

# Pencarian Jalur Terpendek dari Kota Surabaya ke Wilayah Jawa Timur dengan Algoritma Genetika

Achmad Fahmi Al Hafidz<sup>1</sup>, Diaz Prisheila Dharmawan<sup>2</sup>, Imeldha Elzandy<sup>3</sup>, Adelia Putri Adyani<sup>4</sup>, Shinta Dwi Anggraeni<sup>5</sup>, Firyal Wishal Nabili<sup>6</sup>, Andreas Nugroho Sihananto<sup>7</sup>

<sup>1,2,3,4,5,6,7</sup> Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur

<sup>1</sup>[21081010223@student.upnjatim.ac.id](mailto:21081010223@student.upnjatim.ac.id)

<sup>2</sup>[21081010027@student.upnjatim.ac.id](mailto:21081010027@student.upnjatim.ac.id)

<sup>3</sup>[21081010039@student.upnjatim.ac.id](mailto:21081010039@student.upnjatim.ac.id)

<sup>4</sup>[21081010047@student.upnjatim.ac.id](mailto:21081010047@student.upnjatim.ac.id)

<sup>5</sup>[21081010215@student.upnjatim.ac.id](mailto:21081010215@student.upnjatim.ac.id)

<sup>6</sup>[21081010301@student.upnjatim.ac.id](mailto:21081010301@student.upnjatim.ac.id)

<sup>7</sup>[21081010301@student.upnjatim.ac.id](mailto:21081010301@student.upnjatim.ac.id)

\*Corresponding author email: [andreas\\_nugroho.jarkom@upnjatim.ac.id](mailto:andreas_nugroho.jarkom@upnjatim.ac.id)

**Abstrak**— Penelitian ini mengevaluasi efektivitas Algoritma Genetika dalam menyelesaikan masalah Travelling Salesman Problem (TSP) antara Kota Surabaya dan wilayah-wilayah di Jawa Timur. TSP merupakan masalah optimasi di mana penjual harus mengunjungi sejumlah kota dengan jarak total minimum dan kembali ke kota asal. Algoritma Genetika digunakan karena kemampuannya yang kuat dalam menangani ruang pencarian yang luas dan menghindari terjebak dalam solusi optimal lokal, dengan melibatkan proses seleksi, crossover, dan mutasi untuk menghasilkan solusi yang lebih baik dari generasi ke generasi. Data jarak antara Kota Surabaya dan berbagai kota di Jawa Timur dikumpulkan sebagai input, dengan jarak mulai dari 28.5 km ke Kabupaten Gresik hingga 303.0 km ke Kabupaten Banyuwangi. Hasil penelitian menunjukkan bahwa Algoritma Genetika berhasil menemukan rute yang lebih efisien dibandingkan metode pencarian rute biasa. Rute optimal pada penelitian ini mendapatkan hasil total jarak 2335.30 km. Penelitian ini menunjukkan bahwa Algoritma Genetika dapat mengoptimalkan rute perjalanan, menghemat waktu dan biaya, serta meningkatkan efisiensi transportasi. Namun, kelemahan seperti kebutuhan waktu komputasi tinggi dan ketergantungan pada ukuran populasi serta probabilitas crossover dan mutasi diidentifikasi. Meskipun demikian, hasilnya konsisten dan signifikan, membuktikan potensi besar Algoritma Genetika dalam aplikasi praktis untuk masalah optimasi rute. Penelitian ini membuka peluang untuk pengembangan lebih lanjut dengan mempertimbangkan faktor tambahan seperti kondisi lalu lintas, waktu tempuh aktual, dan cuaca. Implementasi Algoritma Genetika dalam sistem navigasi yang lebih canggih dapat memberikan manfaat besar bagi sistem transportasi di Jawa Timur dan daerah lainnya.

**Kata Kunci**— Travelling Salesman Problem, Algoritma Genetika, Jalur Terpendek, Navigasi, Heuristik

## I. PENDAHULUAN

Di zaman sekarang, navigasi menjadi semakin mudah dengan berbagai pilihan rute jalan yang tersedia. Pada kasus navigasi tentunya kita mencari rute yang paling efektif dan efisien. Masalah ini disebut Travelling Salesman Problem (TSP). Dalam Travelling Salesman Problem, seorang penjual harus mengunjungi sejumlah kota dengan meminimalkan jarak total yang ditempuh. Setiap kota harus dikunjungi tepat satu kali, dan penjual harus kembali ke titik awal. Masalah ini dapat diterapkan untuk kasus mencari rute terdekat dari kota A

ke kota I. Surabaya sebagai ibu kota provinsi Jawa Timur dikenal dengan kepadatan lalu lintas perjalanan ke berbagai kota lain [1]. Hal ini dapat menjadi penyelesaian masalah untuk pencarian Jalur terpendek antara kota Surabaya dan wilayah di provinsi Jawa Timur. Permasalahan ini dikenal dengan istilah "pencarian rute optimum" dalam *Travelling Salesman Problem* dan bertujuan untuk meminimalkan jarak tempuh total [2].

