

OPTIMISASI URUTAN KUNJUNGAN WAHANA BERBASIS MODEL *TIME-DEPENDENT TRAVELING SALESMAN* *PROBLEM* MENGGUNAKAN ALGORITMA GENETIKA ADAPTIF

SHAFA AQILAH FAHDAH, KHUSNUL NOVIANINGSIH*, DAN SOFIHARA AL HAZMY

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas
Pendidikan Indonesia, Bandung, Indonesia

*Penulis Korespondensi: k_novianingsih@upi.edu

ABSTRAK

Time-dependent traveling salesman problem (TD-TSP) merupakan pengembangan dari *traveling salesman problem* (TSP) yang mempertimbangkan kondisi perjalanan yang bergantung pada waktu. Penelitian ini membahas optimisasi urutan kunjungan wahana pada sebuah taman hiburan di Jakarta dengan mempertimbangkan waktu tempuh antarwahana dan waktu tunggu antrean yang bergantung pada waktu kedatangan pengunjung. Model TD-TSP berbasis waktu tunggu antrean diintegrasikan dengan Algoritma Genetika Adaptif (AGA) yang menyesuaikan probabilitas crossover dan mutasi berdasarkan nilai fitness populasi. Integrasi tersebut diterapkan untuk meminimumkan total waktu kunjungan sekaligus mengevaluasi pengaruh mekanisme adaptif terhadap kualitas dan kestabilan solusi. Kinerja AGA dibandingkan dengan Algoritma Genetika (AG) berparameter tetap. Hasil penelitian menunjukkan bahwa AGA menghasilkan solusi terbaik dengan total waktu kunjungan sebesar 193,53 menit, rata-rata 194,98 menit, dan standar deviasi 0,93 menit. Sementara itu, AG menghasilkan rata-rata total waktu 195,43 menit dengan standar deviasi 1,01 menit. Meskipun AGA menghasilkan rata-rata total waktu dan standar deviasi yang lebih rendah, perbedaan rata-rata kedua algoritma tidak signifikan secara statistik berdasarkan Welch's t-test. Analisis sensitivitas menunjukkan bahwa perubahan waktu mulai kunjungan memengaruhi total waktu kunjungan melalui perubahan waktu tunggu antrean.

Kata Kunci: Algoritma Genetika Adaptif, Optimisasi Rute, Taman Hiburan, *Time-Dependent Traveling Salesman Problem*, Waktu Tunggu Antrean.

ABSTRACT

The time-dependent traveling salesman problem (TD-TSP) extends the traveling salesman problem (TSP) by accounting for time-dependent conditions. This study optimizes attraction visit sequences at an amusement park in Jakarta by considering travel times between attractions and queue waiting times that depend on visitors' arrival times. A queue-based TD-TSP model is integrated with an Adaptive Genetic Algorithm (AGA), which dynamically adjusts crossover and mutation probabilities based on population fitness values. This integration aims to minimize total visit time and evaluate the adaptive mechanism's effect on solution quality and stability. The performance of AGA is compared with that of a conventional Genetic Algorithm (GA) using fixed crossover and mutation probabilities. The results show that AGA achieved the best solution with a total visit time of 193.53 minutes, a mean of 194.98 minutes, and a standard deviation of 0.93 minutes. Meanwhile, GA produced a mean total visit time of 195.43 minutes with a standard deviation of 1.01 minutes. Although AGA achieved a lower mean total visit time and standard deviation, the difference in mean performance was not statistically

significant according to Welch's t-test. Sensitivity analysis shows that changes in the starting time affect the total visit time through variations in queue waiting times.

Keywords: *Adaptive Genetic Algorithm, Amusement Park, Optimal Route, Time-Dependent Traveling Salesman Problem, Waiting Time.*

1. Pendahuluan

Traveling Salesman Problem (TSP) merupakan salah satu permasalahan optimisasi kombinatorial yang banyak diterapkan dalam berbagai bidang, seperti logistik, transportasi, distribusi, dan perencanaan perjalanan wisata. Permasalahan TSP berfokus pada penentuan rute bagi seorang agen perjalanan untuk mengunjungi sejumlah lokasi tepat satu kali sebelum kembali ke lokasi awal dengan meminimumkan jarak atau biaya perjalanan [1]. Namun, model TSP klasik mengasumsikan bahwa biaya perjalanan antar lokasi bersifat tetap sehingga belum dapat merepresentasikan permasalahan yang dipengaruhi oleh perubahan kondisi terhadap waktu.

Salah satu pengembangan TSP yang mempertimbangkan dinamika tersebut adalah *Time-dependent traveling salesman problem* (TD-TSP). TD-TSP merupakan generalisasi dari TSP klasik dengan biaya perjalanan yang bergantung pada posisi lokasi dalam urutan perjalanan dan kondisi waktu [2]. Model ini lebih sesuai untuk permasalahan yang dipengaruhi oleh perubahan kondisi terhadap waktu, seperti kemacetan lalu lintas, distribusi barang, maupun perencanaan rute kunjungan tempat wisata yang dipengaruhi oleh kepadatan pengunjung dan waktu tunggu antrean. Dengan mempertimbangkan komponen yang bergantung pada waktu, TD-TSP dapat menghasilkan solusi rute yang lebih realistis dibandingkan model TSP statis. Salah satu pendekatan eksak untuk permasalahan *Time-dependent traveling salesman problem with Time Windows* (TD-TSPTW) dikembangkan melalui *Dynamic Discretization Discovery* [3].

Algoritma Genetika (AG) merupakan salah satu metode metaheuristik berbasis evolusi yang dikembangkan oleh Holland (1975) dan terinspirasi dari mekanisme seleksi alam. AG melakukan pencarian solusi melalui evolusi populasi dengan menggunakan operator seleksi, *crossover*, dan mutasi, sedangkan kualitas setiap individu dievaluasi menggunakan fungsi *fitness*. Algoritma ini telah diterapkan pada berbagai permasalahan optimasi rute karena mampu melakukan pencarian pada ruang solusi yang kompleks. Suprayogi dan Mahmudy menerapkan AG pada *traveling salesman problem with Time Window* (TSP-TW) untuk optimasi rute antar-jemput laundry dengan mempertimbangkan kendala waktu pelayanan [4]. Pada permasalahan TD-TSP, fungsi *fitness* dapat disesuaikan untuk mempertimbangkan komponen waktu yang berubah berdasarkan waktu kedatangan sehingga solusi yang diperoleh dapat merepresentasikan kondisi dinamis. Pendekatan evolusioner juga telah diterapkan untuk menentukan urutan kunjungan wahana dengan mempertimbangkan waktu tunggu yang berubah sesuai waktu kedatangan pengunjung [5].

Meskipun AG memiliki kemampuan yang baik dalam menyelesaikan permasalahan optimasi, kinerjanya dipengaruhi oleh pemilihan parameter seperti probabilitas *crossover* dan mutasi. Penggunaan parameter yang tetap dapat membatasi keseimbangan antara eksplorasi dan eksploitasi selama proses pencarian solusi. Kondisi tersebut dapat meningkatkan risiko konvergensi prematur atau menyebabkan pencarian solusi kurang efektif pada permasalahan yang kompleks. Salah satu pendekatan untuk mengatasi permasalahan tersebut adalah Algoritma Genetika Adaptif (AGA), yaitu pengembangan AG yang melakukan penyesuaian parameter secara dinamis selama proses evolusi. Penyesuaian probabilitas *crossover* dan mutasi berdasarkan kondisi populasi atau nilai *fitness* individu memungkinkan mekanisme pencarian beradaptasi terhadap kondisi solusi pada setiap generasi. Beberapa penelitian menunjukkan bahwa mekanisme adaptif pada parameter AG dapat meningkatkan kualitas solusi dibandingkan penggunaan parameter tetap [6], [7]. Mekanisme mutasi adaptif juga dapat membantu mempertahankan keragaman populasi dan meningkatkan kemampuan algoritma

dalam menemukan solusi yang lebih baik [8]. Pengembangan AG melalui mekanisme adaptasi parameter juga berpotensi meningkatkan kualitas solusi dan mengurangi risiko konvergensi prematur [9].

