ini populasi awal diinisialisasi dari kumpulan kromosom dan gen-gen yang telah ditentukan dari tahap sebelumnya.

3. Fungsi *fitness*

Tahap ini digunakan fungsi *fitness* untuk mengukur dan mengevaluasi nilai (jarak rute) yang telah dihasilkan dari populasi. Fungsi *fitness* akan menilai seberapa optimal jarak tempuh sebuah rute. Oleh karena itu, fungsi *fitness* bertujuan untuk membandingkan solusi dari setiap generasinya sehingga dapat ditentukan mana solusi terbaik atau mendekati optimal.

4. Seleksi

Pada tahap ini akan dipilih individu-individu dari populasi saat ini berdasarkan nilai *fitness* yang terbaik.

5. *Crossover*

Pada tahap *crossover* atau persilangan dilakukan operasi untuk menghasilkan keturunan (individu baru) yang berasal dari hasil penggabungan individu-individu yang telah diseleksi.

6. Mutasi

Pada tahap ini diterapkan operator mutasi pada keturunan yang sudah dihasilkan dari proses crossover. Mutasi dilakukan dengan probabilitas yang kecil untuk menjaga keragaman genetik dan membantu keluar dari jebakan lokal optimal.

7. Pencarian lokal

Pada tahap ini akan diambil keturunan yang dihasilkan dari tahap *crossover* dan telah dilakukan mutasi, lalu diterapkan algoritma 2-Opt kepada keturunan tersebut. Hal ini dapat memperbaiki keturunan tersebut menjadi versi yang lebih optimal secara lokal.

8. Generasi populasi baru

Setelah individu dihasilkan dari tahap seleksi, *crossover*, mutasi, dan pencarian lokal, berikutnya individu/keturunan baru tersebut dimasukkan ke dalam populasi. Pada tahap ini juga dilakukan operasi *elitist* dengan menyalin satu atau beberapa individu terbaik dari populasi lama secara langsung ke populasi baru. Ini memastikan bahwa solusi terbaik yang pernah ditemukan tidak akan hilang.

9. Konvergensi

Hal terpenting yang perlu dilakukan dalam GA atau HGA adalah mengulang kembali langkah-langkah sebelumnya hingga mencapai konvergensi, di mana solusi-solusi yang

dihasilkan sudah cukup baik atau mendekati optimal. Hal tersebut dapat ditandai dengan fungsi *fitness* yang tidak lagi menghasilkan solusi yang lebih baik dari generasi sebelumnya.

Sistem rekomendasi rute yang dikembangkan mempertimbangkan batasan-batasan dan variabel keputusan dengan fungsi tujuan sebagai berikut:

1. Fungsi tujuan
 - a. Meminimalkan jarak tempuh rute. Jarak tempuh rute diukur dengan fungsi *fitness* pada HGA.
2. Variabel keputusan
 - a. Kunjungan dari satu titik ke titik lainnya.
3. Batasan-batasan
 - a. Terdapat titik awal dan akhir dalam rute.
 - b. Waktu tempuh maksimal rute selama satu hari.
 - c. Tidak ada *subtours*.
 - d. Dalam satu rute mencakup minimal tiga destinasi wisata kuliner dengan menu makanan berat dan halal, tiga destinasi wisata non-kuliner, dua lokasi penjualan makanan ringan atau cemilan, dan tepat satu lokasi penjualan oleh-oleh atau souvenir.
 - e. Rute yang dihasilkan dari sistem rekomendasi rute ini berurut dengan notasi urutan $K_1, C_1, W_1, K_2, W_2, C_2, K_3, O$. Penjelasan setiap notasi dalam urutan rute tersebut dijelaskan pada tabel 1.1.

Untuk melihat kinerja HGA dari model sistem rekomendasi yang telah dihasilkan pada tahap ini, berikutnya akan dievaluasi dengan dibuatkan grafik konvergensi. Grafik konvergensi menunjukkan perbaikan solusi setiap generasi. Dengan demikian, luaran dari tahap pengembangan sistem rekomendasi rute adalah solusi model sistem rekomendasi rute yang dapat diimplementasikan ke dalam sebuah website.

3.1.6 Pengembangan Website

Dalam tugas akhir ini, sistem rekomendasi rute akan diimplementasikan pada sebuah website. Pengembangan website dilakukan dengan menggunakan kerangka kerja Next.js yang mendukung pengembangan aplikasi web mulai dari *frontend* hingga *backend*. Masukan yang digunakan dalam pengembangan website adalah model sistem rekomendasi rute yang telah dikembangkan pada tahap sebelumnya. Dalam tahap ini, sistem rekomendasi rute dan chatbot akan diintegrasikan dengan website sehingga dapat dihasilkan website rekomendasi rute wisata halal di Surabaya.

3.1.7 Penyusunan Buku Tugas Akhir

Tahap penyusunan buku tugas akhir dilakukan untuk mendokumentasikan seluruh hal yang dilakukan pada tugas akhir ini. Penyusunan buku tugas akhir mengikuti format dan ketentuan penulisan buku tugas akhir berdasarkan pedoman penyusunan tugas akhir program sarjana (S1) Institut Teknologi Sepuluh Nopember (ITS). Masukan dari tahap ini adalah seluruh hal yang telah dilakukan selama pengerjaan tugas akhir, terutama solusi model dan sistem rekomendasi rute dan website rekomendasi rute. Dengan demikian, luaran yang dihasilkan dari tahap ini adalah buku tugas akhir yang mencakup seluruh hal dalam pengerjaan tugas akhir ini, mulai dari bab pendahuluan, tinjauan pustaka, metodologi, hasil dan pembahasan, hingga kesimpulan dan saran.

3.2 Urutan Pengerjaan Tugas Akhir

Pada subbab ini dilampirkan rincian jadwal pengerjaan tugas akhir ini dalam jangka waktu tiga bulan seperti yang ditunjukkan pada tabel 3.1.

Tabel 3.1 Jadwal pengerjaan tugas akhir

No.	Kegiatan	Bulan											
		I				II				III			
		1	2	3	4	1	2	3	4	1	2	3	4
1	Identifikasi masalah	■	■										
2	Studi literatur	■	■										
3	Pengumpulan data			■									
4	Praproses data				■	■							
5	Pengembangan algoritma sistem rekomendasi rute							■	■	■			
6	Pengembangan website									■	■	■	
7	Penyusunan laporan tugas akhir	■	■	■	■	■	■	■	■				

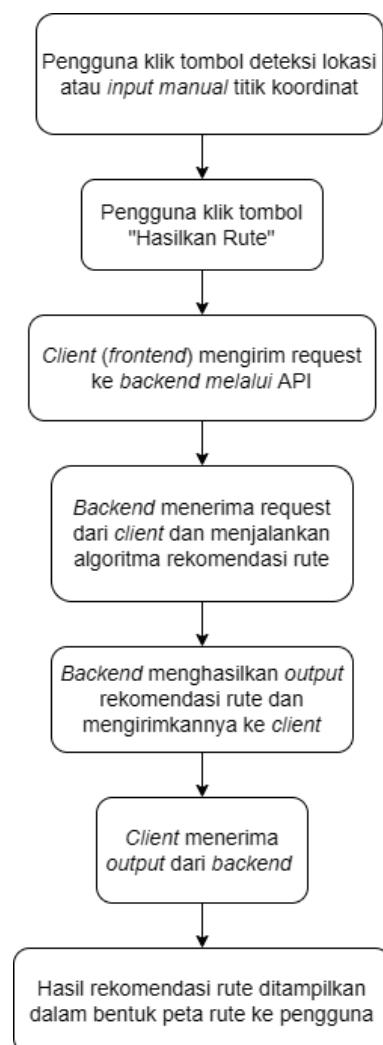
Jadwal tersebut disusun berdasarkan perkiraan durasi yang diperlukan `untuk melakukan setiap tahapan pada metodologi dan kemampuan penulis dalam pengerjaan tugas akhir secara berurutan.

BAB 4 Perancangan

Pada bab ini akan disajikan desain perancangan sistem rekomendasi rute berdasarkan metodologi yang telah dijelaskan pada bab sebelumnya. Tahapan perancangan sistem rekomendasi rute dijelaskan secara runut pada bab ini, mulai dari perancangan arsitektur sistem, pengumpulan dan deskripsi data, perancangan algoritma sistem rekomendasi rute, hingga perancangan uji coba sistem.

4.1 Perancangan Arsitektur *Website*

Arsitektur sistem dikembangkan dengan tujuan untuk menghasilkan *website* interaktif yang dapat digunakan oleh pengguna untuk menghasilkan rekomendasi rute wisata halal di Surabaya. Oleh karena itu, karena hasil akhir dari tugas akhir ini adalah sebuah *website*, maka dalam pengembangan tugas akhir ini terdapat dua komponen utama yang dikembangkan, yakni, *frontend* dan *backend*. Gambar 4.1 menunjukkan alur pengguna dalam menghasilkan rekomendasi rute pada *website* sistem rekomendasi rute wisata di tugas akhir ini.



Gambar 4.1 Alur Pengguna untuk Menghasilkan Rekomendasi Rute Wisata pada *Website*

Dilakukan pemisahan *frontend* dan *backend* bertujuan untuk mengoptimalkan kinerja *website* dan sistem rekomendasi sehingga *frontend* hanya berfokus untuk menampilkan antarmuka *website* saja dan *backend* berfokus untuk menghasilkan rekomendasi rute.

4.1.1 *Frontend*

Frontend dikembangkan dengan Next.js versi 16.0.1 yang merupakan *framework* React yang dapat digunakan untuk membangun aplikasi *website* yang siap produksi dan ramah mesin pencari (*SEO-friendly*) tanpa konfigurasi manual yang rumit. Dalam tugas akhir ini, *frontend* berfungsi untuk menampilkan antarmuka *website* ke pengguna dan mengirimkan data koordinat pengguna ke *backend* untuk kemudian dihasilkan rekomendasi rute wisata halal di Surabaya.