Algoritma genetika adalah jenis kecerdasan buatan (AI) yang meniru seleksi alam untuk menyelesaikan masalah optimasi. Ini melibatkan pembuatan populasi solusi, memilih solusi terbaik, dan menggabungkannya untuk menciptakan generasi solusi baru. Algoritma genetika sangat cocok untuk menyelesaikan masalah optimasi yang kompleks karena dapat menangani ruang pencarian yang luas dan tidak mudah terjebak dalam optima lokal. Algoritma Genetika (GA) merupakan metode yang sering digunakan untuk menyelesaikan TSP, dengan meniru proses genetika pada makhluk hidup. Proses seleksi, crossover, dan mutasi dilakukan untuk menghasilkan kromosom terbaik pada setiap generasi [3]. Dengan menerapkan algoritma genetika pada pencarian jalur terpendek antara kota Surabaya dan wilayah di provinsi Jawa Timur kita dapat mengoptimalkan rute dan meningkatkan efisiensi, mengurangi konsumsi bahan bakar dan menghemat waktu. Pendekatan ini juga dapat diterapkan pada masalah optimasi lainnya di berbagai sektor.

Tujuan penelitian ini adalah untuk mengevaluasi efektivitas algoritma genetika dalam menemukan solusi rute yang mendekati optimal untuk pemodelan kasus pencarian jalur terpendek antara kota Surabaya dan wilayah di provinsi Jawa Timur, dengan menggunakan pendekatan Travelling Salesman Problem (TSP) yang diselesaikan dengan metode Algoritma Genetika.

## II. LANDASAN TEORI

### A. Travelling Salesman Problem

Travelling Salesman Problem dikenal sebagai salah satu masalah optimasi yang termasuk dalam kategori persoalan teori graf. Nama masalah ini diilhami oleh skenario di mana seorang pedagang berkeliling mengunjungi sejumlah kota,

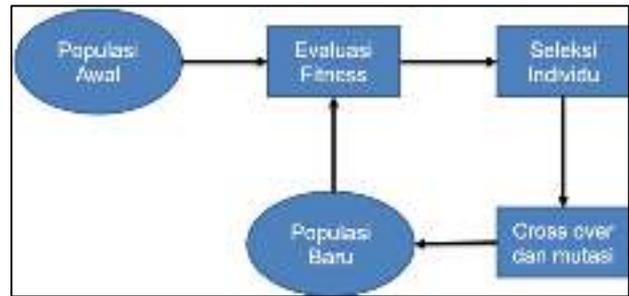
Idengan tujuan untuk menemukan rute terpendek yang memungkinkan dia mengunjungi setiap kota sekali saja dan kembali ke kota asalnya [4]. Permasalahan Travelling Salesman Problem (TSP) berakar dari permainan Icosian Hamilton yang diciptakan oleh William Rowan Hamilton dan Thomas Pennington pada tahun 1800. Permainan ini mengharuskan pemain untuk menghubungkan 20 titik dengan jalur tertentu, yang kemudian dikenal sebagai teori Sirkuit Hamilton dalam matematika diskrit. Bentuk umum TSP pertama kali dipelajari oleh Karl Menger di Vienna dan Harvard pada tahun 1930. Kemudian, Hassler Whitney dan Merrill Flood mempublikasikannya di Princeton. Kemudian perkembangan Travelling Salesman Problem dikembangkan dalam makalah "On the history of combinatorial optimization (till 1960)" [5]. Dalam konteks Travelling Salesman Problem, setiap kota direpresentasikan sebagai node dalam sebuah graf, dan perjalanan antara dua kota direpresentasikan sebagai edge dengan bobot tertentu yang menunjukkan jarak atau biaya perjalanan. Tantangan utama dalam Travelling Salesman Problem adalah menemukan rute optimal yang meminimalkan total jarak atau biaya perjalanan.

Masalah ini memiliki aplikasi luas dalam berbagai bidang, termasuk logistik, manufaktur, dan pemrograman rute kendaraan. Karena kompleksitasnya yang tinggi [6], Travelling Salesman Problem termasuk dalam kategori NP-hard, yang berarti tidak ada algoritma yang dapat menyelesaikannya dalam waktu polinomial untuk semua kasus. Namun, berbagai pendekatan heuristik dan algoritma eksak telah dikembangkan untuk mencari solusi yang mendekati optimal, seperti algoritma genetika, simulated annealing, dan algoritma branch and bound[14].

### B. Algoritma Genetika

Algoritma Genetika (GA) adalah salah satu metode yang sering digunakan untuk menyelesaikan Travelling Salesman Problem. Algoritma ini meniru cara kerja proses genetika pada makhluk hidup, dimana terdapat proses seleksi, *crossover*, dan mutasi untuk mendapatkan kromosom terbaik pada suatu generasi [3].