Permasalahan penentuan urutan kunjungan wahana memiliki karakteristik yang berbeda dari TSP klasik. Pada penelitian ini, sejumlah wahana telah ditetapkan untuk dikunjungi dengan mempertimbangkan waktu tempuh antarwahana dan waktu tunggu antrean. Waktu tunggu antrean berubah berdasarkan waktu kedatangan pengunjung sehingga urutan kunjungan dengan total waktu tempuh terpendek belum tentu menghasilkan total waktu kunjungan minimum. Oleh karena itu, pendekatan TD-TSP digunakan untuk menentukan urutan kunjungan yang meminimumkan total waktu dengan mempertimbangkan perubahan waktu tunggu antrean.

Berbagai pendekatan telah dikembangkan untuk menyelesaikan TSP dengan komponen waktu yang dinamis. Cacchiani dkk. [10] membahas TSP dengan waktu pelayanan yang bergantung pada waktu, sedangkan Vu dkk. [11] menggunakan *Dynamic Discretization Discovery* untuk menyelesaikan permasalahan rute dengan waktu perjalanan dinamis. Lera-Romero dkk. [12] menerapkan *dynamic programming*, sementara Fontaine dkk. [13] mengembangkan pendekatan berbasis A* dan *local search* untuk TD-TSP dengan *time windows* [13]. Chen dkk. [14] menggunakan *deep reinforcement learning* untuk menangani waktu perjalanan yang dinamis dan stokastik. Yang dkk. [15] menerapkan *Partheno-Genetic Algorithm* untuk mengoptimalkan kunjungan wahana dengan mempertimbangkan waktu perjalanan dan antrean. Penelitian-penelitian tersebut menunjukkan beragam pendekatan penyelesaian TD-TSP, tetapi memiliki perbedaan dalam komponen waktu, metode, dan tujuan optimisasi.

Berdasarkan kajian penelitian terdahulu, TD-TSP telah dikembangkan untuk menangani perubahan waktu perjalanan, waktu pelayanan, dan waktu tunggu antrean. Algoritma Genetika Adaptif juga telah diterapkan untuk meningkatkan kualitas solusi TSP melalui penyesuaian probabilitas crossover dan mutasi. Namun, berdasarkan literatur yang dikaji, evaluasi penerapan Algoritma Genetika Adaptif untuk menentukan urutan kunjungan wahana dengan waktu tunggu antrean yang bergantung pada waktu kedatangan belum pernah diteliti. Oleh karena itu, diperlukan kajian mengenai kinerja mekanisme adaptif pada permasalahan tersebut, khususnya melalui perbandingan dengan Algoritma Genetika berparameter tetap.

Penelitian ini bertujuan menerapkan Algoritma Genetika Adaptif untuk menentukan urutan kunjungan wahana pada sebuah taman hiburan di Jakarta yang dapat meminimumkan total waktu kunjungan. Kontribusi penelitian ini terletak pada evaluasi penerapan probabilitas crossover dan mutasi adaptif pada TD-TSP dengan waktu tunggu antrean yang bergantung pada waktu kedatangan. Kinerja AGA akan dievaluasi dengan cara membandingkan solusinya dengan AG berparameter tetap. Selain itu, analisis sensitivitas akan dilakukan untuk mengevaluasi pengaruh waktu mulai kunjungan terhadap total waktu kunjungan.

2. Metode Penelitian

Penelitian ini menerapkan model TD-TSP pada permasalahan penentuan urutan kunjungan wahana di taman hiburan dan menyelesaikannya menggunakan Algoritma Genetika Adaptif (AGA). Permasalahan dimodelkan dengan mempertimbangkan waktu tempuh antarwahana yang bersifat tetap dan waktu tunggu antrean yang bergantung pada waktu kedatangan pengunjung. Model yang telah dibangun kemudian diselesaikan menggunakan AGA melalui tahapan representasi permutation encoding, *Order crossover* (OX), *swap mutation*, dan penyesuaian probabilitas *crossover* serta mutasi secara adaptif. Kinerja AGA dalam menyelesaikan TD-TSP diukur dengan cara membandingkan solusi optimal yang diperoleh AGA dan GA.

2.1. Data dan Deskripsi Permasalahan

Penelitian ini menerapkan model TD-TSP pada permasalahan kunjungan sejumlah wahana pada suatu taman hiburan. Diberikan daftar wahana yang harus dikunjungi oleh pengunjung. Pengunjung memulai perjalanan dari pintu masuk utama, mengunjungi setiap wahana tepat satu kali, kemudian kembali ke pintu masuk. Oleh karena itu, solusi yang dicari berupa suatu rute yang menentukan urutan kunjungan seluruh wahana dengan total waktu kunjungan minimum. Total waktu kunjungan terdiri atas waktu tempuh antarwahana dan waktu tunggu antrean pada wahana. Waktu tempuh antarwahana diasumsikan tetap, sedangkan waktu tunggu bersifat *time-dependent*, yaitu bergantung pada waktu kedatangan pengunjung di wahana tersebut.

Data yang digunakan dalam penelitian ini terdiri atas data lokasi wahana, waktu tempuh antarwahana, dan waktu tunggu antrean pada wahana. Objek penelitian berupa 26 wahana pada sebuah taman hiburan di Jakarta dengan pintu masuk sebagai titik awal dan titik akhir perjalanan. Setiap wahana direpresentasikan sebagai sebuah node, sedangkan pintu masuk digunakan sebagai node awal dan akhir rute.

Data lokasi wahana berupa koordinat geografis yang digunakan untuk menentukan jarak antarwahana. Jarak tersebut kemudian dikonversi menjadi waktu tempuh dalam satuan menit. Waktu tempuh antarwahana dianggap tetap selama proses optimasi. Sementara itu, waktu tunggu antrean merupakan komponen yang bergantung pada waktu dan dinyatakan berdasarkan interval waktu tertentu. Pengelompokan tersebut digunakan untuk membentuk fungsi waktu tunggu yang menentukan waktu tunggu efektif suatu wahana berdasarkan waktu kedatangan pengunjung. Dengan demikian, waktu tunggu yang digunakan pada suatu wahana ditentukan berdasarkan waktu kedatangan pengunjung pada wahana tersebut.

2.2. Model Optimisasi TD-TSP

Model optimisasi TD-TSP dalam penelitian ini dibangun berdasarkan beberapa asumsi, yaitu:

1. pengunjung memulai dan mengakhiri perjalanan di pintu masuk taman hiburan;
2. setiap wahana dikunjungi tepat satu kali;
3. total waktu kunjung wahana dihitung sebagai penjumlahan waktu tempuh antarwahana dengan waktu tunggu antrean setiap wahana;
4. waktu tempuh antarwahana tidak bergantung pada waktu kedatangan pengunjung dan dihitung berdasarkan jarak antarwahana dan kecepatan berjalan pengunjung;
5. waktu tunggu antrean pada setiap wahana bergantung pada waktu kedatangan pengunjung; dan
6. waktu istirahat pengunjung tidak diperhitungkan;
7. durasi menikmati wahana tidak diperhitungkan secara eksplisit dalam model. Optimisasi difokuskan hanya pada waktu tempuh antarwahana dan waktu tunggu antrean. Penyederhanaan ini dilakukan karena data waktu yang digunakan dalam penelitian terbatas pada waktu tempuh antarwahana dan waktu tunggu antrean. Oleh karena itu, hasil optimisasi merepresentasikan waktu minimum berdasarkan kedua komponen tersebut, bukan menyatakan total waktu aktual seluruh aktivitas pengunjung.