Data koordinat pengguna akan dikirimkan ke *backend* melalui API yang dikembangkan di *backend*. Data yang dikirimkan ke *backend* hanya titik koordinat saja, yakni, *latitude* dan *longitude*. Berikut adalah payload yang dikirimkan dari *client (frontend)* ke *backend*.

Tabel 4.1 Request yang Dikirimkan dari *Frontend* ke *Backend*

Field	Tipe Data	Contoh Value
latitude	Double	-7.2458
longitude	Double	112.7378

Koordinat pengguna tersebut kemudian akan diterima *backend* dan kemudian digunakan oleh *backend* sebagai titik awal rute sehingga dapat dihasilkan rekomendasi rute wisata berdasarkan titik koordinat pengguna.

4.1.2 *Backend*

Backend dikembangkan dengan menggunakan Python versi 3.11+. Bahasa Python dipilih karena kemudahan dalam pengembangan, banyak library yang dapat digunakan dalam pengembangan HGA, dan dukungan komunitas yang besar. Tabel 4.2 menunjukkan daftar *library* yang digunakan dalam pengembangan sistem rekomendasi rute wisata dalam tugas akhir ini.

Tabel 4.2 Daftar Library yang Digunakan dalam Pengembangan HGA

No.	Nama <i>Library/Alat</i>	Versi	Kategori	Deskripsi
1	Python	3.11+	Runtime	Bahasa pemrograman utama.
2	FastAPI	0.104.1+	<i>Web framework</i>	Framework untuk membuat REST API dengan auto-documentation, validasi otomatis, dan performa tinggi.
3	uvicorn	0.24.0+	Server ASGI	Server production untuk menjalankan aplikasi FastAPI dengan dukungan async dan high concurrency.
4	pydantic	2.x	Validasi	Library untuk validasi data request/response secara otomatis menggunakan Python type hints.
5	pandas	2.1.3+	Pemrosesan Data	Manipulasi dan analisis data destinasi wisata dalam format tabular.

6	requests	2.31.0+	HTTP Client	Melakukan HTTP <i>request</i> ke OSRM API untuk mendapatkan jarak rute nyata dengan <i>retry mechanism</i> .
7	python-multipart	0.0.6+	<i>Uploadfile</i>	<i>Handler</i> untuk menerima form-data dan <i>file uploads</i> di API <i>endpoints</i> .
8	OSRM API	API publik	Mesin <i>routing</i>	API untuk menghitung dan mendapatkan rute nyata.
9	json	<i>Built-in</i>	<i>Data format</i>	Parsing file JSONL untuk data destinasi dan <i>serialization distance matrix cache</i> .
10	math	<i>Built-int</i>	Kalkulasi	Perhitungan trigonometri (sin, cos, tan) untuk formula <i>Haversine distance</i> .
11	random	<i>Built-in</i>	<i>Randomization</i>	Operasi random untuk inisialisasi populasi, <i>crossover</i> , mutasi, dan seleksi dalam HGA.
12	dataclasses	<i>Built-in</i>	Struktur data	Membuat <i>data classes</i> untuk model Destination dengan sintaks yang ringkas.
13	contextlib	<i>Built-in</i>	<i>Context manager</i>	<i>Async context manager</i> untuk <i>handle startup/shutdown events</i> di FastAPI lifespan

Dalam tugas akhir ini *backend* berfungsi untuk menerima *request* dari *frontend* berupa titik koordinat dan kemudian titik koordinat tersebut digunakan untuk menghasilkan rekomendasi rute wisata halal di Surabaya dengan menjalankan HGA. Untuk menerima *request* dari *client*, dalam *backend* dikembangkan API dengan menggunakan FastAPI dengan endpoint /api/generate-routes. Titik koordinat pengguna dikirimkan dengan menggunakan request POST dengan contoh payload yang ditunjukkan oleh kode 4.1.

```
{
  latitude: -7.2458,
  longitude: 112.7378,
}
```

Kode 4.1 Contoh *Payload Request* pada Endpoint /api/generate-routes

Selain untuk menghasilkan rekomendasi rute, *backend* juga digunakan untuk menyimpan data destinasi dengan tipe file jsonl. Data destinasi yang disimpan di *backend* berfungsi untuk ditampilkan di *frontend* sehingga terdapat juga *endpoint* api untuk mengambil data destinasi dan dikirimkan ke *frontend*, yakni, pada endpoint /api/destinations. Untuk mendapatkan data destinasi, *client* cukup melakukan *request* GET pada *endpoint* /api/destinations.

4.2 Pengumpulan dan Deskripsi Data

Data wisata yang digunakan dalam sistem rekomendasi rute wisata pada tugas akhir ini didapatkan dari Disbudporapar Kota Surabaya dan Google Maps. Kemudian, data restoran halal didapatkan dari website WaveHalal. Data wisata yang didapatkan dari Google Maps didapatkan dengan cara mencari destinasi satu per satu dengan mencari *keywords* seperti “oleh-oleh”, “monumen”, atau “museum”. Data yang diambil untuk diolah dalam HGA yakni, nama destinasi, kategori, latitude, longitude, alamat, link gambar, dan deskripsi. Tabel 4.3 menunjukkan potongan dataset yang digunakan dalam tugas akhir ini.

Tabel 4.3 Potongan Dataset

id	nama_destinasi	kategori	latitude	longitude
1	Pusat Oleh-oleh Bu Rudy	oleh_oleh	-7.2673018	112.7697516
2	Annyeong Market	non_kuliner	-7.2538255	112.7894579
3	Kenjeran Park	non_kuliner	-7.252296	112.796695
4	Surabaya Kriya Gallery (SKG) MERR	non_kuliner	-7.3060589	112.7806015
5	Pasar Bunga Kayoon	oleh_oleh, non_kuliner	-7.268888	112.747916
6	Spikoe Resep Kuno	makanan_ringan	-7.3311105	112.7742819
7	Burger Up Dharmahusada	makanan_ringan	-7.148586	112.793884
8	Sate Asin Garuda	makanan_berat	-7.280941	112.746563
9	Toko Oleh Oleh Panen Raya	oleh_oleh	-7.258628934	112.7396334
10	Pusat Oleh Oleh Khas Surabaya	oleh_oleh	-7.267355995	112.7700856

Dari data yang telah dikumpulkan, didapatkan 221 destinasi wisata yang dapat digunakan sebagai kombinasi destinasi sehingga menghasilkan rekomendasi rute wisata bagi pengguna. Kolom id berfungsi sebagai penanda/*identifier* data, kolom nama_destinasi berisi nama dari destinasi yang ada, kolom kategori berfungsi sebagai penanda kategori destinasi agar dapat dihasilkan rute wisata sesuai urutan yang ada pada batasan tugas akhir ini, lalu kolom latitude dan longitude adalah data utama yang digunakan sebagai titik koordinat pada HGA. Dengan kolom latitude dan longitude dapat dihitung matriks jarak dengan menggunakan OSRM API.

4.2.1 Matriks Jarak

Dalam pengembangan sistem rekomendasi rute wisata dengan HGA, matriks jarak berfungsi sebagai skor yang diperhitungkan dalam HGA sehingga dapat ditentukan fitness value berdasarkan penalti jarak (maksimal 20 km) pada kromosom dalam setiap generasinya. Matriks jarak dihasilkan dari OSRM API sehingga didapatkan jarak nyata antar setiap titik destinasi. Matriks jarak didapatkan dengan mencari jarak setiap kombinasi antar titik koordinat dari 221 titik destinasi. Tabel 4.4 menunjukkan potongan data matriks jarak antar destinasi pada tugas akhir ini.

Tabel 4.4 Cuplikan Data Matriks Jarak Antar Destinasi

No.	Titik 1	Titik 2	Jarak (Km)
1	-7.275697, 112.780625	-7.267302, 112.769752	2.077
2	-7.275697, 112.780625	-7.253825, 112.789458	4.5199
3	-7.275697, 112.780625	-7.252296, 112.796695	4.7067
4	-7.275697, 112.780625	-7.253240, 112.802719	4.195699
5	-7.298964, 112.735790	-7.275697, 112.780625	11.1759
6	-7.293526, 112.732124	-7.275697, 112.780625	8.5069
7	-7.275697, 112.780625	-7.250181, 112.737188	8.103
8	-7.289402, 112.783431	-7.275697, 112.780625	3.06169
9	-7.277252, 112.769133	-7.275697, 112.780625	2.1094
10	-7.275697, 112.780625	-7.266069, 112.761730	2.9954

Dari 221 destinasi yang ada, dihasilkan sekitar 24091 data matriks jarak antar destinasi sehingga perhitungan jarak dan evaluasi *fitness* dapat lebih akurat. Kalkulasi penalti jarak antar destinasi digunakan dari matriks jarak, sedangkan jarak dari titik koordinat pengguna ke titik koordinat destinasi pertama bersifat dinamis atau bisa berubah-ubah sehingga tidak bisa dibuat dalam matriks jarak. Oleh karena itu, untuk menghitung jarak dari titik koordinat pengguna ke titik koordinat destinasi pertama digunakan kalkulasi dengan menggunakan metode *haversine distance*.

4.2.2 Matriks Waktu Tempuh

Dalam pengembangan sistem rekomendasi rute wisata dengan HGA, matriks waktu berfungsi sebagai skor yang diperhitungkan dalam HGA sehingga dapat ditentukan fitness value berdasarkan penalti waktu (maksimal 5 jam atau 300 menit) pada kromosom dalam setiap generasinya. Matriks jarak dihasilkan dari OSRM API sehingga didapatkan waktu tempuh nyata antar setiap titik destinasi. Matriks jarak didapatkan dengan mencari jarak setiap kombinasi antar titik koordinat dari 221 titik destinasi. Tabel 4.5 menunjukkan potongan data matriks jarak antar destinasi pada tugas akhir ini.