Algoritma Genetika meniru proses evolusi dalam alam untuk menemukan solusi optimal bagi suatu masalah. Algoritma ini dimulai dengan mengambil populasi awal. Populasi awal terdiri dari individu-individu yang mewakili berbagai solusi. Setiap individu dievaluasi dengan nilai fitness yang menunjukkan kecocokan solusinya. Individu dengan fitness tinggi memiliki peluang lebih besar untuk terpilih dan bereproduksi melalui proses *crossover*. *Crossover* menggabungkan gen (bagian dari solusi) dari dua individu terpilih untuk menghasilkan individu baru yang diharapkan memiliki solusi lebih baik. Selanjutnya, individu baru mengalami mutasi, di mana gen diubah secara acak. Proses mutasi ini membantu memperkenalkan variasi dalam populasi, mencegah terjebak pada solusi lokal. Hasil dari reproduksi dan mutasi menghasilkan populasi baru yang diharapkan memiliki solusi yang lebih baik dari generasi sebelumnya. Proses ini diulang sampai solusi yang optimal ditemukan [7]. Proses atau flowchart Algoritma dapat dilihat di gambar 1.



Gambar 1. Flowchart Algoritma Genetika

GA terbukti efektif dalam menyelesaikan berbagai masalah optimasi, termasuk Travelling Salesman Problem.

### C. Metode Shortest Path dan Connected Graph

Untuk menyelesaikan permasalahan Travelling Salesman Problem, salah satu metode yang sering digunakan adalah Shortest Path. Metode ini bertujuan untuk menemukan lintasan terpendek pada sebuah graf bersambung dan berarah. Shortest Path dapat dihitung secara manual atau dengan menggunakan komputer. Hasil Shortest Path akan meminimumkan fungsi linear khusus dari Path seperti jarak, waktu, dan biaya yang dihadapi selama perjalanan. Permasalahan yang dihadapi dalam Travelling Salesman Problem adalah bagaimana cara mengunjungi vertek pada graf dari vertek awal ke vertek akhir dengan bobot minimum [8].

Bobot yang digunakan dalam Travelling Salesman Problem adalah jarak, dan kota-kota yang dikunjungi diasumsikan sebagai graf yang saling terhubung (*connected graph*) antar suatu kota dengan kota lainnya. Suatu graf *G* disebut terhubung jika untuk setiap vertek dari graf terdapat jalur yang menghubungkan kedua vertek tersebut. Dengan kata lain, *graph* terhubung jika setiap vertek yaitu  $V_i$  dan  $V_j$  dalam suatu *graph* terdapat sedikitnya sebuah *edge*. *Edge* pada *graph* berarah disebut *arc* [9].

## III. METODOLOGI PENELITIAN

### A. Pengumpulan data

Metode pengumpulan data dalam penelitian ini melibatkan pencarian daring melalui situs [www.kaggle.com](http://www.kaggle.com). Informasi diperoleh dari Google Maps dan mencakup jarak, durasi perjalanan, dan rute antara kota Surabaya dengan kota dan kabupaten di Jawa Timur, Indonesia. Data ini memiliki peran vital dalam perencanaan kota, pengembangan infrastruktur, serta perbaikan sistem transportasi di wilayah tersebut.

Tabel I  
Dataset

No	Asal kota	Kota tujuan	Jarak	Durasi Perjalanan
1.	Kota	Kota	51,9 km	1 hour

	Surabaya	Mojokerto		2 mins
2.	Kota Surabaya	Kota Blitar	163 km	3 hours 16 mins
3.	Kota Surabaya	Kota Madiun	165 km	2 hours 15 mins
4.	Kota Surabaya	Kota Pasuruan	69,3 km	1 hour 18 mins
5.	Kota Surabaya	Kota Batu	104 km	1 hour 58 mins
37.	Kota Surabaya	Kabupaten Malang	133 km	2 hours 41 mins

Tabel I Dataset merupakan variabel yang diperlukan untuk menghitung jalur terpendek dari kota Surabaya ke kota dan kabupaten di Jawa Timur menggunakan algoritma genetika.

1. Asal Kota

Merupakan variabel yang menunjukkan kota awal perjalanan, digunakan sebagai acuan dalam perhitungan jalur terpendek antara kota dan kabupaten di wilayah Jawa Timur.

2. Kota Tujuan

Merupakan variabel yang menunjukkan kota atau kabupaten tempat perjalanan diakhiri.

3. Jarak

Merupakan variabel yang menyatakan jarak antara kota asal dan kota tujuan, digunakan sebagai parameter dalam perhitungan jalur terpendek.

4. Durasi Perjalanan

Merupakan variabel yang menyatakan waktu yang diperlukan untuk melakukan perjalanan dari kota asal ke kota tujuan.

B. Algoritma Genetika

1) Populasi awal

Populasi awal merupakan pemilihan data secara acak, dimana populasi terdiri dari sejumlah kromosom. Setiap kromosom terdiri dari rute kota yang dimulai dari Surabaya menuju tujuan kota tertentu, kemudian kembali lagi ke Surabaya. Hal ini bertujuan untuk memudahkan pengguna mengetahui rute tercepat.

$$\text{Kromosom 1 } (v_1) = A1 - A30 - A20 - A3 - A15 - A16 - A14$$

$$- A13 - A12 - A11 - A9 - \dots - A1$$

$$\text{Kromosom 2 } (v_2) = A1 - A20 - A30 - A37 - A5 - A2 - A32 - A31 - A27 - A22 - A21 - \dots - A1$$

$$\text{Kromosom 3 } (v_3) = A1 - A35 - A36 - A5 - A6 - A4 - A3 - A38 - A37 - A30 - A29 - \dots - A1$$

$$\text{Kromosom 4 } (v_4) = A1 - A30 - A8 - A36 - A23 - A5 - A32 - A7 - A6 - A4 - A28 - \dots - A1$$

$$\text{Kromosom 5 } (v_5) = A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1$$