Misalkan N adalah himpunan wahana yang harus dikunjungi. Dimisalkan pula bahwa indeks 0 menyatakan pintu masuk taman hiburan. Maka, $N_0 = N \cup \{0\}$ merupakan himpunan lokasi yang harus dikunjungi pada taman hiburan tersebut. Misalkan t_{ij} dan a_i masing-masing adalah waktu tempuh dari lokasi i ke j dan waktu tiba di lokasi i . Misalkan $w_j(a_j)$ adalah waktu tunggu antrean di lokasi j pada waktu kedatangan a_j .

Model optimisasi TD-TSP dibangun untuk menentukan urutan kunjungan wahana dengan total waktu kunjungan terkecil. Model optimisasi ini terdiri dari tiga komponen utama, yaitu variabel keputusan, fungsi tujuan, dan fungsi kendala (*constraints*). Variabel keputusan ditentukan nilainya melalui proses optimisasi. Fungsi tujuan digunakan untuk meminimumkan

total waktu kunjung wahana, sedangkan fungsi kendala terdiri dari sejumlah persamaan yang menyatakan kondisi atau aturan yang harus dipenuhi oleh solusi optimal. Karena tujuan optimisasi adalah untuk menentukan urutan kunjungan wahana dengan total waktu kunjungan minimum, maka variabel keputusan model optimisasi didefinisikan sebagai variabel biner x_{ij} yang bernilai 1 jika lokasi j dikunjungi setelah lokasi i , dan bernilai 0 untuk yang lainnya.

Dengan demikian, fungsi tujuan model optimisasi diformulasikan sebagai berikut:

Meminimumkan:

$$Z = \sum_{i \in N_0} \sum_{j \in N_0} t_{ij} x_{ij} + \sum_{j \in N} w_j(a_j) \quad (1)$$

Berikut adalah kendala dari model optimisasi TD-TSP:

1. Setiap wahana tepat dikunjungi satu kali:

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} = 1, \forall j \in N \quad (2)$$

2. Setiap wahana memiliki tepat satu lokasi tujuan berikutnya:

$$\sum_{\substack{j \in N_0 \\ j \neq i}} x_{ij} = 1, \forall i \in N \quad (3)$$

3. Perjalanan dimulai dari pintu masuk dan kembali lagi ke pintu masuk setelah seluruh wahana dikunjungi:

$$\sum_{j \in N} x_{0j} = 1 \quad (4)$$

$$\sum_{i \in N} x_{i0} = 1 \quad (5)$$

4. Waktu kedatangan pada suatu lokasi bergantung pada urutan kunjungan:

$$a_j \geq a_i + t_{ij} + w_i(a_i) - M(1 - x_{ij}), \quad \forall i \neq j, \forall i \in N_0, \forall j \in N \quad (6)$$

di mana M adalah bilangan positif yang cukup besar (konstanta Big- M). Persamaan (6) menyatakan bahwa jika wahana j dikunjungi langsung setelah wahana i ($x_{ij} = 1$), maka waktu kedatangan di wahana j paling sedikit harus sama dengan waktu kedatangan di wahana i ditambah waktu tunggu antrean dan waktu tempuh menuju wahana j . Sebaliknya, jika wahana j tidak dikunjungi langsung setelah wahana i ($x_{ij} = 0$), maka Persamaan (6) dapat diabaikan. Penambahan konstanta Big- M pada persamaan tersebut digunakan untuk menjamin bahwa Persamaan (6) selalu terpenuhi ketika $x_{ij} = 0$.

5. Seluruh variabel keputusan x_{ij} bernilai biner:

$$x_{ij} \in \{0,1\}, \quad i, j \in N_0 \quad (7)$$

Dalam implementasinya, model di atas diselesaikan menggunakan pendekatan metaheuristik AGA sehingga solusi direpresentasikan sebagai urutan kunjungan wahana.

2.3. Algoritma Genetika Adaptif

AG merupakan metode metaheuristik berbasis evolusi yang digunakan untuk mencari solusi optimal atau mendekati optimal melalui proses seleksi, *crossover*, dan mutasi [16]. Pada AG, setiap kandidat solusi disebut individu dan direpresentasikan sebagai kromosom, yaitu vektor yang elemennya disebut gen. Panjang vektor menyatakan banyaknya wahana yang harus dikunjungi. Sekumpulan individu disebut populasi, sedangkan populasi awal merupakan kumpulan kandidat solusi yang digunakan untuk memulai proses pencarian. Setiap individu dievaluasi menggunakan fungsi *fitness* untuk mengukur kualitas solusinya. Selanjutnya, populasi diperbarui melalui proses seleksi, *crossover*, dan mutasi secara berulang pada setiap

generasi (iterasi). Pada penelitian ini digunakan AGA, yaitu pengembangan AG yang menyesuaikan probabilitas *crossover* dan mutasi secara dinamis berdasarkan nilai *fitness* populasi.

Solusi TD-TSP adalah berupa urutan kunjungan wahana, di mana setiap wahana harus dikunjungi tepat satu kali. Oleh karena itu, kromosom direpresentasikan sebagai permutasi $(1, \dots, n)$ (*permutation encoding*), di mana n adalah banyaknya wahana yang dikunjungi. Setiap kromosom menyatakan urutan kunjungan seluruh wahana. Misalkan terdapat n wahana yang akan dikunjungi. Suatu kromosom dinyatakan sebagai $X = (x_1, x_2, \dots, x_n)$, dengan x_i menyatakan wahana yang dikunjungi pada urutan ke- i . Sebagai contoh, untuk lima wahana, kromosom $X = (3, 1, 5, 2, 4)$ menunjukkan bahwa urutan wahana yang dikunjungi pengunjung adalah 3, 1, 5, 2, dan 4.

Proses *decoding* kromosom digunakan untuk menerjemahkan kromosom menjadi rute perjalanan lengkap dan menghitung total waktu kunjungan berdasarkan urutan wahana yang dikunjungi. Rute lengkap diperoleh dengan menambahkan lokasi pintu masuk wahana di awal dan akhir rute. Misalkan $X = (3, 1, 5, 2, 4)$ dan 0 adalah pintu masuk wahana, maka rute kunjungan lengkapnya adalah

$$0 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 0.$$

Misalkan suatu kromosom dinyatakan sebagai urutan kunjungan $X = (x_1, x_2, \dots, x_n)$. Tetapkan waktu keberangkatan awal a_0 , kemudian waktu kedatangan pada lokasi pertama dihitung dengan menambahkan waktu tempuh dari titik awal ke lokasi tersebut, yaitu:

$$a_{x_1} = a_0 + t_{0x_1}$$

Waktu kedatangan wahana berikutnya dihitung secara rekursif dengan mempertimbangkan waktu tunggu pada lokasi sebelumnya dan waktu tempuh antar lokasi, yaitu:

$$a_{x_i} = a_{x_{i-1}} + w_{x_{i-1}}(a_{x_{i-1}}) + t_{x_{i-1}x_i}, i = 2, \dots, n.$$

Berikut adalah tahapan-tahapan AGA dalam menyelesaikan TD-TSP:

1. Inisialisasi Populasi

Inisialisasi populasi awal merupakan tahap pembentukan sejumlah kandidat solusi awal sebelum proses evolusi dilakukan. Pada tahapan ini, sejumlah kromosom dibangkitkan secara acak sebanyak ukuran populasi yang telah ditentukan.