Tabel 4.5 Cuplikan Data Matriks Waktu Tempuh Antar Destinasi

No.	Titik 1	Titik 2	Waktu (menit)
1	-7.275697, 112.780625	-7.267302, 112.769752	3.52
2	-7.275697, 112.780625	-7.253825, 112.789458	6.36
3	-7.275697, 112.780625	-7.252296, 112.796695	6.29
4	-7.275697, 112.780625	-7.253240, 112.802719	9.18
5	-7.298964, 112.735790	-7.275697, 112.780625	13.61

6	-7.293526, 112.732124	-7.275697, 112.780625	11.04
7	-7.275697, 112.780625	-7.250181, 112.737188	10.0
8	-7.289402, 112.783431	-7.275697, 112.780625	4.38
9	-7.277252, 112.769133	-7.275697, 112.780625	5.47
10	-7.275697, 112.780625	-7.266069, 112.761730	4.53

Dari 221 destinasi yang ada, dihasilkan sekitar 24091 data matriks waktu tempuh antar destinasi sehingga perhitungan waktu tempuh dan evaluasi fitness dengan penalti waktu dapat lebih akurat. Kalkulasi penalti waktu tempuh antar destinasi digunakan dari matriks waktu tempuh, sedangkan waktu tempuh dari titik koordinat pengguna ke titik koordinat destinasi pertama digunakan perhitungan estimasi karena titik koordinat pengguna bersifat dinamis atau bisa berubah-ubah sehingga tidak bisa dibuat dalam matriks waktu tempuh. Kemudian untuk waktu tempuh hasil rekomendasi yang ditampilkan ke pengguna nanti menggunakan waktu tempuh keseluruhan rute yang dikalkulasi ulang menggunakan OSRM API saat rekomendasi rute telah dihasilkan.

4.3 Perancangan Algoritma Sistem Rekomendasi Rute

Algoritma sistem rekomendasi rute pada tugas akhir ini dikembangkan dengan metode *Hybrid Genetic Algorithm* (HGA) yang mengombinasikan metode *Genetic Algorithm* (GA) untuk pencarian global dengan algoritma 2-Opt untuk pencarian lokal. HGA akan dikembangkan dengan merepresentasikan kromosom yang mampu merepresentasikan solusi rute sesuai dengan batasan-batasan urutan kategori yang telah ditetapkan pada bab 3. Dengan demikian, perlu dirancang juga operator genetik yang mempertahankan validitas batasan-batasan selama proses evolusi. Kemudian, karena HGA mengombinasikan GA dengan algoritma 2-Opt, maka perlu dirancang pula mekanisme hibridasi antara GA dan algoritma 2-Opt untuk optimasi rute yang lebih baik.

4.3.1 Perancangan Kromosom

Kromosom direpresentasikan sebagai array dari kumpulan gen, dimana setiap gen berisi rute wisata yang berisi destinasi-destinasi wisata dengan urutan kategori wisata yang harus sesuai dengan batasan yang ada di bab 3. Berikut dipaparkan struktur dan atribut dari kromosom pada sistem rekomendasi wisata di tugas akhir ini.

4.3.1.1 Struktur Kromosom

Kromosom didefinisikan sebagai array dari kumpulan gen. Definisi formal dari sebuah kromosom dicontohkan pada persamaan (4.1).

$$\text{Kromosom } C = [g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8] \quad (4.1)$$

Untuk memahami definisi formal dari sebuah kromosom, tabel 4.6 berikut memaparkan keterangan dan deskripsi dari persamaan (4.1)

Tabel 4.6 Keterangan dan deskripsi dari persamaan (4.1)

Keterangan	Deskripsi
$g_i \in D$	g_i merupakan gen ke- i dimana i adalah bilangan bulat dan D adalah himpunan destinasi wisata.

$ C = 8$	$ C $ adalah panjang kromosom dimana panjangnya harus tetap untuk memenuhi batasan-batasan yang telah ditetapkan di bab 3.
$g_i \neq g_j$ untuk $i \neq j$	Tidak terdapat destinasi duplikat dimana i dan j adalah bilangan bulat.

Kromosom diinisiasi dengan melakukan pemilihan secara acak dari dataset untuk setiap gen pada kromosom dengan melakukan penyaringan pada kategori destinasi untuk menyesuaikan urutan sesuai kategori pada kromosom, dan untuk memastikan tidak ada destinasi yang duplikat dalam satu rute, maka setiap id destinasi yang sudah dimasukkan ke dalam kromosom, dimasukkan ke dalam sebuah variabel yang berisi list id destinasi yang sudah digunakan dalam kromosom agar variabel tersebut digunakan sebagai penyaring destinasi yang tidak diikutkan untuk inisiasi berikutnya. Dengan mendefinisikan kromosom seperti demikian, diharapkan hasil rekomendasi rute wisata dari sistem ini dapat memenuhi batasan-batasan yang telah ditetapkan sebelumnya.

4.3.1.2 Atribut Kromosom

Dalam tugas akhir ini, setiap kromosom memiliki atribut-atribut yang digunakan untuk menjalankan HGA. Tabel 4.7 menunjukkan atribut-atribut yang digunakan dalam tugas akhir ini beserta deskripsinya.

Tabel 4.7 Atribut-atribut dari kromosom

Atribut	Tipe Data	Deskripsi
genes	List[Destination]	Urutan destinasi dalam rute
start_point	Tuple[float, float]	Koordinat titik awal (<i>latitude, longitude</i>)
fitness_value	float atau none	Nilai <i>fitness</i>
penalty_value	float atau none	Nilai penalty jika terdapat pelanggaran pada batasan-batasan
_total_distance	float atau none	Cache total jarak rute (km)
_total_time	float atau none	Cache total waktu tempuh (menit)

Nilai *fitness* pada kromosom merepresentasikan satuan untuk menilai apakah sebuah kromosom sudah memiliki hasil yang cukup optimal dan sebagai pembanding dengan kromosom lainnya.

4.3.2 Perancangan Fungsi *Fitness*

Fungsi *fitness* digunakan untuk menghitung nilai *fitness* yang berfungsi untuk menilai kualitas suatu solusi. Fungsi *fitness* dikembangkan dengan mekanisme penalti jika ada jarak dan waktu tempuh yang melebihi batasan. Pada bagian ini akan dijelaskan terkait perancangan fungsi *fitness* yang mencakup formulasi fungsi *fitness* dan perhitungan jarak dalam fungsi *fitness*.

4.3.2.1 Formulasi Fungsi *Fitness*

Dalam HGA pada tugas akhir ini fungsi *fitness* dirancang berbanding terbalik dengan total jarak rute dengan tujuan untuk mengukur bahwa jarak yang lebih pendek mengindikasikan *fitness* yang lebih tinggi (minimasi). Persamaan (4.2) menunjukkan formula fungsi fitness pada HGA di tugas akhir ini.

$$f(C) = \frac{1}{D_{total}(C)(1+P_{total}(C))} \quad (4.2)$$

Pada persamaan (4.2), $D_{total}(C)$ adalah total jarak rute yang dihitung dengan persamaan (4.3).

$$D_{total}(C) = d(A, g_1) + \sum_{i=1}^7 d(g_i, g_{i+1}) \quad (4.3)$$

Untuk memahami persamaan (4.3), tabel 4.8 menunjukkan keterangan dan deskripsi dari persamaan (4.3).

Tabel 4.8 Keterangan dan Deskripsi dari Persamaan (4.3)

Keterangan	Deskripsi
A	Titik awal yang merupakan lokasi yang ditentukan oleh pengguna
g_i	Destinasi ke- i dalam kromosom, dimana i adalah bilangan bulat
$d(g_i, g_j)$	Fungsi untuk menghitung jarak antara g_i dan g_j
$P_{total}(C)$	Total <i>penalty</i> dari semua pelanggaran batasan-batasan

Dengan rancangan fungsi *fitness* demikian, solusi yang melanggar batasan-batasan akan memiliki *fitness* yang lebih rendah karena nilai penalti berbanding terbalik dengan nilai *fitness*. Persamaan (4.4) menunjukkan bagaimana total penalti dikalkulasikan.

$$P_{total}(C) = P_{distance}(C) + P_{time}(C) \quad (4.4)$$

Total penalti dihasilkan dari hasil penjumlahan dari *penalty* dari pelanggaran batasan jarak dan batasan waktu. Penalti jarak akan bernilai nol jika jarak total (D_{total}) pada suatu rute bernilai kurang dari jarak maksimal (D_{max}). Jarak maksimal berdasarkan batasan yang telah ditetapkan di bab 1 adalah 20 km. Persamaan (4.5) menunjukkan kalkulasi dari penalti jarak jika jarak total dari rute yang dihasilkan lebih dari 20 km.

$$P_{distance}(C) = w_d \times \left(\frac{D_{total} - D_{max}}{D_{max}} \right)^2 \quad (4.5)$$

Untuk memahami persamaan (4.5), tabel 4.9 menunjukkan keterangan dan deskripsi dari persamaan (4.5).

Tabel 4.9 Keterangan dan Deskripsi dari Pesamaan (4.5)

Keterangan	Deskripsi
$P_{distance}(C)$	Total penalti dari semua pelanggaran batasan jarak
w_d	Bobot penalti jarak dengan nilai 0.5
D_{total}	Jarak total dari rute hasil rekomendasi
D_{max}	Jarak maksimal rute yang merupakan batasan sistem, yakni, 20 km

Selain penalti jarak, terdapat juga penalti waktu tempuh yang akan bernilai nol jika waktu tempuh total bernilai kurang dari waktu tempuh maksimal yang ditentukan pada batasan di bab 2 adalah selama lima jam atau 300 menit. Persamaan (4.6) menunjukkan kalkulasi dari penalti waktu jika waktu tempuh total suatu rute kurang dari 300 menit.

$$P_{time}(C) = w_t \times \left(\frac{T_{total} - T_{max}}{T_{max}} \right)^2 \quad (4.6)$$

Untuk memahami persamaan (4.6), tabel 4.10 menunjukkan keterangan dan deskripsi dari persamaan (4.6).