2) Proses evaluasi

Proses evaluasi dalam algoritma genetika merupakan tahap di mana setiap kromosom dinilai untuk mengetahui kualitasnya sebagai solusi potensial melalui perhitungan nilai kebugaran (fitness) yang digunakan untuk memberikan nilai numerik untuk mencerminkan seberapa baik nilai pada kromosom. kromosom yang memiliki nilai kebugaran (fitness) yang lebih rendah akan dianggap menjadi solusi yang lebih baik dan memiliki peluang besar untuk dipilih dalam proses seleksi, crossover, dan mutasi [11]. Proses evaluasi dapat dilakukan dengan rumus berikut :

$$F_i = \frac{1}{f_i}$$

dengan i merupakan kromosom

Keterangan :

$F_i$  : Nilai Fitness kromosom ke-i

$f_i$  : Panjang lintasan kromosom ke-i

Pada proses evaluasi akan melakukan pencarian nilai fitness setiap kromosom. Hasil perhitungan panjang lintasan dan nilai fitness sebagai berikut :

$$f_1 (A1 - A30 - A20 - A3 - A15 - A16 - A14 - A13 - A12 - A11 - A9 - \dots - A1) = 2348.9493724823941 \text{ km}$$

$$f_2 (A1 - A20 - A30 - A37 - A5 - A2 - A32 - A31 - A27 - A22 - A21 - \dots - A1) = 2129.0903486670986 \text{ km}$$

$$f_3 (A1 - A35 - A36 - A5 - A6 - A4 - A3 - A38 - A37 - A30 - A29 - \dots - A1) = 2298.0192502369555 \text{ km}$$

$$f_4 (A1 - A30 - A8 - A36 - A23 - A5 - A32 - A7 - A6 - A4 - A28 - \dots - A1) = 2498.2604569741175 \text{ km}$$

$$f_5 (A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1) = 2195.712093190961 \text{ km}$$

$$f_5 (A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1) = 2195.712093190961 \text{ km}$$

$$f_5 (A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1) = 2195.712093190961 \text{ km}$$

$$f_5 (A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1) = 2195.712093190961 \text{ km}$$

$$f_5 (A1 - A14 - A31 - A32 - A36 - A2 - A5 - A7 - A6 - A4 - A3 - \dots - A1) = 2195.712093190961 \text{ km}$$

Perhitungan nilai fitness dilakukan menggunakan rumus

$$F_i = \frac{1}{f_i}$$

dengan hasil :

Tabel II  
Penilaian Fitness

Kromosom	$f_i$	$F_i$
$v_1$	2348.949	0,00042572

$v_2$	2129.090	0,00046968
$v_3$	2298.019	0,00043516
$v_4$	2498.260	0,00040028
$v_5$	2195.712	0,00045543
Total		0,00218627

3) Seleksi

Seleksi adalah tahap penting dalam algoritma genetika di mana kita mencari kromosom terbaik dalam satu generasi berdasarkan nilai fitness mereka[11]. Dalam proses ini, kromosom dipilih sesuai dengan nilai fitness yang telah dihitung sebelumnya. Tujuan dari proses seleksi adalah untuk memilih individu-individu yang akan menjadi calon induk untuk proses *crossover* dan mutasi.

Tabel III  
Perhitungan Seleksi

Kromosom	$f_i$	$F_i$	P	Q
$v_1$	2348.949	0,00042572	0,194	0,194
$v_2$	2129.090	0,00046968	0,214	0,408
$v_3$	2298.019	0,00043516	0,199	0,607
$v_4$	2498.260	0,00040028	0,183	0,790
$v_5$	2195.712	0,00045543	0,208	0,998
Total		0,00218627		

Setelah nilai fitness kumulatif ditemukan, langkah selanjutnya adalah melakukan perhitungan random permutasi menggunakan python. Langkah ini dilakukan untuk menciptakan calon induk baru yang akan digunakan dalam proses *crossover*. Berikut adalah hasil dari acak mutasi yang

diperoleh:

Tabel IV

Perhitungan Permutasi

Kromosom	P	Q	r	Calon Induk
$v_1$	0,194	0,194	0,251	$v_2$
$v_2$	0,214	0,408	0,687	$v_4$

$v_3$	0,199	0,607	0,134	$v_1$
$v_4$	0,183	0,790	0,802	$v_5$
$v_5$	0,208	0,998	0,423	$v_3$

Dari pencarian acak permutasi, calon induk untuk *crossover* ditentukan dengan aturan sebagai berikut: jika nilai  $r < Q_1$ , maka kromosom pertama dipilih tetapi jika nilai  $r > Q_1$ . Jika nilai  $r$  berada di antara  $Q_i$  dan  $Q_{(i+j)}$  dengan  $j = (1, 2, \dots, UkPop)$ , maka kromosom yang dipilih adalah kromosom ke- $(i+j)$ .