2. Evaluasi nilai *Fitness*

Evaluasi nilai *fitness* dilakukan untuk mengukur kualitas setiap individu dalam populasi. Semakin besar nilai *fitness* suatu individu, semakin baik kualitas solusi yang dihasilkan. Karena tujuan optimisasi adalah meminimumkan total waktu kunjungan, fungsi *fitness* individu ke i didefinisikan sebagai kebalikan dari total waktu kunjungan, yaitu:

$$f_i = \frac{1}{Z(C_i)} \quad (8)$$

di mana $Z(C_i)$ adalah total waktu waktu kunjungan individu C_i . Dengan demikian, semakin kecil total waktu kunjungan, semakin besar nilai *fitness* yang diperoleh.

3. Seleksi

Seleksi merupakan proses pemilihan individu-individu terbaik dari populasi untuk dijadikan sebagai induk pada proses reproduksi selanjutnya, yaitu *crossover* dan mutasi. Proses ini bertujuan untuk mempertahankan individu dengan kualitas solusi yang baik serta meningkatkan peluang terbentuknya solusi yang lebih optimal pada generasi berikutnya. Pada penelitian ini, seleksi dilakukan menggunakan *roulette wheel selection*, di mana setiap individu memiliki probabilitas terpilih yang sebanding dengan nilai *fitness*-nya. Probabilitas seleksi individu ke- i dihitung sebagai:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (9)$$

di mana f_i merupakan nilai *fitness* individu ke- i dan N merupakan ukuran populasi. Selanjutnya, probabilitas kumulatif digunakan untuk menentukan individu yang terpilih sebagai induk (parent) berdasarkan bilangan acak yang dibangkitkan.

4. Crossover Adaptif

Crossover merupakan operator yang menggabungkan dua kromosom induk untuk menghasilkan kromosom keturunan (*offspring*) yang mewarisi sebagian informasi genetik dari masing-masing induk. Pada penelitian ini, metode *crossover* yang digunakan adalah *Order crossover* (OX). Dua titik potong dipilih secara acak pada kromosom induk. Segmen di antara kedua titik tersebut dipertahankan dari salah satu induk, sedangkan posisi yang tersisa diisi menggunakan urutan gen dari induk lainnya dengan menghindari pengulangan gen. Dengan mekanisme tersebut, kromosom hasil *crossover* tetap merupakan permutasi yang valid.

Berbeda dengan AG konvensional, probabilitas *crossover* pada penelitian ini tidak ditetapkan secara tetap pada setiap generasi, tetapi disesuaikan berdasarkan nilai *fitness* individu terhadap *fitness* rata-rata dan *fitness* maksimum populasi. Mekanisme yang digunakan diadaptasi dari [17], yaitu

$$P_c = \begin{cases} P_{cmax} - \frac{(P_{cmax} - P_{cmin})(f - f_{avg})}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ P_{cmax}, & f < f_{avg} \end{cases} \quad (10)$$

di mana P_{cmax} dan P_{cmin} masing-masing merupakan batas maksimum dan minimum probabilitas *crossover*, f merupakan nilai *fitness* terbesar dari kedua induk, f_{avg} merupakan nilai *fitness* rata-rata populasi, dan f_{max} merupakan nilai *fitness* maksimum populasi.

Persamaan (10) menunjukkan bahwa probabilitas *crossover* ditentukan berdasarkan kualitas individu relatif terhadap populasi. Jika $f < f_{avg}$, probabilitas *crossover* ditetapkan sebesar P_{cmax} untuk meningkatkan eksplorasi solusi. Sebaliknya, jika $f \geq f_{avg}$, probabilitas *crossover* menurun secara linear seiring meningkatnya nilai *fitness*, dan tetap pada rentang $[P_{cmin}, P_{cmax}]$. Mekanisme ini memberikan peluang *crossover* lebih besar pada pasangan induk dengan kualitas relatif rendah, sedangkan pasangan yang memiliki individu berkualitas tinggi memperoleh probabilitas lebih kecil untuk mengurangi perubahan terhadap solusi yang baik. Nilai probabilitas dihitung kembali pada setiap generasi berdasarkan kondisi *fitness* populasi. Jika $f_{max} = f_{avg}$, probabilitas ditetapkan sebesar P_{cmin} untuk menghindari pembagian dengan nol.

5. Mutasi Adaptif

Mutasi merupakan operator yang mengubah susunan gen pada suatu kromosom untuk menghasilkan variasi solusi baru. Pada penelitian ini digunakan *swap mutation*, di mana dua posisi gen dipilih secara acak, kemudian gen pada kedua posisi tersebut dipertukarkan. Sebagai contoh, kromosom [1,2,3,4,5] dapat berubah menjadi [1,4,3,2,5] setelah gen pada posisi kedua dan keempat dipertukarkan. Operator ini mempertahankan bentuk permutasi agar setiap wahana tetap muncul tepat satu kali.

Probabilitas mutasi ditentukan secara adaptif berdasarkan nilai *fitness* individu terhadap *fitness* rata-rata dan maksimum populasi. Mekanisme yang digunakan diadaptasi dari [17], yaitu

$$P_m = \begin{cases} P_{mmax} - \frac{(P_{mmax} - P_{mmin})(f - f_{avg})}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ P_{mmax}, & f < f_{avg} \end{cases} \quad (11)$$

dengan P_{mmax} dan P_{mmin} masing-masing merupakan batas maksimum dan minimum probabilitas mutasi.

Persamaan (11) menunjukkan bahwa probabilitas mutasi disesuaikan berdasarkan kualitas individu relatif terhadap populasi. Jika $f < f_{avg}$, probabilitas mutasi ditetapkan sebesar P_{mmax} untuk meningkatkan keragaman solusi. Sebaliknya, jika $f \geq f_{avg}$, probabilitas mutasi menurun secara linear seiring meningkatnya nilai fitness, hingga mencapai P_{mmin} ketika $f = f_{max}$. Mekanisme ini memberikan peluang mutasi lebih besar pada individu dengan kualitas relatif rendah untuk meningkatkan eksplorasi ruang solusi, sedangkan individu dengan kualitas lebih tinggi memperoleh probabilitas mutasi lebih kecil untuk mempertahankan karakteristik solusi yang baik. Probabilitas mutasi dihitung kembali pada setiap generasi berdasarkan kondisi fitness populasi. Jika $f_{max} = f_{avg}$, probabilitas mutasi ditetapkan sebesar P_{mmin} untuk menghindari pembagian dengan nol.

6. Elitisme dan Pembentukan Populasi Baru

Setelah proses seleksi, *crossover*, dan mutasi, populasi baru dibentuk dengan mempertahankan sejumlah individu terbaik dari generasi sebelumnya melalui mekanisme elitism. Individu hasil reproduksi kemudian digabungkan dengan individu elit dan dipilih sejumlah individu terbaik sesuai ukuran populasi. Mekanisme tersebut digunakan untuk memastikan solusi terbaik yang telah diperoleh tidak hilang selama proses evolusi.

7. Kriteria Pemberhentian

Proses evolusi dihentikan apabila jumlah generasi telah mencapai nilai maksimum yang ditentukan. Individu dengan nilai *fitness* terbaik yang diperoleh selama proses evolusi kemudian ditetapkan sebagai solusi terbaik.

3. Hasil dan Pembahasan

Penelitian ini mengimplementasikan model TD-TSP menggunakan AGA pada data 26 wahana di sebuah taman hiburan di Jakarta. Analisis dilakukan untuk mengevaluasi validitas implementasi, kualitas solusi yang diperoleh, perkembangan proses optimasi, perbandingan AGA dengan Algoritma Genetika (AG) konvensional, pengaruh perubahan waktu mulai kunjungan, serta pengaruh parameter AGA terhadap kualitas solusi.