Tabel 4.10 Keterangan dan Deskripsi dari Pesamaan (4.6)

Keterangan	Deskripsi
$P_{time}(C)$	Total penalti dari semua pelanggaran batasan waktu
w_t	Bobot penalti waktu dengan nilai 0.3
T_{total}	Waktu tempuh total dari rute hasil rekomendasi
T_{max}	Waktu tempuh maksimal rute yang merupakan batasan sistem, yakni, 300 menit

Pada implementasi penalti ini, semakin besar pelanggaran, maka penalti akan semakin besar secara kuadratik. Kode 4.2 menunjukkan *pseudocode* dari proses kalkulasi penalti di HGA pada tugas akhir ini.

INPUT:

- `total_distance_km`: Total jarak rute dalam km
- `total_time_minutes`: Total waktu tempuh dalam menit

OUTPUT:

- `total_penalty`: Nilai total penalti

KONSTANTA:

- `MAX_ROUTE_DISTANCE_KM` = 20.0
- `MAX_ROUTE_TIME_MINUTES` = 300.0
- `DISTANCE_PENALTY_WEIGHT` = 0.5
- `TIME_PENALTY_WEIGHT` = 0.3

LANGKAH-LANGKAH:

1. HITUNG PENALTI JARAK:

```
IF total_distance_km > MAX_ROUTE_DISTANCE_KM THEN:
    excess_distance ← total_distance_km - MAX_ROUTE_DISTANCE_KM
    penalty_ratio ← excess_distance / MAX_ROUTE_DISTANCE_KM
    distance_penalty ← DISTANCE_PENALTY_WEIGHT × (penalty_ratio)2
ELSE:
    distance_penalty ← 0
```

2. HITUNG PENALTI WAKTU:

```
IF total_time_minutes > MAX_ROUTE_TIME_MINUTES THEN:
    excess_time ← total_time_minutes - MAX_ROUTE_TIME_MINUTES
```

```

penalty_ratio ← excess_time / MAX_ROUTE_TIME_MINUTES
time_penalty ← TIME_PENALTY_WEIGHT × (penalty_ratio)2
ELSE:
    time_penalty ← 0

```

3. RETURN distance_penalty + time_penalty

Kode 4.2 *Pseudocode* Kalkulasi Penalti

Total penalti tersebut berikutnya akan digunakan dalam perhitungan *fitness*. Kode 4.3 menunjukkan *pseudocode* dari kalkulasi *fitness* dengan mekanisme penalti.

INPUT:

- base_fitness: Nilai fitness dasar
- total_penalty: Total penalty

OUTPUT:

- penalized_fitness: Fitness yang sudah dipenalti

LANGKAH-LANGKAH:

1. IF total_penalty ≤ 0 THEN:
RETURN base_fitness
2. penalized_fitness ← base_fitness / (1 + total_penalty)
3. RETURN penalized_fitness

Kode 4.3 *Pseudocode* Kalkulasi *Fitness* dengan Mekanisme Penalti

4.3.2.2 Perhitungan Jarak dan Waktu Tempuh

Untuk menghitung estimasi jarak dan waktu tempuh antara satu destinasi ke destinasi lainnya, digunakan dua pendekatan, yakni dengan menggunakan matriks jarak yang *pre-calculated* dan menggunakan perhitungan Haversine Distance untuk jarak dan perhitungan estimasi waktu ($\frac{\text{jarak (km)}}{\text{kecepatan (km/jam)}}$) dengan asumsi kecepatan rata-rata pengguna dalam berkendara adalah 50 km/jam. Matriks jarak dan waktu yang *pre-calculated* dihitung dengan memanfaatkan OSRM API untuk mengambil jarak dan waktu tempuh antar desintasi berdasarkan titik koordinat masing-masing destinasi. Jarak dan waktu tempuh yang diambil dari OSRM API merupakan jarak dan waktu tempuh antar destinasi berdasarkan rute nyata, dengan demikian, penggunaan OSRM API dapat meningkatkan akurasi estimasi jarak dan waktu tempuh pada sistem ini. Berikut adalah URL yang digunakan untuk mengambil jarak dan waktu tempuh antar destinasi, <http://router.project-osrm.org/route/v1/driving/{longitude1},{latitude1};{longitude2},{latitude2}>. Seluruh jarak dan waktu tempuh pada setiap kombinasi destinasi perlu dihitung atau didapatkan untuk kemudian dimasukkan ke dalam matriks jarak, dan nantinya matriks jarak digunakan dalam perhitungan fungsi *fitness*

Matriks jarak disimpan dalam format .json yang memiliki key berupa “latitude1, longitude1 | latitude2, longitude2” dengan tujuan agar bisa *bi-directional* dengan nilai dari tiap key adalah jarak dan waktu tempuh antar dua destinasi dalam kilometer dan kilometer per jam.

Selain menggunakan OSRM API, digunakan juga metode Haversine Distance untuk menghitung jarak dari titik awal pengguna ke destinasi pertama. Hal ini dilakukan karena pencatatan jarak dari titik awal pengguna ke destinasi pertama dengan menggunakan OSRM API tidak dimungkinkan untuk dilakukan karena titik awal pengguna yang dinamis. Selain itu, Haversine Distance juga digunakan sebagai *fallback* jika jarak dari suatu destinasi ke destinasi lain tidak ditemukan.

4.3.3 Perancangan Populasi

Populasi berisi kumpulan kromosom dan merupakan tempat untuk mencari solusi global yang mendekati optimal. Dalam bagian ini akan dibahas tentang struktur populasi dan proses inisialisasi populasi.

4.3.3.1 Struktur Populasi

Populasi didesfinisikan sebagai kumpulan kromosom. Persamaan (4.7) menunjukkan struktur populasi pada HGA di tugas akhir ini.

$$\text{Populasi } P = [C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8] \quad (4.7)$$

Pada persamaan (4.4), C_i adalah kromosom ke- i , dimana i adalah bilangan bulat. Ukuran populasi diukur berdasarkan parameter HGA. Parameter untuk ukuran populasi adalah bilangan bulat dan dijelaskan lebih lanjut pada subbab perancangan pengujian. Terdapat dua atribut dalam populasi di HGA pada tugas akhir ini seperti yang ditunjukkan oleh tabel 4.10.

Tabel 4.11 Atribut-Atribut Pada Populasi di HGA

Atribut	Tipe Data	Deskripsi
chromosomes	List[Chromosome]	Array dari kromosom-kromosom
population_size	int	Ukuran jumlah populasi

Atribut chromosomes berfungsi untuk menyimpan kumpulan kromosom dan population_size merupakan atribut untuk menentukan jumlah populasi yang digunakan dalam HGA.

4.3.3.2 Inisialisasi Populasi

Salah satu aspek yang menentukan kualitas solusi dalam *Genetic Algorithm* (GA) adalah inisiasi populasi sehingga dibutuhkan metode khusus untuk menetapkan populasi agar selalu konsisten untuk memenuhi batasan-batasan yang telah ditentukan. Pada bagian ini akan dijelaskan tentang algoritma untuk inisialisasi populasi. Pseudocode untuk inisialisasi populasi ditunjukkan oleh kode 4.4.

INPUT:

- all_destinations: List semua destinasi
- start_point: Koordinat awal
- population_size: Ukuran populasi target

OUTPUT:

- Population dengan population_size kromosom valid

LANGKAH-LANGKAH:

1. Kelompokkan destinasi berdasarkan kategori:

- makanan_berat → K_destinations
- makanan_ringan → C_destinations
- non_kuliner → W_destinations

- oleh_oleh → O_destinations
- all → All_destinations

2. Validasi ketersediaan destinasi:

```
IF count(K_destinations) < 3 THEN ERROR
IF count(C_destinations) < 2 THEN ERROR
IF count(W_destinations) < 2 THEN ERROR
IF count(O_destinations) < 1 THEN ERROR
```

3. FOR i = 1 TO population_size DO:

- 3.1. genes ← empty array
- 3.2. genes[1] ← random_choice(K_destinations)
- 3.3. genes[2] ← random_choice(C_destinations)
- 3.4. genes[3] ← random_choice(W_destinations ∪ All_destinations)
- 3.5. genes[4] ← random_choice(K_destinations \ {genes[1]})
- 3.6. genes[5] ← random_choice((W_destinations ∪ All_destinations) \ {genes[3]})
- 3.7. genes[6] ← random_choice(C_destinations \ {genes[2]})
- 3.8. genes[7] ← random_choice(K_destinations \ {genes[1], genes[4]})
- 3.9. genes[8] ← random_choice((O_destinations ∪ All_destinations) \ genes)
- 3.10. CREATE Chromosome(genes, start_point)
- 3.11. ADD to population

4. RETURN population

Kode 4.4 *Pseudocode* Inisialisasi Populasi HGA

Pada kode 4.4 ditunjukkan bahwa setiap kromosom dinisialisasi secara acak untuk memberi keberagaman pada populasi sehingga tidak terjadi konvergensi yang prematur dan dapat menghasilkan solusi yang mendekati optimal. Kemudian, setiap kromosom juga dijamin valid sejak awal dengan cara menempatkan destinasi dalam gen sesuai dengan kategorinya dan tidak ada destinasi yang duplikat dalam satu kromosom, hal ini ditunjukkan oleh langkah ketiga pada kode 4.4.

4.3.4 Perancangan Operator Seleksi

Dalam GA seleksi dilakukan untuk memilih kromosom atau individu yang kemudian akan dilakukan *crossover*. Pada HGA dalam tugas akhir ini, seleksi dilakukan dengan menggunakan metode *tournament selection*. *Tournament selection* merupakan metode untuk memilih individu dengan cara memilih subset acak dari populasi (*tournament*), kemudian memilih individu terbaik dari subset tersebut. Pseudocode dari algoritma untuk *tournament selection* pada tugas akhir ini ditunjukkan oleh kode 4.5.