4) Crossover

*Crossover* (pindah silang) dilakukan dengan memilih dua induk secara acak dari populasi untuk menghasilkan gen baru[10]. Penggunaan metode *one-cut-point* (satu titik potong) pada *crossover* adalah dengan secara acak memilih satu titik potong, lalu menukarkan sebagian dari induk untuk menghasilkan keturunan baru[15].

Proses *crossover* berlangsung setelah seleksi induk selesai. Semakin tinggi nilai probabilitas persilangan, semakin cepat struktur tunggal baru terbentuk dalam populasi[11]. Calon induk yang terpilih menggunakan probabilitas persilangan 50% untuk dilakukan *crossover*.

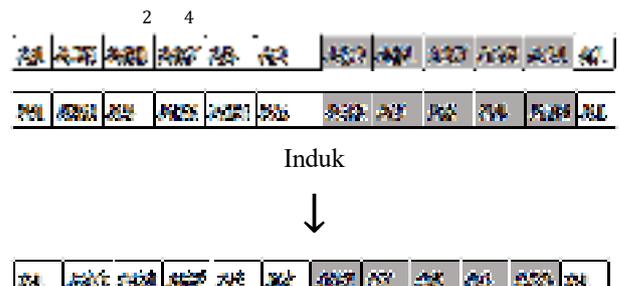
Tabel V

Crossover

Calon Induk	R	Induk Terpilih
$v_2$	0,241	$v_2$
$v_4$	0,488	$v_4$
$v_1$	0,849	
$v_5$	0,344	$v_5$
$v_3$	0,150	$v_3$

Induk terpilih  $v_2, v_4, v_5, v_3$  yang akan disatukan seperti

proses pernikahan di mana gen-gen mereka saling bertukar. Ini terlihat dari hasil *crossover* berikut:  
*Crossover 1* :  $v_2, v_4$



A1	A14	A31	A32	A36	A2	A5	A7	A6	A4	A3	A1
----	-----	-----	-----	-----	----	----	----	----	----	----	----

Anak

Crossover 2 :  $v_5, v_3$

A1	A14	A31	A32	A36	A2	A5	A7	A6	A4	A3	A1
----	-----	-----	-----	-----	----	----	----	----	----	----	----

A1	A35	A36	A5	A6	A4	A3	A38	A37	A30	A29	A1
----	-----	-----	----	----	----	----	-----	-----	-----	-----	----

Induk



A1	A14	A31	A32	A36	A2	A5	A7	A6	A4	A3	A1
----	-----	-----	-----	-----	----	----	----	----	----	----	----

A1	A35	A36	A5	A6	A4	A3	A38	A37	A30	A29	A1
----	-----	-----	----	----	----	----	-----	-----	-----	-----	----

Anak

5) Mutasi

Proses mutasi merupakan tahap yang penting dalam algoritma genetika di mana individu baru dihasilkan dengan memodifikasi gen-gen individu yang ada. mutasi ini terjadi secara acak dan jarang, mirip dengan kejadian mutasi genetik dalam kehidupan nyata. Probabilitas mutasi gen biasanya sangat rendah, karena mencerminkan kejadian alami yang memungkinkan terjadinya mutasi gen tetapi tetap memungkinkan variasi genetik.

Setelah melakukan proses *crossover* didapatkan keturunan-keturunan yang akan di proses mutasi. Proses mutasi yang digunakan yaitu mutasi untuk optimasi *kombinatorial order based mutation*, *position based mutation*, dan *scramble mutation*[11]. Berikut mutasi yang diperoleh :

1. Anak 1 ( $v_6$ )

A1	A20	A30	A37	A5	A2	A32	A7	A6	A4	A28	A1
----	-----	-----	-----	----	----	-----	----	----	----	-----	----

Dengan mutasi yang berurutan penulis memindahkan titik A20 dengan A30 :

A1	A30	A20	A37	A5	A2	A32	A7	A6	A4	A28	A1
----	-----	-----	-----	----	----	-----	----	----	----	-----	----

Sehingga titik A20 dan A30 bertukar posisi menjadi :

A1	A30	A20	A37	A5	A2	A32	A7	A6	A4	A28	A1
----	-----	-----	-----	----	----	-----	----	----	----	-----	----

2. Anak 2 ( $v_7$ )

A1	A30	A8	A36	A23	A5	A32	A31	A27	A22	A21	A1
----	-----	----	-----	-----	----	-----	-----	-----	-----	-----	----

Dengan mutasi yang berurutan penulis memindahkan titik A36 dengan A21 :

A1	A30	A8	A36	A23	A5	A32	A31	A27	A22	A21	A1
----	-----	----	-----	-----	----	-----	-----	-----	-----	-----	----

Sehingga titik A36 dan A21 bertukar posisi menjadi :

A1	A30	A8	A21	A23	A5	A32	A31	A27	A22	A36	A1
----	-----	----	-----	-----	----	-----	-----	-----	-----	-----	----

3. Anak 1 ( $v_8$ )

A1	A14	A31	A32	A36	A2	A3	A38	A37	A30	A29	A1
----	-----	-----	-----	-----	----	----	-----	-----	-----	-----	----

Dengan mutasi yang berurutan penulis memindahkan titik A14 dengan A30 :