3.1. Data Penelitian

Penelitian ini menggunakan data sekunder dari sebuah taman hiburan di Jakarta. Data terdiri atas lokasi dan jam operasional wahana, jarak antarwahana, waktu tempuh, serta waktu tunggu antrian. Sebanyak 26 wahana dipilih berdasarkan ketersediaan data dan kondisi operasional. Pintu masuk taman hiburan ditetapkan sebagai titik awal dan akhir perjalanan (node 0), sedangkan setiap wahana diberi kode node 1 sampai 26. Wahana memiliki waktu mulai operasional yang berbeda, yaitu pukul 10.00, 11.00, dan 12.00. Data tersebut digunakan sebagai parameter dalam model TD-TSP untuk menentukan urutan kunjungan yang meminimumkan total waktu perjalanan dan antrian. Ringkasan karakteristik data penelitian disajikan pada Tabel 1.

Tabel 1. Karakteristik Data Penelitian

Komponen Data	Sumber dan Pengolahan
Lokasi penelitian	Dunia Fantasi, Jakarta
Jumlah lokasi	26 wahana dan 1 pintu masuk
Lokasi wahana	Koordinat geografis dari Google Maps
Jam operasional	Pukul 10.00 - 15.00

Jarak antarwahana	Dihitung menggunakan rumus Haversine
Waktu tempuh	Dihitung berdasarkan kecepatan berjalan 4,5 km/jam
Waktu tunggu antrean	Data situs resmi Dufan: https://rest.ancol.com/signage/dufan/wahanadufan.php , diamati setiap satu jam
Periode pengamatan	25 Desember 2025–5 Januari 2026

Jarak antarwahana dihitung berdasarkan koordinat lintang dan bujur yang diperoleh melalui Google Maps menggunakan rumus Haversine. Metode ini menghasilkan estimasi jarak garis lurus antara dua lokasi di permukaan bumi. Selanjutnya, jarak dikonversi menjadi waktu tempuh dengan mengasumsikan kecepatan berjalan pengunjung konstan sebesar $v = 4.5$ km/jam. Waktu tempuh dari wahana i menuju wahana j dihitung menggunakan persamaan:

$$t_{ij} = \frac{d_{ij}}{v} \times 60$$

dengan t_{ij} dan d_{ij} masing-masing adalah waktu tempuh dan jarak dari i ke j .

Data waktu tunggu antrean diperoleh dari situs resmi taman hiburan yang menyediakan informasi waktu antrean secara *real-time*. Pengumpulan data dilakukan setiap satu jam selama jam operasional pada periode 25 Desember 2025 hingga 5 Januari 2026. Rata-rata waktu tunggu dihitung untuk setiap wahana pada setiap interval pengamatan. Nilai rata-rata tersebut digunakan sebagai parameter waktu tunggu yang bergantung pada waktu kedatangan pengunjung. Selengkapnya, data penelitian dan hasil pengolahan data dapat diakses pada <https://drive.google.com/drive/folders/1C6rndcAZXMu0FwzV0dDV7ljIP7Ha8kM-?usp=sharing>.

3.2. Hasil Implementasi AGA

Tahap awal implementasi AGA adalah melakukan uji parameter AGA untuk mendapatkan nilai parameter AGA yang dapat memberikan solusi terbaik. Parameter AGA ditentukan melalui pengujian pendahuluan terhadap ukuran populasi, jumlah generasi maksimum, probabilitas *crossover* maksimum, dan probabilitas mutasi maksimum. Pengujian dilakukan dengan mengubah nilai satu parameter, sedangkan parameter lainnya diambil tetap. Setiap nilai parameter diuji sebanyak lima kali untuk mengevaluasi kualitas solusi berdasarkan rata-rata total waktu kunjungan. Hasil pengujian disajikan pada Tabel 2.

Tabel 2. Hasil Pengujian Parameter AGA

Parameter	Nilai yang diuji	Nilai terpilih	Rata-rata waktu terbaik (menit)
Ukuran populasi	20; 50; 100; 150; 200	100	194,87
Jumlah generasi maksimum	50; 100; 150; 200; 250	200	195,13
P_{cmax}	0,60; 0,70; 0,80; 0,90	0,90	194,83
P_{mmax}	0,10; 0,20; 0,30; 0,40	0,20	195,09

Berdasarkan Tabel 2, ukuran populasi 100 individu menghasilkan rata-rata total waktu terendah sebesar 194,87 menit. Jumlah generasi maksimum 200 memberikan rata-rata waktu terbaik sebesar 195,13 menit, sedangkan probabilitas *crossover* maksimum 0,90 dan probabilitas mutasi maksimum 0,20 masing-masing menghasilkan rata-rata waktu terendah sebesar 194,83 dan 195,09 menit. Peningkatan nilai parameter melebihi nilai tersebut tidak memberikan perbaikan rata-rata solusi pada pengujian yang dilakukan. Oleh karena itu, implementasi AGA menggunakan ukuran populasi 100 individu, jumlah generasi maksimum 200, probabilitas *crossover* maksimum 0,90, dan probabilitas mutasi maksimum 0,20.

Probabilitas *crossover* minimum ditetapkan sebesar 0,05 dan probabilitas mutasi minimum sebesar 0,001, dengan dua individu terbaik dipertahankan melalui mekanisme elitisme pada setiap generasi.

Selanjutnya, algoritma AGA dijalankan sebanyak 10 kali dengan parameter yang sama dan inialisasi populasi yang dibangkitkan secara acak (lihat Tabel 3). Dari seluruh percobaan diperoleh solusi terbaik dengan total waktu kunjungan sebesar 193,53 menit dan nilai *fitness* sebesar 0,00516716. Solusi tersebut memiliki total waktu tempuh sebesar 71,20 menit dan total waktu tunggu sebesar 122,33 menit. Kromosom terbaik yang diperoleh adalah

[22,12,25,9,26,8,17,21,2,16,4,15,18,10,1,20,3,24,14,11,5,19,13,7,23,6].

Dengan node 0 sebagai pintu masuk, kromosom tersebut menghasilkan rute lengkap:

0-22-12-25-9-26-8-17-21-2-16-4-15-18-10-1-20-3-24-14-11-5-19-13-7-23-6-0.

Rute tersebut mengunjungi seluruh 26 wahana tepat satu kali dan kembali ke titik awal. Hasil ini menunjukkan bahwa AGA mampu memperoleh solusi layak untuk permasalahan TD-TSP dengan mempertimbangkan waktu tempuh antarwahana dan waktu tunggu antrean yang bergantung pada waktu kedatangan.