INPUT:

- population: List kromosom
- tournament_size: Ukuran tournament (default: 8)

OUTPUT:

- selected_chromosome: Kromosom terpilih

LANGKAH-LANGKAH:

```

1. tournament ← random_sample(population, min(tournament_size, |population|))
2. best_chromosome ← NULL
3. best_fitness ← -∞

4. FOR EACH chromosome IN tournament DO:
    4.1. fitness ← chromosome.get_fitness()
    4.2. IF fitness > best_fitness THEN:
        best_fitness ← fitness
        best_chromosome ← chromosome

5. RETURN best_chromosome

```

Kode 4.5 Pseudocode *Tournament Selection*

Dengan menerapkan *tournament selection* pada HGA, diharapkan dapat meningkatkan kualitas solusi melalui pemilihan orang tua yang tepat pada bagian *crossover*.

4.3.5 Perancangan Operator *Crossover*

Dalam GA *crossover* bertujuan untuk menghasilkan individu-individu baru melalui persilangan dua atau lebih orang tua yang didapatkan dari hasil seleksi. Pada tugas akhir ini, digunakan metode *order crossover* yang dimodifikasi dengan menggunakan empat orang tua agar dapat mempertahankan urutan gen sesuai kategori pada kromosom sehingga solusi yang dihasilkan bisa mematuhi batasan-batasan yang telah ditetapkan dan juga diharapkan dapat meningkatkan keberagaman dan eksplorasi ruang solusi. *Order crossover* dilakukan dengan mempertahankan urutan relatif gen dari orang tua sambil mengambil subset dari orang tua lain. Kode 4.6 menunjukkan pseudocode dari *crossover* dalam HGA pada tugas akhir ini.

INPUT:

- parent1, parent2, parent3, parent4: Empat kromosom parent

OUTPUT:

- offspring1, offspring2: Dua kromosom offspring

LANGKAH-LANGKAH:

1. size ← 8
2. point1 ← random_int(0, size-1)
3. point2 ← random_int(point1+1, size)

Offspring 1: Kombinasi Parent1 dan Parent3

4. offspring1_genes ← [NULL] × size
5. offspring1_genes[point1:point2] ← parent1.genes[point1:point2]
6. Fill sisanya dengan urutan dari parent3 (tanpa duplikat)

Offspring 2: Kombinasi Parent2 dan Parent4

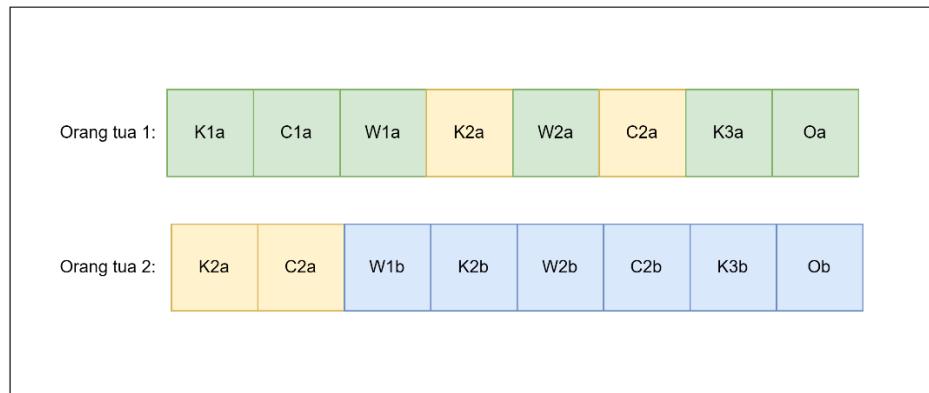
7. offspring2_genes ← [NULL] × size
8. offspring2_genes[point1:point2] ← parent2.genes[point1:point2]
9. Fill sisanya dengan urutan dari parent4 (tanpa duplikat)

10. RETURN (offspring1, offspring2)

Kode 4.6 Pseudocode *Crossover*

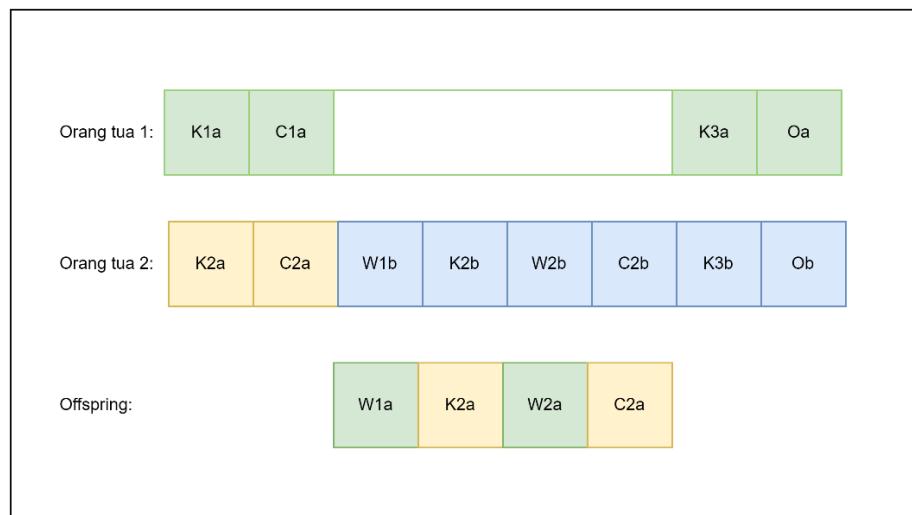
Langkah pertama yang dilakukan pada *crossover* adalah melakukan inisialisasi jumlah individu pada variabel *size* yang nilai *default*-nya adalah delapan karena orang tua yang digunakan adalah empat sehingga satu orang tua berisi dua individu yang didapatkan dari hasil seleksi. Kemudian, dalam *crossover* terdapat istilah *offspring* yang merupakan anak yang dihasilkan dari *crossover*. Dihasilkan dua *offspring* dalam setiap *crossover* pada HGA di tugas akhir ini. Berikut adalah proses dihasilkannya sebuah *offspring* pada HGA di tugas akhir ini.

- Pertama, digunakan dua orang tua sebagai individu utama sebagai penghasil *offspring*. Tahap ini ditunjukkan oleh gambar 4.2.



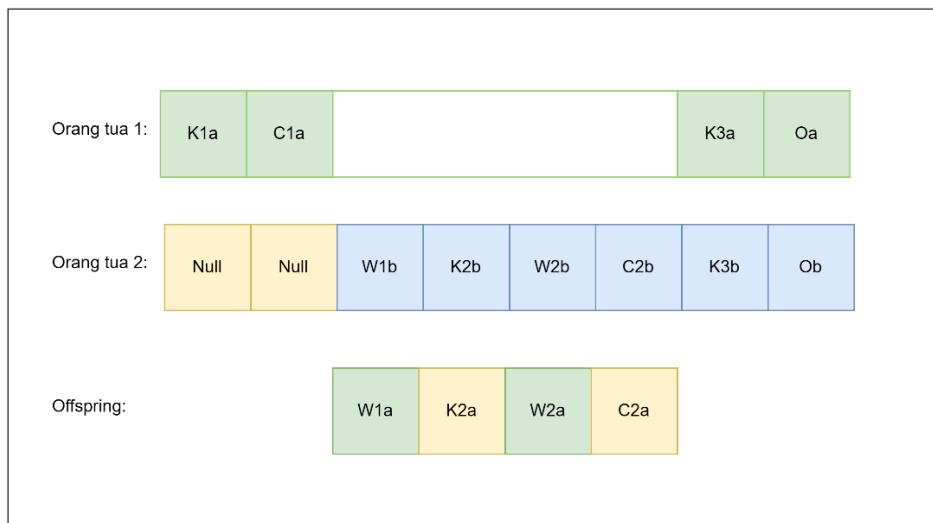
Gambar 4.2 Tahap Pertama dalam *Crossover*

- Kedua, sebagian gen dari orang tua 1 diambil secara acak untuk membentuk *offspring* baru, lalu id dari sebagian gen tersebut disimpan dalam sebuah variabel yang nantinya digunakan untuk menyaring gen-gen berikutnya pada *offspring* sehingga tidak ada gen yang duplikat. Tahap ini ditunjukkan oleh gambar 4.3.



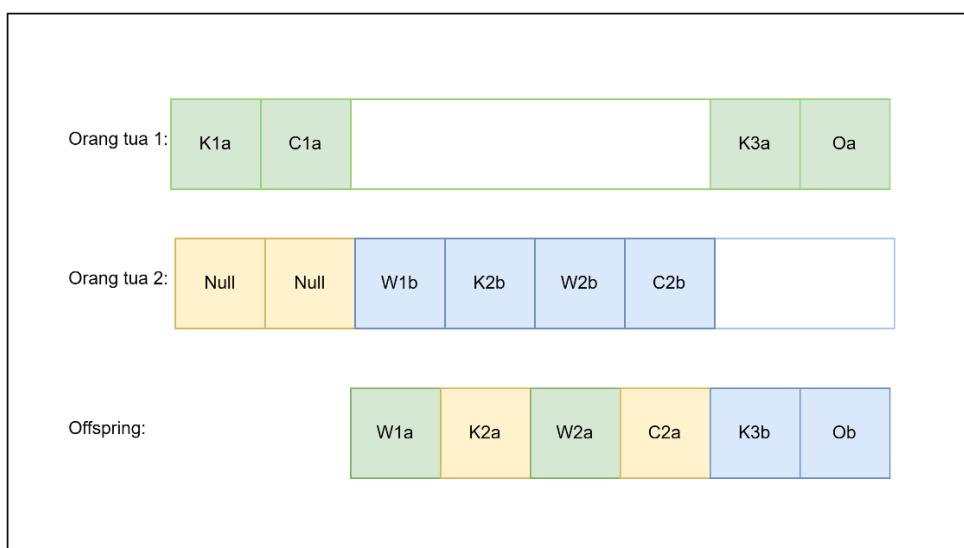
Gambar 4.3 Tahap Kedua dalam *Crossover*

- Ketiga, eliminasi gen-gen pada orang tua 2 yang duplikat dengan gen-gen pada *offspring* dengan mengganti nilainya dengan null. Pada gambar 4.3, dapat dilihat bahwa gen-gen yang duplikat adalah K2a dan C2a. Tahap ini ditunjukkan oleh gambar 4.4.