A1	A14	A31	A32	A36	A2	A3	A38	A37	A30	A29	A1
----	-----	-----	-----	-----	----	----	-----	-----	-----	-----	----

Sehingga titik A14 dan A30 bertukar posisi menjadi :

A1	A30	A31	A32	A36	A2	A3	A38	A37	A14	A29	A1
----	-----	-----	-----	-----	----	----	-----	-----	-----	-----	----

4. Anak 1 ( $v_9$ )

A1	A35	A36	A5	A6	A4	A5	A7	A6	A4	A3	A1
----	-----	-----	----	----	----	----	----	----	----	----	----

Dengan mutasi yang berurutan penulis memindahkan titik A36 dengan A7 :

A1	A35	A36	A5	A6	A4	A5	A7	A6	A4	A3	A1
----	-----	-----	----	----	----	----	----	----	----	----	----

Sehingga titik A36 dan A7 bertukar posisi menjadi :

A1	A35	A7	A5	A6	A4	A5	A36	A6	A4	A3	A1
----	-----	----	----	----	----	----	-----	----	----	----	----

IV. HASIL DAN PEMBAHASAN

Penelitian ini mengkaji tentang Algoritma Genetika yang efektif dalam menemukan solusi rute yang mendekati optimal bagi masalah *Travelling Salesman Problem (TSP)* antara Surabaya dan wilayah di Jawa Timur. Melalui proses seleksi, *crossover*, dan mutasi berhasil menghasilkan individu -

individu dengan nilai *fitness* yang lebih tinggi, yaitu dengan rute yang lebih efisien dan lebih pendek[12]. Simulasi dan evaluasi hasil menemukan bahwa efisiensi rute dengan cara ini dapat signifikan lebih nyata daripada menggunakan algoritma pencarian rute biasa. Oleh karena itu, aplikasi dari Algoritma Genetika untuk menyelesaikan TSP menunjukkan manfaat besar dan potensi optimalisasi untuk menghemat sumber daya, yaitu waktu dan bahan bakar. ini membenarkan penerapan Algoritma Genetika dalam solusi praktis dari masalah optimasi yang rumit[13].

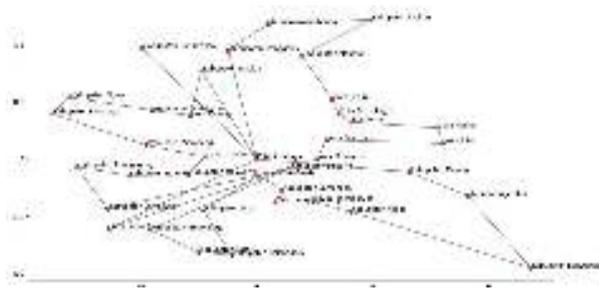
Peneliti memperoleh data nama kota tujuan dan jarak yang ditentukan dari kota asal yaitu Kota Surabaya, kemudian dilakukan proses pencarian rute dengan jarak terdekat berdasarkan data dari dataset[11]. Jarak dari Kota Surabaya ke kota tujuan dalam satuan km sebagai berikut:

1. Kota Surabaya ke Kabupaten Gresik = 28.5 km
2. Kota Surabaya ke Kabupaten Sidoarjo = 32.7 km
3. Kota Surabaya ke Kabupaten Bangkalan = 44.8 km
4. Kota Surabaya ke Kota Mojokerto = 51.9 km
5. Kota Surabaya ke Kabupaten Lamongan = 58.0 km
6. Kota Surabaya ke Kabupaten Mojokerto = 64.4 km
7. Kota Surabaya ke Kota Pasuruan = 69.3 km
8. Kota Surabaya ke Kabupaten Jombang = 76.9 km
9. Kota Surabaya ke Kabupaten Pasuruan = 84.9 km
10. Kota Surabaya ke Kota Malang = 95.3 km
11. Kota Surabaya ke Kabupaten Sampang = 98.0 km
12. Kota Surabaya ke Kabupaten Tuban = 101.0 km
13. Kota Surabaya ke Kota Batu = 104.0 km
14. Kota Surabaya ke Kota Probolinggo = 107.0 km
15. Kota Surabaya ke Kabupaten Nganjuk = 114.0 km
16. Kota Surabaya ke Kabupaten Kediri = 122.0 km
17. Kota Surabaya ke Kota Kediri = 124.0 km
18. Kota Surabaya ke Kabupaten Pamekasan = 124.0 km
19. Kota Surabaya ke Kabupaten Malang = 133.0 km
20. Kota Surabaya ke Kabupaten Probolinggo = 154.0 km
21. Kota Surabaya ke Kabupaten Tulungagung = 155.0 km
22. Kota Surabaya ke Kabupaten Lumajang = 155.0 km
23. Kota Surabaya ke Kabupaten Madiun = 162.0 km
24. Kota Surabaya ke Kota Blitar = 163.0 km
25. Kota Surabaya ke Kota Madiun = 165.0 km
26. Kota Surabaya ke Kabupaten Bojonegoro = 169.0 km
27. Kota Surabaya ke Kabupaten Blitar = 170.0 km