Tabel 3. Hasil 10 Kali Percobaan AGA

Percobaan	Total Waktu (menit)	<i>Fitness</i>
1	196,13	0,00509866
2	195,03	0,00512742
3	194,33	0,00514589
4	195,53	0,0051143
5	193,53	0,00516716
6	195,63	0,00511169
7	193,53	0,00516716
8	195,83	0,00510647
9	194,73	0,00513532
10	195,53	0,00511430

Tabel 4. Statistik Hasil Percobaan

Statistik	Total Waktu (menit)	<i>Fitness</i>
Solusi Terbaik	193,53	0,00516716
Solusi Terburuk	196,13	0,00509866
Rata-Rata	194,98	0,00512884
Standar Deviasi	0,93	0,00002449

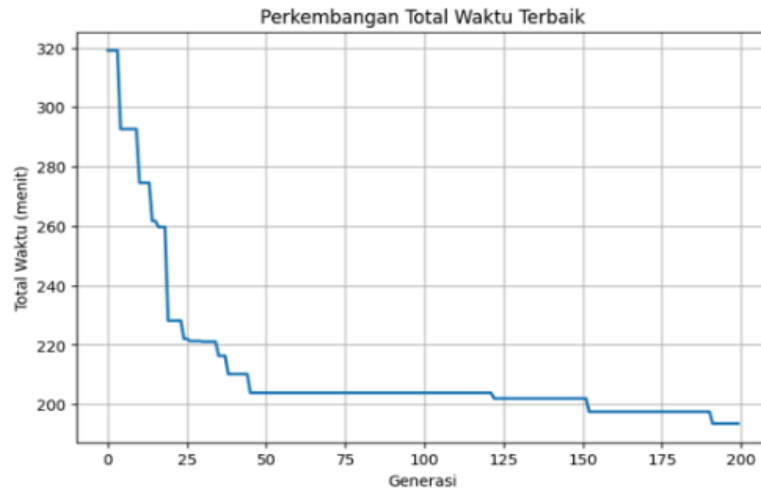
Berdasarkan sepuluh kali percobaan tersebut, diperoleh statistik hasil seperti pada Tabel 4. Rata-rata total waktu sebesar 194,98 menit dengan standar deviasi 0,93 menit menunjukkan bahwa variasi hasil antar percobaan relatif kecil. Selisih antara solusi terbaik dan terburuk hanya sebesar 2,60 menit. Hal tersebut menunjukkan bahwa meskipun proses AGA melibatkan pembangkitan populasi secara acak, hasil yang diperoleh pada 10 percobaan relatif konsisten.

3.3. Perkembangan Proses Optimisasi

Perkembangan proses optimisasi diamati melalui perubahan *fitness* terbaik, *fitness* rata-rata, probabilitas *crossover* adaptif, probabilitas mutasi adaptif, dan total waktu kunjungan terbaik pada setiap generasi.

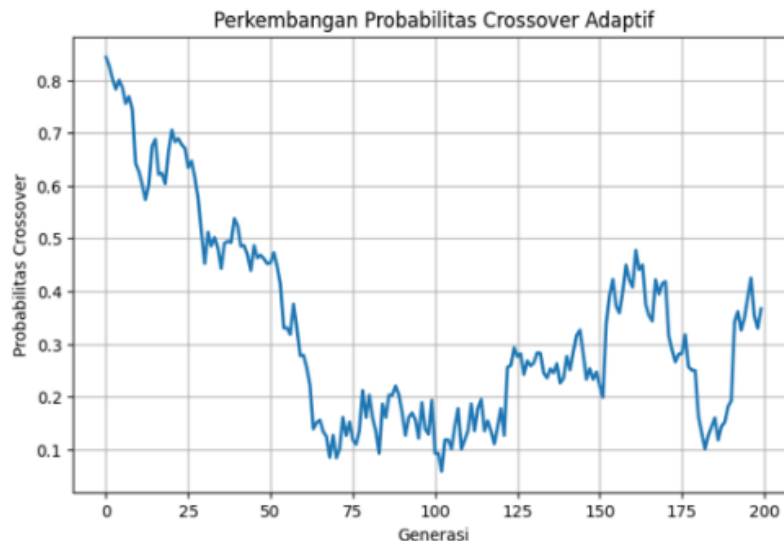
Pada generasi pertama, *fitness* terbaik sebesar 0,00313411 dan meningkat menjadi 0,00516716 pada generasi ke-200. Sementara itu, *fitness* rata-rata meningkat dari 0,00163806 pada generasi pertama menjadi 0,00422158 pada generasi ke-200. Meskipun terdapat beberapa periode ketika nilai *fitness* terbaik tidak berubah, peningkatan kembali terjadi pada beberapa generasi hingga diperoleh solusi terbaik pada generasi ke-200.

Karena fungsi *fitness* merupakan kebalikan dari total waktu kunjungan, peningkatan nilai *fitness* tersebut sejalan dengan penurunan total waktu kunjungan. Total waktu terbaik menurun dari 319,07 menit pada generasi pertama menjadi 203,88 menit pada generasi ke-50 dan mencapai 193,53 menit pada generasi ke-200. Beberapa perbaikan solusi kembali ditemukan pada generasi ke-130, ke-160, dan ke-200.

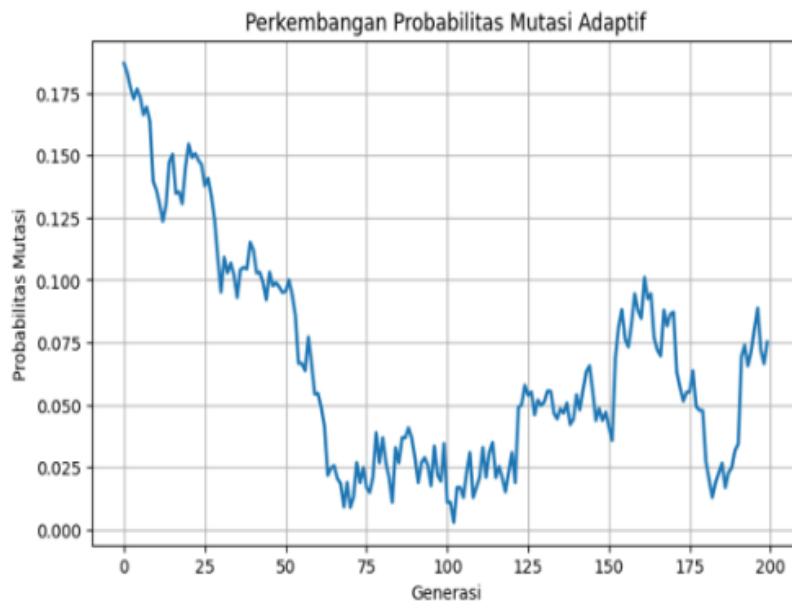


Gambar 1. Perkembangan Total Waktu Kunjungan Terbaik

Perubahan probabilitas *crossover* dan mutasi menunjukkan bahwa kedua parameter tidak menggunakan nilai tetap selama proses evolusi. Probabilitas *crossover* sebesar 0,8434 pada generasi pertama, kemudian berubah mengikuti kondisi populasi dan mencapai 0,3665 pada generasi ke-200. Probabilitas mutasi sebesar 0,1868 pada generasi pertama, kemudian berubah selama proses evolusi dan mencapai 0,0751 pada generasi ke-200.



Gambar 2. Perkembangan Probabilitas *Crossover* Adaptif



Gambar 3. Perkembangan Probabilitas Mutasi Adaptif

Perubahan kedua probabilitas tersebut menunjukkan bahwa mekanisme adaptif memungkinkan operator reproduksi menyesuaikan diri terhadap kondisi populasi selama pencarian solusi. Pada sisi lain, peningkatan *fitness* dan penurunan total waktu menunjukkan bahwa proses evolusi secara keseluruhan menghasilkan solusi yang semakin baik hingga generasi ke-200.

3.4. Perbandingan AGA dan AG

Kinerja AGA dibandingkan dengan AG konvensional. Kedua metode menggunakan data, ukuran populasi, jumlah generasi, metode seleksi, operator *crossover*, operator mutasi, dan strategi elitisme yang sama. Perbedaan terletak pada probabilitas *crossover* dan mutasi. AG probabilitas *crossover* ditetapkan sebesar 0,90 dan probabilitas mutasi sebesar 0,20, sedangkan AGA menyesuaikan kedua probabilitas tersebut secara adaptif. Masing-masing algoritma dijalankan sebanyak 10 kali. Ringkasan hasil perbandingan ditunjukkan pada Tabel 5.