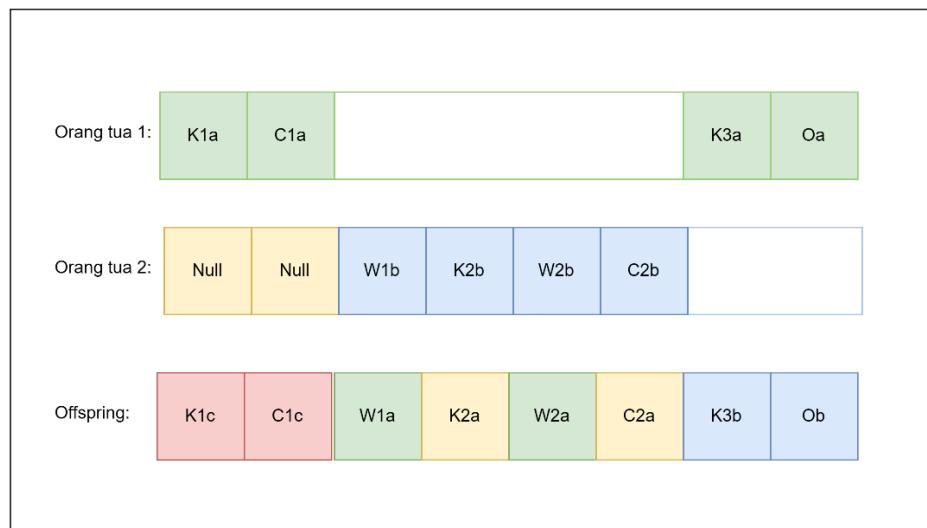
Gambar 4.4 Tahap Ketiga dalam *Crossover*

- d. Keempat, salin dan pindahkan sisa gen dari orang tua 2 yang urutannya belum ada di *offspring*, yakni, gen K3b dan Ob. Tahap ini ditunjukkan oleh gambar 4.5.



Gambar 4.5 Tahap Keempat dalam *Crossover*

- e. Kelima, karena pada kasus ini terdapat gen-gen yang duplikat antara orang tua 1 dan orang tua 2 sehingga dihasilkan *offspring* dengan gen yang masih kurang dalam sebuah kromosom, maka, gen dalam *offspring* diisi oleh gen-gen dari orang tua 3 sesuai dengan kategori yang dibutuhkan dalam *offspring*. Jika pada orang tua 3 masih ada juga yang duplikat dengan orang tua 1 atau 2, maka digunakan orang tua 4, dan jika masih ada yang duplikat, akan digunakan kembali gen-gen dari orang tua 1. Tahap ini ditunjukkan oleh gambar 4.6.



Gambar 4.6 Tahap Kelima dalam *Crossover*

Dengan demikian, metode *crossover* ini dapat memastikan setiap rute yang dihasilkan dari HGA lebih beragam dan terus konsisten sesuai dengan batasan-batasan yang telah ditetapkan.

4.3.6 Perancangan Operator Mutasi

Mutasi perlu dilakukan agar tidak terjadi konvergensi prematur setelah dilakukan *crossover*. Pada HGA di tugas akhir ini digunakan metode *swap mutation*. Metode ini menukar posisi dua gen secara acak untuk memberi variasi pada sebuah kromosom. *Swap mutation* cocok digunakan untuk sistem pada tugas akhir ini karena bisa disesuaikan agar gen-gen yang ditukar adalah gen dengan kategori yang sama sehingga urutan kategori destinasi masih sesuai dengan batasan-batasan yang telah ditetapkan. Kode 4.7 menunjukkan pseudocode dari metode mutasi pada HGA.

INPUT:

- chromosome: Kromosom yang akan dimutasi
- mutation_rate: Probabilitas mutasi (*default*: 0.05)

OUTPUT:

- mutated_chromosome: Kromosom hasil mutasi

LANGKAH-LANGKAH:

1. IF `random_float(0, 1) > mutation_rate` THEN:
RETURN `chromosome.copy()` // tidak terjadi mutasi
2. `mutated_genes ← chromosome.genes.copy()`
3. `size ← length(mutated_genes)`
4. `position1 ← random_int(0, size-1)`
5. `position2 ← random_int(0, size-1)`
6. WHILE `pos1 = pos2` DO:
`position2 ← random_int(0, size-1)` // Memastikan posisi yang berbeda
7. `SWAP(mutated_genes[position1], mutated_genes[position2])`
8. `mutated_chromosome ← Chromosome(mutated_genes, chromosome.start_point)`
9. RETURN `mutated_chromosome`

Kode 4.7 Pseudocode Mutasi

4.3.7 Perancangan Algoritma 2-Opt

Algoritma 2-Opt adalah digunakan sebagai pencarian lokal untuk memperbaiki atau meningkatkan kualitas solusi rute dengan menghilangkan rute yang bersilangan. Pseudocode dari algoritma 2-Opt dalam HGA pada tugas akhir ini ditunjukkan oleh kode 4.8.

INPUT:

- chromosome: Kromosom
- max_iterations: Iterasi Maksimum

OUTPUT:

- optimized_chromosome: Kromosom hasil optimasi

LANGKAH-LANGKAH:

1. Define category_positions:
 - makanan_berat: [0, 3, 6] // K1, K2, K3
 - makanan_ringan: [1, 5] // C1, C2
 - non_kuliner: [2, 4] // W1, W2
 - oleh_oleh: [7] // O
2. current_genes ← chromosome.genes.copy()
3. current_distance ← calculate_total_distance(current_genes)
4. improved ← TRUE
5. iteration ← 0
6. WHILE improved AND iteration < max_iterations DO:
 - 6.1. improved ← FALSE
 - 6.2. iteration ← iteration + 1

// Hanya swap dalam kategori yang sama

 - 6.3. FOR EACH category IN category_positions DO:
positions ← category_positions[category]
 - 6.3.1. FOR EACH pair (i, j) IN positions WHERE i ≠ j DO:
// Coba swap destinasi di posisi i dan j
new_genes ← current_genes.copy()
SWAP(new_genes[i], new_genes[j])

new_distance ← calculate_total_distance(new_genes)

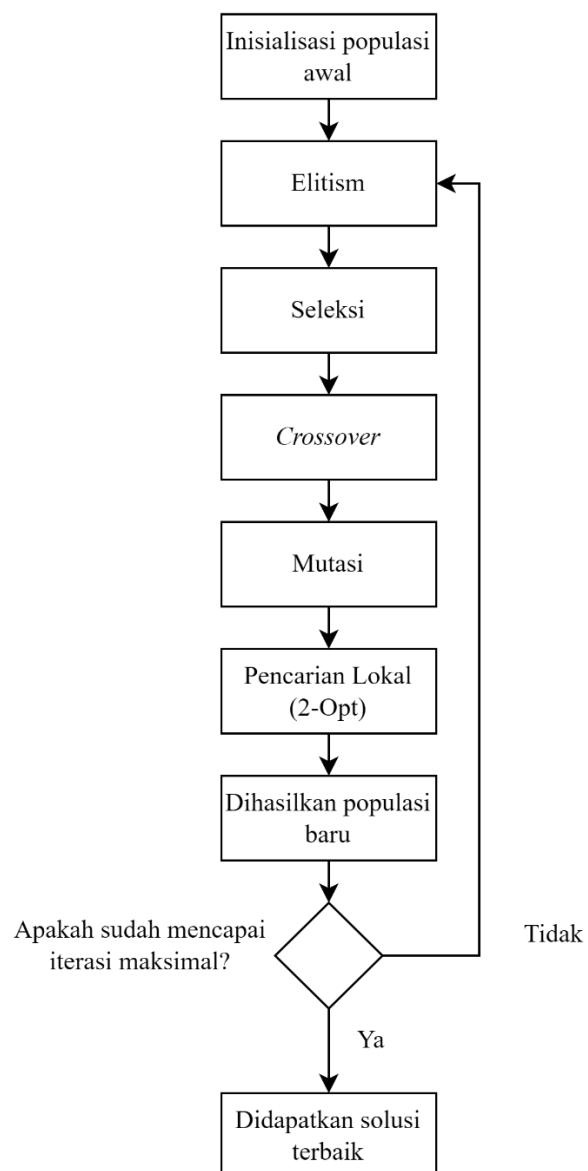
IF new_distance < current_distance THEN:
current_genes ← new_genes
current_distance ← new_distance
improved ← TRUE
7. RETURN Chromosome(current_genes, chromosome.start_point)

Kode 4.8 Pseudocode Algoritma 2-Opt

Algoritma 2-Opt pada HGA dalam tugas akhir ini telah disesuaikan agar bisa memenuhi batasan-batasan urutan destinasi pada rute wisata dengan cara menukar destinasi yang hanya memiliki kategori yang sama saja. Perbedaan utama antara 2-Opt dengan *swap mutation* adalah solusi 2-Opt hanya diterima jika terjadi peningkatan kualitas solusi saja, sedangkan *swap mutation* tetap menukar posisi destinasi pada rute tanpa memerhatikan kualitas solusi, hal ini dilakukan untuk menjaga keberagaman solusi.

4.3.8 Perancangan Proses Evolusi

Proses utama dalam GA adalah proses evolusi, proses ini mengulang tahapan-tahapan dalam GA, mulai dari inisialisasi solusi yang dilakukan sekali saja, kemudian pengulangan pada proses seleksi, *crossover*, mutasi, optimasi lokal dengan 2-Opt, hingga membentuk generasi baru dan kembali ke tahap seleksi hingga mencapai konvergensi sesuai dengan banyaknya iterasi yang dilakukan. Gambar 4.7 menunjukkan diagram alir proses evolusi dalam HGA di tugas akhir ini.



Gambar 4.7 Diagram Alir Proses Evolusi

Pemaparan teknis dari proses evolusi pada HGA dalam tugas akhir ini ditunjukkan oleh *pseudocode* pada kode 4.9.