- 28. Kota Surabaya ke Kabupaten Sumenep = 176.0 km
- 29. Kota Surabaya ke Kabupaten Magetan = 185.0 km
- 30. Kota Surabaya ke Kabupaten Trenggalek = 192.0 km
- 31. Kota Surabaya ke Kabupaten Ponorogo = 193.0 km
- 32. Kota Surabaya ke Kabupaten Jember = 193.0 km
- 33. Kota Surabaya ke Kabupaten Ngawi = 194.0 km
- 34. Kota Surabaya ke Kabupaten Bondowoso = 218.0 km
- 35. Kota Surabaya ke Kabupaten Situbondo = 237.0 km
- 36. Kota Surabaya ke Kabupaten Pacitan = 263.0 km
- 37. Kota Surabaya ke Kabupaten Banyuwangi = 303.0 km

Dari data jarak yang ditemukan dari dataset, peneliti memperoleh hasil pencarian rute dengan jarak terdekat dari Kota Surabaya yang cukup akurat dari berbagai perhitungan yang sudah peneliti lakukan. Penggunaan perhitungan rute dengan jarak yang terdekat sesuai dengan kota asal dapat menghemat waktu dan biaya dalam perjalanan antar kota/kabupaten yang dituju. Hasil pencarian solusi yang optimal dengan menggunakan metode Algoritma Genetika dilakukan secara otomatis menggunakan program Python, sampai perhitungan yang dilakukan mendapatkan hasil yang sama dan dianggap sudah cukup optimal untuk digunakan.

Solusi optimal dari hasil permasalahan TSP dengan menggunakan Algoritma Genetika menghasilkan rute terbaik untuk melakukan perjalanan antar kota/kabupaten[11]. Rute terbaik dari Kota Surabaya sebagai kota asalnya adalah Kota Surabaya - Kabupaten Bondowoso - Kota Surabaya - Kabupaten Sumenep - Kota Mojokerto - Kota Pasuruan - Kota Blitar - Kota Madiun - Kota Batu - Kota Probolinggo - Kota Kediri - Kabupaten Magetan - Kabupaten Pacitan - Kabupaten Situbondo - Kabupaten Trenggalek - Kota Surabaya - Kabupaten Madiun - Kota Malang - Kabupaten pamekasan - Kabupaten Ngawi - Kabupaten Ponorogo - Kabupaten Sampang - Kabupaten Bangkalan - Kabupaten Mojokerto - Kabupaten Nganjuk - Kabupaten Tulungagung - Kabupaten Lumajang - Kabupaten Sidoarjo - Kabupaten Probolinggo - Kabupaten Blitar - Kabupaten Bojonegoro - Kabupaten Tuban - Kabupaten Gresik - Kabupaten Lamongan - Kabupaten Jombang - Kabupaten Pasuruan - Kabupaten Kediri - Kabupaten Banyuwangi - Kabupaten Jember - Kabupaten Malang - Kota Surabaya - Kota Surabaya.



Gambar 2. Draw Map Rute Terpendek dari Kota Surabaya ke Wilayah Jawa Timur

Berdasarkan hasil pencarian solusi rute terpendek dalam perjalanan di wilayah Jawa Timur dengan Kota Surabaya menjadi kota asalnya dengan didapatkan 1 solusi yang paling baik dibandingkan dengan yang lain.

Algoritma Genetika memiliki kelebihan dan kelemahan, yaitu waktu komputasi yang stabil dan memberikan hasil rute

terpendek tetapi membutuhkan waktu yang banyak meskipun dengan jumlah kota yang banyak. Sedangkan kelemahan dari Algoritma Genetika adalah terletak pada ukuran populasi, besar generasi, ukuran peluang *crossover* dan mutasi[11]. Peneliti melakukan beberapa siklus untuk mencari jarak terdekat dan ternyata sebagian besar hasil pencarian mendapatkan kesimpulan yang sama, maka peneliti beranggapan bahwa siklus yang didapatkan sudah cukup untuk menentukan rute terpendek dikarenakan sifat dari Algoritma Genetika adalah random sehingga tidak diketahui kapan akan mendapatkan kesimpulan hasil pada generasi dan populasi yang optimal.

## V. PENUTUP

### A. Kesimpulan

Penelitian ini menggunakan penerapan algoritma genetika untuk menemukan solusi yang optimal dalam mencari jalur terpendek antara Surabaya dan daerah-daerah di Provinsi Jawa Timur. Dengan menggunakan metode Travelling Salesman Problem (TSP), algoritma genetika berhasil mengidentifikasi rute terbaik yang tidak hanya memperbaiki efisiensi dalam perjalanan, tetapi juga dapat mengurangi biaya. Hasil uji coba yang telah dilakukan menunjukkan keberhasilan dalam penggunaan algoritma genetika untuk menangani permasalahan ini, dengan memperhitungkan sejumlah faktor seperti ukuran populasi, probabilitas *crossover*, dan probabilitas mutasi. Penelitian ini dapat menjadi landasan untuk mengembangkan sistem navigasi yang lebih canggih dan aplikatif untuk menyelesaikan tantangan optimasi lainnya.

Berdasarkan penelitian yang telah dilakukan, didapatkan rute terbaik dari Kota Surabaya sebagai kota asal dengan melewati berbagai rute kota lain yang berada pada Provinsi Jawa Timur, telah terpilih dan didapatkan total jarak terpendek dari perjalanan pada kasus di penelitian kali ini, yaitu didapatkan 2335.3049828354365 km.