Tabel 5. Perbandingan Hasil AG dan AGA

Statistik	AG	AGA
Solusi Terbaik	193,53	193,53
Solusi Terburuk	196,73	196,13
Rata-Rata	195,43	194,98
Standar Deviasi	1,01	0,93

Berdasarkan Tabel 5, kedua algoritma menghasilkan solusi terbaik yang sama, yaitu 193,53 menit. Namun, AGA menghasilkan rata-rata total waktu sebesar 194,98 menit, lebih rendah dibandingkan AG sebesar 195,43 menit. Standar deviasi AGA juga sedikit lebih kecil, yaitu 0,93 menit dibandingkan 1,01 menit pada AG. Hasil ini menunjukkan bahwa AGA menghasilkan rata-rata waktu kunjungan dan variasi hasil yang lebih rendah pada percobaan yang dilakukan.

Untuk mengevaluasi signifikansi perbedaan rata-rata kedua algoritma, dilakukan *Welch's t-test* pada taraf signifikansi 5%. Hasil pengujian menunjukkan nilai $p = 0,313$ dengan interval kepercayaan 95% untuk selisih rata-rata sebesar $[-1,361, 0,461]$ menit. Karena nilai $p > 0,05$ dan interval kepercayaan memuat nol, perbedaan rata-rata kedua algoritma tidak signifikan secara statistik. Dengan demikian, meskipun AGA menghasilkan rata-rata total waktu

kunjungan yang lebih rendah, hasil pengujian belum menunjukkan keunggulan yang signifikan dibandingkan AG.

3.5. Analisis Sensitivitas terhadap Waktu Mulai Kunjungan

Analisis sensitivitas dilakukan dengan mengubah waktu mulai kunjungan menjadi pukul 10.00, 10.30, 11.00, 11.30, dan 12.00, sedangkan data dan parameter algoritma lainnya dipertahankan tetap. Setiap kondisi waktu mulai diuji sebanyak lima kali. Hasil analisis sensitivitas disajikan pada Tabel 6.

Tabel 6. Hasil Analisis Sensitivitas Waktu Mulai

Waktu Mulai	Rata-rata waktu tempuh (menit)	Rata-rata waktu tunggu (menit)	Rata-rata total waktu (menit)
10.00	81,90	113,01	194,91
10.30	63,68	131,82	195,50
11.00	59,02	227,19	286,21
11.30	64,92	312,30	377,22
12.00	69,14	414,01	483,15

Berdasarkan Tabel 6, perubahan waktu mulai kunjungan menyebabkan peningkatan yang besar pada total waktu kunjungan. Waktu mulai pukul 10.00 menghasilkan rata-rata total waktu terendah sebesar 194,91 menit, sedangkan waktu mulai pukul 12.00 menghasilkan rata-rata total waktu tertinggi sebesar 483,15 menit. Peningkatan tersebut terutama disebabkan oleh perubahan waktu tunggu antrean, yang meningkat dari 113,01 menit menjadi 414,01 menit.

Peningkatan waktu tunggu berkaitan dengan pola antrean wahana yang cenderung lebih panjang pada siang hari. Berdasarkan data pengamatan, rata-rata waktu tunggu wahana Arung Jeram meningkat dari 10,00 menit pada pukul 12.00 menjadi 49,17 menit pada pukul 14.00. Pola serupa terjadi pada wahana *Ice Age*, dengan waktu tunggu meningkat dari 39,44 menit menjadi 59,17 menit. Ketika kunjungan dimulai lebih siang, waktu kedatangan pada sejumlah wahana bergeser ke interval dengan antrean lebih panjang. Waktu tunggu yang lebih lama juga menunda kedatangan pada wahana berikutnya sehingga dapat meningkatkan akumulasi waktu tunggu sepanjang rute. Sebaliknya, kunjungan yang dimulai lebih awal memberikan peluang untuk mengunjungi sejumlah wahana pada interval dengan waktu tunggu lebih rendah.

Hasil analisis menunjukkan bahwa perubahan waktu mulai terutama memengaruhi komponen waktu tunggu, sedangkan waktu tempuh berubah akibat perbedaan urutan kunjungan yang dihasilkan. Temuan ini memperlihatkan karakteristik *time-dependent* pada model, yaitu total waktu kunjungan ditentukan tidak hanya oleh urutan wahana, tetapi juga oleh waktu kedatangan pada setiap wahana. Dengan demikian, pemilihan waktu mulai kunjungan menjadi faktor penting dalam meminimumkan total waktu kunjungan.

3.6. Analisis Pengaruh Parameter Algoritma

Analisis parameter dilakukan untuk mengevaluasi pengaruh ukuran populasi, jumlah generasi maksimum, probabilitas *crossover* maksimum, dan probabilitas mutasi maksimum terhadap kualitas solusi AGA. Pengujian dilakukan dengan mengubah satu parameter, sedangkan parameter lainnya dipertahankan tetap. Setiap nilai parameter diuji sebanyak lima kali dan dievaluasi berdasarkan rata-rata total waktu kunjungan.

1. Pengaruh Ukuran Populasi

Ukuran populasi yang diuji adalah 20, 50, 100, 150, dan 200 individu, dengan rata-rata total waktu masing-masing sebesar 201,35; 197,46; 194,87; 195,11; dan 195,77 menit. Peningkatan ukuran populasi hingga 100 individu cenderung memperbaiki kualitas solusi karena menyediakan lebih banyak alternatif dalam proses pencarian. Namun, peningkatan menjadi 150 dan 200 individu tidak memberikan perbaikan lebih lanjut. Pada pengujian ini,

ukuran populasi 100 individu menghasilkan rata-rata total waktu terendah sebesar 194,87 menit.

2. Pengaruh Jumlah Generasi Maksimum

Jumlah generasi maksimum yang diuji adalah 50, 100, 150, 200, dan 250 generasi, dengan rata-rata total waktu masing-masing sebesar 206,18; 197,80; 196,08; 195,13; dan 196,53 menit. Peningkatan jumlah generasi hingga 200 cenderung menghasilkan solusi yang lebih baik karena memberikan kesempatan lebih banyak bagi algoritma untuk memperbaiki populasi. Namun, peningkatan menjadi 250 generasi tidak menurunkan rata-rata total waktu. Dengan demikian, jumlah generasi maksimum 200 memberikan hasil terbaik pada pengujian ini.

3. Pengaruh Probabilitas Crossover Maksimum

Nilai P_{cmax} yang diuji adalah 0,60; 0,70; 0,80; dan 0,90, dengan rata-rata total waktu masing-masing sebesar 197,19; 195,75; 195,23; dan 194,83 menit. Hasil tersebut menunjukkan bahwa peningkatan probabilitas *crossover* maksimum cenderung memperbaiki kualitas solusi pada rentang pengujian. Nilai $P_{cmax} = 0,90$ menghasilkan rata-rata total waktu terendah sebesar 194,83 menit. Jadi, probabilitas *crossover* yang lebih tinggi memberikan peluang lebih besar untuk membentuk kombinasi urutan kunjungan baru dari kromosom induk.

4. Pengaruh Probabilitas Mutasi Maksimum

Nilai P_{mmax} yang diuji adalah 0,10; 0,20; 0,30; dan 0,40, dengan rata-rata total waktu masing-masing sebesar 196,31; 195,09; 195,21; dan 196,05 menit. Hasil pengujian menunjukkan bahwa peningkatan probabilitas mutasi tidak selalu memperbaiki kualitas solusi. Nilai $P_{mmax} = 0,20$ menghasilkan rata-rata total waktu terendah sebesar 195,09 menit. Dengan demikian, probabilitas mutasi yang terlalu rendah dapat membatasi keragaman solusi, sedangkan probabilitas yang terlalu tinggi berpotensi mengganggu susunan gen yang telah menghasilkan solusi baik.