INPUT:

- destinations: List semua destinasi
- start_point: Koordinat awal
- num_solutions: Jumlah solusi yang diinginkan (*default*: 3)
- config: Parameter konfigurasi HGA

OUTPUT:

- best_routes: List kromosom terbaik

LANGKAH-LANGKAH:

1. INISIALISASI PELACAKAN:

```
best_solution ← NULL  
best_fitness_history ← []  
average_fitness_history ← []
```

2. FOR route_number = 1 TO num_solutions DO:

2.1. INISIALISASI POPULASI:

```
population ← Initialize_Random_Population(  
    destinations, start_point, config.population_size)
```

2.2. EVALUASI FITNESS POPULASI AWAL:

```
FOR EACH chromosome IN population DO:  
    chromosome.calculate_fitness()  
population.sort_by_fitness()  
best_initial ← population.get_best_chromosome()
```

2.3. PENGULANGAN EVOLUSI:

```
FOR generation = 1 TO config.generations DO:
```

2.3.1. EVALUASI FITNESS:

```
FOR EACH chromosome IN population DO:  
    chromosome.calculate_fitness()  
population.sort_by_fitness()
```

2.3.2. LACAK SOLUSI TERBAIK:

```
current_best ← population.get_best_chromosome()  
best_fitness_history.append(current_best.get_fitness())  
average_fitness_history.append(population.get_average_fitness())
```

```
IF best_solution = NULL OR  
    current_best.get_fitness() > best_solution.get_fitness() THEN:  
    best_solution ← current_best.copy()
```

2.3.3. CEK KONVERGENSI

```
IF check_convergence(generation, best_fitness_history) THEN:  
    PRINT "Konvergensi tercapai pada generasi", generation  
    BREAK
```

2.3.4. BUAT GENERASI BARU:

```
new_population ← Create_New_Generation(population, config)  
population ← new_population
```

2.4. MASUKKAN KE DALAM HASIL:

```
best_routes.append(best_solution)
```

3. RETURN best_routes

Kode 4.9 Pseudocode Proses Evolusi

Salah satu tahapan penting dalam proses evolusi adalah pembuatan generasi baru yang di dalamnya terdapat proses seleksi, *crossover*, mutasi, dan optimasi lokal dengan 2-Opt. Kode 4.10 menunjukkan pseudocode dari tahap pembuatan generasi baru.

INPUT:

- population: Populasi saat ini
- config: Parameter konfigurasi

OUTPUT:

- new_population: Populasi generasi baru

LANGKAH-LANGKAH:

1. new_chromosomes ← []

2. ELITISM (Pertahankan solusi terbaik):

```
elite_chromosomes ← population.get_best_n_chromosomes(config.elitism_count)  
FOR EACH elite IN elite_chromosomes DO:  
    new_chromosomes.append(elite.copy())
```

3. HASILKAN OFFSPRING:

```
WHILE length(new_chromosomes) < config.population_size DO:
```

3.1. SELEKSI (*TOURNAMENT*):

```
parent1 ← Tournament_Selection(population, config.tournament_size)  
parent2 ← Tournament_Selection(population, config.tournament_size)  
parent3 ← Tournament_Selection(population, config.tournament_size)  
parent4 ← Tournament_Selection(population, config.tournament_size)
```

3.2. *CROSSOVER*:

```
IF random_float(0, 1) < config.crossover_rate THEN:  
    offspring1, offspring2 ← Order_Crossover (  
        parent1, parent2, parent3, parent4)
```

ELSE:

```
    offspring1 ← parent1.copy()  
    offspring2 ← parent2.copy()
```

3.3. MUTATION:

```
offspring1 ← Swap_Mutation(offspring1, config.mutation_rate)  
offspring2 ← Swap_Mutation(offspring2, config.mutation_rate)
```

3.4. LOCAL SEARCH (2-Opt):

```
IF config.use_2opt THEN:  
    offspring1 ← Two_Opt_Optimize_With_Constraints(  
        offspring1, config.two_opt_iterations)  
    offspring2 ← Two_Opt_Optimize_With_Constraints(  
        offspring2, config.two_opt_iterations)
```

3.5. ADD TO NEW POPULATION:

```
new_chromosomes.append(offspring1)  
IF length(new_chromosomes) < config.population_size THEN:  
    new_chromosomes.append(offspring2)
```

4. POTONG JIKA LEBIH DARI JUMLAH POPULASI:

```
new_chromosomes ← new_chromosomes[0:config.population_size]
```

5. RETURN Population(new_chromosomes, config.population_size)

Kode 4.10 *Pseudocode* Pembuatan Generasi Baru

Setelah proses evolusi selesai dan telah mencapai konvergensi sesuai dengan iterasi yang ditentukan, solusi akhir akan diambil dari kromosom yang memiliki nilai *fitness* terbaik dari seluruh generasi yang dihasilkan dari HGA. Pada tugas akhir ini akan dihasilkan tiga rute wisata untuk direkomendasikan ke pengguna. Oleh karena itu, HGA akan dijalankan sebanyak tiga kali untuk menghasilkan tiga rute wisata dengan waktu ekskusif untuk menghasilkan tiga rute wisata tersebut harus kurang dari satu menit.

4.4 Perancangan Uji Coba Sistem

Uji coba dilakukan untuk memastikan hasil rekomendasi rute yang diberikan mengalami peningkatan nilai *fitness* dari rute hasil inisiasi awal. Uji coba dilakukan dengan tujuan untuk memvalidasi bahwa *Hybrid Genetic Algorithm* (HGA) yang dikembangkan mampu menghasilkan solusi rute wisata halal di Surabaya yang optimal dengan memenuhi batasan-batasan yang telah ditetapkan pada bab 3. Uji coba yang akan dilakukan pada tugas akhir ini adalah uji fungsional dan uji performa algoritma.

4.4.1 Perancangan Uji Fungsional

Uji fungsional dilakukan untuk melakukan validasi fungsionalitas sistem, memastikan bahwa sistem bekerja sesuai dengan batasan-batasan yang telah ditentukan. Terdapat dua hal yang akan dilakukan dalam uji fungsional yakni, validasi batasan-batasan dan validasi rute dan jarak.

4.4.1.1 Perancangan Validasi Batasan-Batasan

Validasi batasan-batasan dilakukan dengan cara mengecek hasil rekomendasi rute. Hasil rekomendasi rute yang dicek harus mengikuti batasan-batasan yang telah ditetapkan sebelumnya. Dari batasan-batasan yang telah ditetapkan sebelumnya, hasil rekomendasi rute wisata yang dihasilkan harus mengikuti tiga aturan berikut.

- Rute yang dihasilkan harus berisi delapan destinasi wisata.
- Tidak ada destinasi yang muncul dua kali (duplikat) dalam satu rute.
- Setiap urutan destinasi harus sesuai urutan kategori yang telah ditetapkan pada batasan-batasan di bab 3.

Validasi batasan-batasan dilakukan dengan mengecek secara langsung hasil rekomendasi rute apakah sudah sesui dengan ketiga aturan yang telah disebutkan sebelumnya.

4.4.2 Perancangan Uji Performa Algoritma

Uji performa algoritma dilakukan untuk menilai kemampuan *Hybrid Genetic Algorithm* (HGA) dalam menghasilkan rute dengan jarak tempuh minimum sekaligus menjaga waktu komputasi tetap layak untuk digunakan pada *website* sistem rekomendasi rute wisata halal ini. Dalam uji performa algoritma ini kualitas solusi dan parameter dari HGA akan diuji. Uji kualitas solusi dan parameter HGA dilakukan untuk mengukur kualitas solusi dari HGA dalam menghasilkan rekomendasi rute dengan jarak yang mendekati optimal. Berikut adalah parameter yang akan diuji.

a. Populasi

Pengujian dengan parameter populasi dilakukan untuk melihat apakah populasi yang semakin besar memberikan solusi yang lebih baik. Pengujian dilakukan dengan menjalankan HGA dengan ukuran populasi secara berturut-turut 100, 300, 500, dan 700. Dari percobaan tersebut akan ditentukan populasi mana yang lebih optimal dengan mempertimbangkan *trade-off* waktu eksekusi.

b. Generasi

Pengujian dengan parameter generasi dilakukan untuk mengetahui pada generasi berapa algoritma dapat konvergen. Generasi yang konvergen berarti bahwa algoritma sudah tidak menghasilkan solusi yang lebih optimal. Konvergennya suatu generasi pada HGA menandakan bahwa algoritma sudah dapat dihentikan karena telah mendapatkan solusi akhir. Pengujian dilakukan dengan menjalankan HGA dengan ukuran generasi secara berturut-turut 20, 40, dan 80.

c. *Crossover Rate*

Pengujian dengan parameter *crossover rate* dilakukan untuk menentukan *crossover rate* mana yang dapat menghasilkan hasil rekomendasi rute yang lebih optimal. Pengujian dilakukan dengan menjalankan HGA dengan *crossover rate* berturut-turut sebesar 0.6, 0.7, 0.8, dan 0.9.

d. *Mutation Rate*

Pengujian dengan parameter *mutation rate* dilakukan untuk menentukan berapa *mutation rate* yang dibutuhkan algoritma untuk menghasilkan solusi yang lebih optimal. Pengujian dilakukan dengan menjalankan HGA dengan *mutation rate* berturut-turut sebesar 0.01, 0.05, 0.2, dan 0.5.

e. 2-Opt

2-Opt merupakan algoritma pencarian lokal yang menjadi bagian dalam HGA ini. Penggunaan 2-Opt akan diukur untuk melihat apakah adanya 2-Opt membantu HGA dalam menghasilkan solusi yang lebih optimal. Pengujian dilakukan dengan menjalankan HGA dengan 2-Opt dan tanpa 2-Opt.