### B. Saran

Penelitian ini dapat diperluas dengan mempertimbangkan faktor lain seperti kondisi lalu lintas, waktu tempuh aktual, hingga kondisi cuaca. Implementasi algoritma genetika dalam sistem navigasi perlu diuji coba lebih luas sebagai bahan validasi. Penelitian selanjutnya diharapkan dapat membandingkan kinerja algoritma genetika dengan pendekatan lain. Analisis lebih lanjut tentang dampak penggunaan algoritma genetika dalam meningkatkan efisiensi transportasi dan pengurangan biaya, termasuk evaluasi dampak lingkungan dan ekonomi yang juga diperlukan. Dengan langkah-langkah ini, pengembangan algoritma genetika dalam pencarian jalur terpendek dapat signifikan dalam meningkatkan sistem transportasi di Jawa Timur dan di wilayah lainnya.

## UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada dosen pembimbing atas arahan dan dukungannya, serta kepada rekan-rekan dan keluarga atas dukungan moralnya. Terima kasih juga kepada Tim SANTIKA sebagaimana dengan

sukarela menyediakan template untuk paper atau jurnal ini. Tidak luput juga kepada para penulis yang telah berpartisipasi dalam penulisan jurnal ini. Tanpa upaya keras dan dedikasi kalian, jurnal ini tidak akan selesai tepat waktu. Semoga hasil penelitian ini bermanfaat bagi para peneliti, pembuat kebijakan, dan bisnis untuk membuat keputusan berdasarkan informasi terkait konektivitas dan mobilitas regional di provinsi Jawa Timur.

#### REFERENSI

- [1] Badan Pemeriksa Keuangan Republik Indonesia. Kota Surabaya. Diakses pada 3 Juni 2024, dari <https://jatim.bpk.go.id/kota-surabaya/>.
- [2] Utomo, R. G., Maylawati, D. S., Alam, C. N., (2018). Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP). *JOIN (Jurnal Online Informatika)*, 3 (1), 61-67.
- [3] Syamsudin, Aries., (2004). Pengenalan Algoritma Genetik. ([IlmuKomputer.com](http://IlmuKomputer.com))
- [4] Simanjuntak, O.S., Seminar Nasional Informatika 2012 (semnasIF 2012). In : UPN “Veteran”, Pengembangan Shortest Path Algorithm (SPA) Dalam Rangka Pencarian Lintasan Terpendek Pada Graf Bersambung Berarah Terurai, Yogyakarta, 30 Juni 2012. UPN “Veteran”: Yogyakarta.
- [5] "History of the TSP." Diakses pada Juni 2024, dari <http://www.math.uwaterloo.ca/tsp/history/>.
- [6] Amri, Faisal., Nababan, Erna Budhiarti., dan Syahputra, Mohammad Fadly. 2012. Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem. *Jurnal Dunia Teknologi Informasi*, Vol. 1, No. 1, pp. 8-13\
- [7] Manggolo, Inu., Marzuki, Marza Ihsan., dan Alaydrus, Mudrik. 2011. Optimalisasi Perencanaan Jaringan Akses Serat Optik Fiber To The Home Menggunakan Algoritma Genetika. *InComTech, Jurnal Telekomunikasi dan Komputer*, Vol. 2, No. 2, pp. 21-36.
- [8] Salaki, D. T., (2011). Penentuan Lintasan Terpendek Dari FMIPA ke Rektorat dan Fakultas di UNSRAT manado Menggunakan Algoritma Dijkstra. *Jurnal Ilmiah Sains*. 11(1), 73-78
- [9] Salaki, D. T., (2011). Penentuan Lintasan Terpendek Dari FMIPA ke Rektorat dan Fakultas di UNSRAT manado Menggunakan Algoritma Dijkstra. *Jurnal Ilmiah Sains*. 11(1), 73-78
- [10] W. F. Mahmudy, "Algoritma evolusi," Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang, pp. 1-101, 2013.
- [11] A. Tohari and Y. P. Astuti, "Penerapan algoritma genetika dalam menentukan rute terpendek PT. Pos Cabang Lamongan," *MATHunesa: Jurnal Ilmiah Matematika*, vol. 11, no. 03, pp. 458-467, 2023.
- [12] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran," *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, pp. 329-336, 2022.
- [13] D. Gunawan, D. Riana, D. Ardiansyah, F. Akbar, and S. Alfarizi, "Komparasi Algoritma Support Vector Machine dan Naïve Bayes dengan Algoritma Genetika pada Analisis Sentimen Calon Gubernur Jabar 2018-2023," 2020.
- [14] R. Hartono and A. Zein, "Penerapan Algoritma Genetika dan Jaringan Syaraf Tiruan dalam Penjadwalan Mata Kuliah," *Jurnal Ilmu Komputer JIK*, vol. VI, no., 2023.
- [15] I. W. Supriana, M. A. Raharja, I. M. S. Bimantara, and D. Bramantya, "Implementasi dua model crossover pada algoritma genetika untuk optimasi penggunaan ruang perkuliahan," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 4, no. 2, pp. 167-177, 2021.