5. Pengaruh Kombinasi Parameter

Kualitas solusi AGA berpotensi dipengaruhi oleh interaksi antarparameter. Ukuran populasi dan jumlah generasi menentukan banyaknya kandidat solusi dan kesempatan pencarian, sedangkan probabilitas *crossover* dan mutasi mengatur pembentukan serta perubahan solusi selama proses evolusi. Sebagai contoh, pengaruh probabilitas mutasi dapat berbeda pada ukuran populasi yang berbeda karena keduanya berkaitan dengan keragaman solusi. Demikian pula, probabilitas *crossover* yang tinggi dapat memberikan hasil berbeda bergantung pada jumlah generasi yang digunakan. Berdasarkan pengujian masing-masing parameter, konfigurasi yang dipilih adalah ukuran populasi 100 individu, jumlah generasi maksimum 200, $P_{cmax} = 0,90$, dan $P_{mmax} = 0,20$. Namun, karena pengujian dilakukan dengan mengubah satu parameter pada satu waktu, pengaruh interaksi antarparameter belum dievaluasi secara langsung. Oleh karena itu, pengujian kombinasi parameter secara sistematis diperlukan pada penelitian selanjutnya untuk mengevaluasi pengaruh interaksi tersebut terhadap kualitas solusi.

4. Kesimpulan

Berdasarkan hasil penelitian, model TD-TSP berbasis waktu tunggu dapat diselesaikan menggunakan AGA untuk menentukan urutan kunjungan 26 wahana pada sebuah taman hiburan di Jakarta. AGA menghasilkan solusi terbaik dengan total waktu kunjungan sebesar 193,53 menit dan rata-rata sebesar 194,98 menit dengan standar deviasi 0,93 menit dari 10 percobaan. Dibandingkan dengan AG konvensional, AGA menghasilkan rata-rata total waktu yang lebih rendah sebesar 0,45 menit dan standar deviasi yang lebih kecil, meskipun perbedaan rata-rata tersebut tidak signifikan secara statistik berdasarkan *Welch's t-test*. Analisis sensitivitas menunjukkan bahwa waktu mulai kunjungan memengaruhi total waktu, terutama

akibat perubahan waktu tunggu antrean yang bergantung pada waktu kedatangan. Pengujian parameter menunjukkan bahwa ukuran populasi 100 individu, 200 generasi, probabilitas *crossover* maksimum 0,90, dan probabilitas mutasi maksimum 0,20 memberikan hasil terbaik di antara nilai parameter yang diuji.

Penelitian ini masih menggunakan waktu tempuh antarwahana yang konstan dan rata-rata waktu tunggu berdasarkan interval pengamatan. Oleh karena itu, penelitian selanjutnya dapat mempertimbangkan waktu tempuh yang bergantung pada waktu dan kepadatan jalur pengunjung, serta mengintegrasikan data antrean *real-time* untuk memperbarui estimasi waktu tunggu dan urutan kunjungan secara dinamis. Pengujian interaksi antarparameter AGA juga dapat dilakukan untuk mengevaluasi pengaruh kombinasi parameter terhadap kualitas solusi.

Ucapan Terima Kasih

Tulis ucapan terima kasih Saudara disini (jika ada). Gunakan fasilitas Reference Program seperti Mendeley, EndNote, atau Zotero. Referensi yang ditulis dalam daftar pustaka hanyalah referensi yang digunakan di dalam artikel saja. Format daftar pustaka dan sitasi menggunakan format IEEE.

Daftar Pustaka

- [1] L. G. A. Candrawati and I. G. A. G. A. Kadyanan, "Optimasi traveling salesman Problem (TSP) Untuk rute paket wisata di bali dengan algoritma genetika," *Jurnal Ilmiah Komputer*, vol. 10, no. 1, pp. 27–32, 2017.
- [2] H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa, "The time dependent traveling salesman problem: Polyhedra and algorithm," *Math. Program. Comput.*, vol. 5, no. 1, pp. 27–55, 2013, doi: 10.1007/s12532-012-0047-y.
- [3] D. Minh Vu, M. Hewitt, N. Boland, and M. Savelsbergh, "Solving time dependent traveling salesman problems with time windows," *Optim. Online*, vol. 6640, pp. 1–30, 2018.
- [4] D. A. Suprayogi and W. F. Mahmudy, "Penerapan algoritma genetika traveling salesman problem with time window: Studi kasus rute antar jemput laundry," *J. Buana Inform.*, vol. 6, no. 2, pp. 121–130, 2015, doi: 10.24002/jbi.v6i2.407.
- [5] Z. Yang, L. Jiacheng, and L. Lei, "Time-dependent theme park routing problem by partheno-genetic algorithm," *Mathematics*, vol. 8, no. 12, pp. 1–19, 2020, doi: 10.3390/math8122193.
- [6] I. K. Herdiana, I. M. Candiasa, and G. Indrawan, "Optimization of adaptive genetic algorithm parameters in traveling salesman problem," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 4, no. 2, pp. 169–176, Jul. 2022, doi: 10.47709/cnahpc.v4i2.1581.
- [7] C. R. Gunawan, "Optimization of the travelling salesman problem based on genetic algorithm with adaptive crossover and mutation probabilitie," *J. Ilmu Komput. dan Sist. Inf*, vol. 3, no. 3, pp. 234–242, Sep. 2024, doi: 10.70340/jirsi.v3i3.166.
- [8] H. Li, "Solving the TSP problem based on improved genetic algorithm," *Appl. Comput. Eng.*, vol. 133, no. 1, pp. 73–79, 2025, doi: 10.54254/2755-2721/2025.20603.

- [9] Y. Taishan, C. Xin, and Y. Baoshuang, “An adaptive genetic algorithm with double populations for solving traveling salesman problems,” *J. Intell. Syst.*, vol. 34, no. 1, 2025, doi: 10.1515/jisys-2025-0040.
- [10] V. Cacchiani, C. Contreras-Bolton, and P. Toth, “Models and algorithms for the traveling salesman problem with Time-dependent service times,” *Eur. J. Oper. Res.*, vol. 283, no. 3, pp. 825–843, Jun. 2020, doi: 10.1016/j.ejor.2019.11.046.
- [11] D. M. Vu, M. Hewitt, and D. D. Vu, “Solving the time dependent minimum tour duration and delivery man problems with dynamic discretization discovery,” *Eur. J. Oper. Res.*, vol. 302, no. 3, pp. 831–846, Nov. 2022, doi: 10.1016/j.ejor.2022.01.029.
- [12] G. Lera-Romero, J. J. Miranda Bront, and F. J. Soullignac, “Dynamic programming for the time-dependent traveling salesman problem with time windows,” *INFORMS J. Comput.*, vol. 34, no. 6, pp. 3292–3308, Nov. 2022, doi: 10.1287/ijoc.2022.1236.
- [13] R. Fontaine, J. Dibangoye, and C. Solnon, “Exact and anytime approach for solving the time dependent traveling salesman problem with time windows,” *Eur. J. Oper. Res.*, vol. 311, no. 3, pp. 833–844, Dec. 2023, doi: 10.1016/j.ejor.2023.06.001.
- [14] D. Chen, C. Imdahl, D. Lai, and T. Van Woensel, “The dynamic traveling salesman problem with time-dependent and stochastic travel times: A deep reinforcement learning approach,” *Transp. Res. Part C Emerg. Technol.*, vol. 172, p. 105022, Mar. 2025, doi: 10.1016/j.trc.2025.105022.
- [15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 13th ed. USA: Addison-Wesley, 1989.