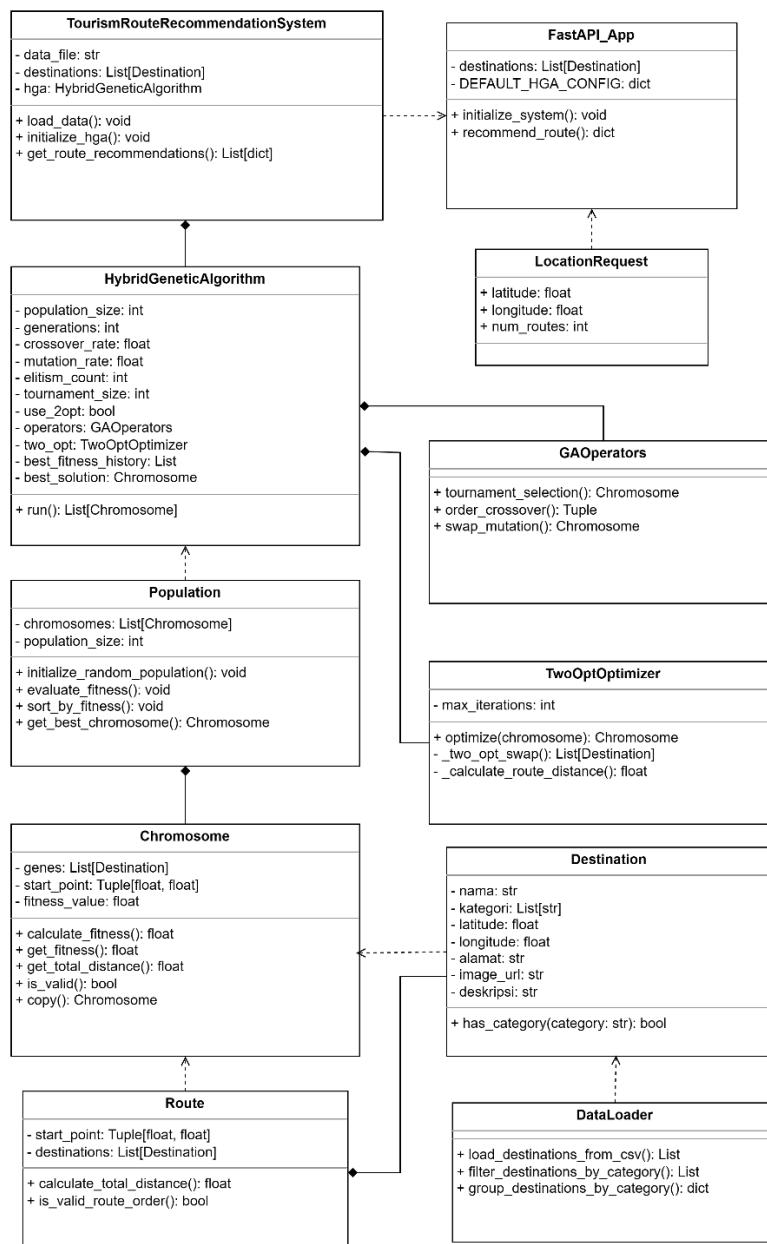
Parameter-parameter tersebut akan diuji dengan menggunakan skrip *python* untuk menjalankan HGA dengan kombinasi parameter-parameter tersebut. Metrik utama yang dilihat dalam pengujian ini adalah jarak terbaik dari suatu generasi pada HGA dan waktu eksekusinya.

BAB 5 Implementasi

Setelah dilakukan perancangan, selanjutnya pada bab 5 dijelaskan implementasi dari sistem rekomendasi rute wisata halal di Surabaya dengan HGA sesuai dengan rancangan pada bab 4. Penjelasan pada bab ini mencakup implementasi perancangan melalui pengembangan algoritma sistem rekomendasi rute dan integrasi algoritma sistem rekomendasi rute dengan *website*.

5.1 Pengembangan Algoritma Sistem Rekomendasi Rute

Algoritma sistem rekomendasi rute dikembangkan menggunakan bahasa Python dengan paradigma *Object-Oriented Programming* (OOP). Oleh karena itu, kelas, atribut, dan fungsi yang digunakan dalam algoritma ini dapat divisualisasikan melalui diagram kelas yang ditunjukkan oleh gambar 5.1.



Gambar 5.1 Diagram Kelas Algoritma Sistem Rekomendasi Rute

5.2 Integrasi Algoritma Sistem Rekomendasi Rute dengan *Website*

BAB 6 Hasil dan Pembahasan

Hasil implementasi website sistem rekomendasi rute wisata, dan pengujian

DAFTAR PUSTAKA

- Ahmed, Z. H., Altukhaim, F. F., Saudagar, A. K., & Khan, S. (2024). Optimization of the Travelling Salesman Problem Using a New Hybrid Genetic Algorithm. *International Journal of Computer Science and Network Security*, 12-22.
- Akpan, I. J., Kobora, Y. M., Owolabi, J., Akpan, A. A., & Offodile, O. F. (2025). Conversational and Generative Artificial Intelligence and Human-Chatbot Interaction in Education and Research. *International Transactions In Operational Research*, 1251-1281.
- Anshari, M. R., & Baizal, Z. K. (2023). N-Days Tourist Route Recommender System in Yogyakarta Using Genetic Algorithm Method. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 736-743.
- Arifin, R. R. (2024). *Pengembangan Sistem Rekomendasi Rute Wisata Kuliner Halal Surabaya dengan Metode Genetic Algorithm*. Surabaya: Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
- Bappedalitbang Kota Surabaya. (2025, Maret 7). *PDRB Surabaya 2024, Potret Pergerakan Ekonomi Surabaya*. Retrieved from Bappedalitbang Kota Surabaya: <https://bappedalitbang.surabaya.go.id/berita/2025-03-07/pdrb-surabaya-2024-potret-pergerakan-ekonomi-surabaya>
- Dhouib, S. (2021). Optimization of Travelling Salesman Problem on Single Valued Triangular Neutrosophic Number using Dhouib-Matrix-TSP1 Heuristic. *International Journal of Engineering*, 2642-2647.
- Edwardo, D. M. (2023). *Pengembangan Sistem Informasi Rekomendasi Pariwisata di Kota Surabaya Menggunakan Algoritme Genetika*. Surabaya: Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
- Google AI for Developers. (2025, Mei 22). *Gemini Developer API*. Retrieved from Gemini AI For Developers: <https://ai.google.dev/gemini-api/docs>
- Hossain, M. S., Choudhury, S. S., Hayat, S. M., Tanim, A. S., Kabir, M. N., & Islam, M. M. (2019). An Efficient Solution to Travelling Salesman Problem using Genetic Algorithm with a Modified Crossover Operation. *EMITTER International Journal of Engineering Technology*, 480-493.
- Kurniawan, M. R. (2024). *Pengembangan Desain Antarmuka Pengguna dan Implementasi Front End Aplikasi Chatbot Tanya.In Menggunakan Pendekatan User-Centered Design*. Surabaya: Departemen Teknik Informatika Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember.
- Mahmoudinazlou, S., & Kwon, C. (2024). A Hybrid Genetic Algorithm for The Min-Max Multiple Travelling Salesman Problem. *Computers and Operations Research*, 106455.
- Nasution, S. M., Husni, E., Kuspriyanto, Yusuf, R., & Yahya, B. N. (2021). Contextual Route Recommendation System in Heterogeneous Traffic Flow. *Sustainability*, 13191.

- Noviantoro, K. M., & Zurohman, A. (2020). Prospek Pariwisata Syariah (Halal Tourism): Sebuah Tantangan di Era Revolusi Industri 4.0. *EQUILIBRIUM: Jurnal Ekonomi Syariah*, 275-296.
- Nozari, H., Szmelter-Jarosz, A., & Irani, H. R. (2025). Designing an Ambulance Routing Optimization Model Using the Combination of Machine Learning and Genetic Algorithm in Conditions of Uncertainty. *Systems and Soft Computing*, 200276.
- Pak, E. (2025, Juni 19). *Global Muslim Travel Index Spotlights Southeast Asia's Growth in Halal Tourism*. Retrieved from Branding in Asia: <https://www.brandinginasia.com/global-muslim-travel-index-spotlights-southeast-asias-growth-in-halal-tourism/>
- Pemerintah Kota Surabaya. (2024, Januari 25). *Sepanjang Tahun 2023, Wisatawan yang Berkunjung ke Surabaya Tembus 17,4 Juta*. Retrieved from Pemerintah Kota Surabaya: <https://www.surabaya.go.id/id/berita/78329/sepanjang-tahun-2023-wisatawan-yang-berkunjung-ke-surabaya-tembus-174-juta>
- Pemerintah Pusat Indonesia. (2025, Februari 10). *Peraturan Presiden (Perpres) Nomor 12 Tahun 2025 tentang Rencana Pembangunan Jangka Menengah Nasional Tahun 2025 - 2029*. Retrieved from Badan Pembinaan dan Pengembangan Hukum Pemeriksaan Keuangan Negara: <https://peraturan.bpk.go.id/Details/314638/perpres-no-12-tahun-2025>
- Rakhmawati, N. A., Fatawi, J., Najib, A. C., & Firmansyah, A. A. (2021). Linked Open Data for Halal Food Products. *Journal of King Saud University - Computer and Information Sciences*, 728-739.
- Soares, G., Bulhões, T., & Bruck, B. (2024). An efficient Hybrid Genetic Algorithm for The Travelling Salesman Problem with Release Dates. *European Journal of Operational Research*, 31-42.
- Stoilova, E. (2021). AI Chatbots as A Customer Service and Support Tool. *ROBONOMICS The Journal of the Automated Economy*, 21.
- Suanpang, P., Jamjuntr, P., Jermisittiparsert, K., & Kaewyong, P. (2022). Tourism Service Scheduling in Smart City Based on Hybrid Genetic Simulated Annealing Algorithm. *Sustainability*, 16293.
- Wahab, M. N., Nazir, A., Khalil, A., Ho, W. J., Akbar, M. F., Noor, M. H., & Mohamed, A. S. (2024). Improved Genetic Algorithm for Mobile Path Planning in Static Environments. *Expert Systems With Applications*, 123762.

LAMPIRAN