

## IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMALISASI RUTE PENGIRIMAN PESANAN DI RESTO PAK LANJAR SLEMAN

Anang Hidayat<sup>1</sup>, Herdiesel Santoso<sup>2\*</sup>

<sup>1</sup> Program Studi Informatika STMIK EL RAHMA Yogyakarta, <sup>2</sup> Program Studi Sistem Informasi STMIK EL RAHMA Yogyakarta, \*Penulis Koresponden  
e-mail:<sup>1</sup>ananghidayat101001@gmail.com,<sup>2</sup>herdiesel.santoso@stmikelrahma.ac.id

### ABSTRACT

Multi-destination travel is one of the problems in the Traveling Salesman Problem (TSP), which has a large problem space if solved combinatorially. This study aims to design and implement a Genetic Algorithm model to provide route recommendations for order deliveries at Resto Pak Lanjar, Sleman. The proposed Genetic Algorithm takes into account both symmetric and asymmetric distances. The route recommendations consider not only the distance but also the travel time, which is obtained using Google Maps API. The encoding scheme uses permutation encoding, parent selection is done through roulette-wheel selection, with order crossover as the crossover method and swap mutation as the mutation method. The algorithm also ensures that the best individual from a given generation is not lost during the evolutionary process. This study fills the gap in the literature, especially in applying Genetic Algorithms for route optimization in small restaurants by considering both time and distance factors. The experimental results show that for fewer than 8 objects, the optimal population size consists of 30 individuals, while for more than 8 objects, the optimal population size consists of 180 individuals. The stopping criterion is set when the highest fitness value remains unchanged for 30 consecutive generations. The optimal combination of crossover and mutation probabilities is {0.5:0.5}.

**Keywords:** Genetic Algorithm, Google Maps API, Order Delivery, Route Optimization, Traveling Salesman Problem

### INTISARI

Perjalanan multi destinasi merupakan salah satu permasalahan dalam Traveling Salesman Problem (TSP) yang memiliki ruang masalah sangat besar jika diselesaikan secara kombinatorial. Penelitian ini bertujuan untuk merancang dan menerapkan model Algoritma Genetika guna memberikan rekomendasi rute pengiriman pesanan di Resto Pak Lanjar, Sleman. Algoritma Genetika yang diusulkan mampu memperhitungkan jarak simetris maupun asimetris. Rekomendasi rute tidak hanya mempertimbangkan jarak, tetapi juga waktu tempuh yang diperoleh dari Google Maps menggunakan Google Maps API. Skema pengkodean menggunakan permutation encoding, seleksi orang tua menggunakan roulette-wheel selection, dengan metode crossover order crossover dan metode mutasi swap mutation. Algoritma ini juga menjaga agar individu terbaik dari generasi tertentu tidak hilang selama proses evolusi. Penelitian ini mengisi gap yang ada dalam literatur, terutama pada aplikasi Algoritma Genetika untuk optimasi rute pengiriman di restoran kecil dengan mempertimbangkan faktor waktu dan jarak. Hasil pengujian menunjukkan bahwa untuk objek kurang dari 8, individu terbaik terdiri dari 30 individu dalam satu populasi, sementara untuk objek lebih dari 8, individu terbaik terdiri dari 180 individu dalam satu populasi. Kriteria berhenti ditetapkan jika selama 30 generasi berturut-turut nilai fitness tertinggi tidak berubah. Kombinasi probabilitas crossover dan mutasi yang paling optimal adalah {0.5:0.5}.

**Kata kunci:** Algoritma Genetika, Optimasi Rute, Pengiriman Pesanan, Traveling Salesman Problem, Google Maps API.

### 1. PENDAHULUAN

Algoritma Genetika adalah metode pencarian solusi berbasis evolusi yang meniru proses seleksi alam. Dalam konteks penelitian ini, Algoritma Genetika digunakan untuk mencari rute pengiriman pesanan yang optimal, dengan mempertimbangkan berbagai parameter seperti jarak, waktu, dan biaya. Dalam dunia bisnis modern, optimalisasi rute pengiriman menjadi salah satu aspek yang sangat penting untuk meningkatkan efisiensi operasional dan kepuasan pelanggan. Restoran tidak hanya berfungsi sebagai tempat makan tetapi juga sebagai entitas bisnis yang harus efisien dalam pengelolaan operasionalnya. Kecepatan dan efisiensi dalam pengiriman

pesanan menjadi salah satu faktor kunci yang mempengaruhi kepuasan pelanggan (Sofia Tussoliha, 2019) Penggunaan algoritma ini memungkinkan perusahaan untuk menentukan rute terbaik, mengurangi biaya operasional, dan memastikan pengiriman tepat waktu. Salah satu penerapan algoritma genetika adalah dalam masalah optimasi kombinasi, yaitu untuk mendapatkan solusi optimal dari suatu masalah yang memiliki banyak kemungkinan solusi (Denny Hermawanto, 2003).

Masalah *Traveling Sales Problem* (TSP) adalah persoalan optimasi yang bertujuan menemukan rute terpendek yang harus ditempuh oleh seorang *salesman*. Dalam masalah ini, *salesman* harus mengunjungi sejumlah kota tepat satu kali dan kembali ke kota asal dengan total biaya perjalanan yang seminimal mungkin. Biaya perjalanan ini biasanya diukur dalam bentuk jarak antar kota. TSP memiliki ruang solusi yang sangat besar, di mana jumlah kemungkinan jalur dapat dihitung dengan menggunakan rumus permutasi. Algoritma genetika terdiri dari beberapa komponen. Setiap komponen memiliki metode yang sangat bervariasi yang telah diusulkan untuk mengimplementasikan masing-masing komponen tersebut. Masing-masing metode mempunyai kelebihan dan kekurangan. Suatu metode yang bagus untuk menyelesaikan suatu masalah, belum tentu bagus untuk masalah lain, atau bahkan tidak bisa digunakan untuk masalah lain lagi (Suyanto, 2011).

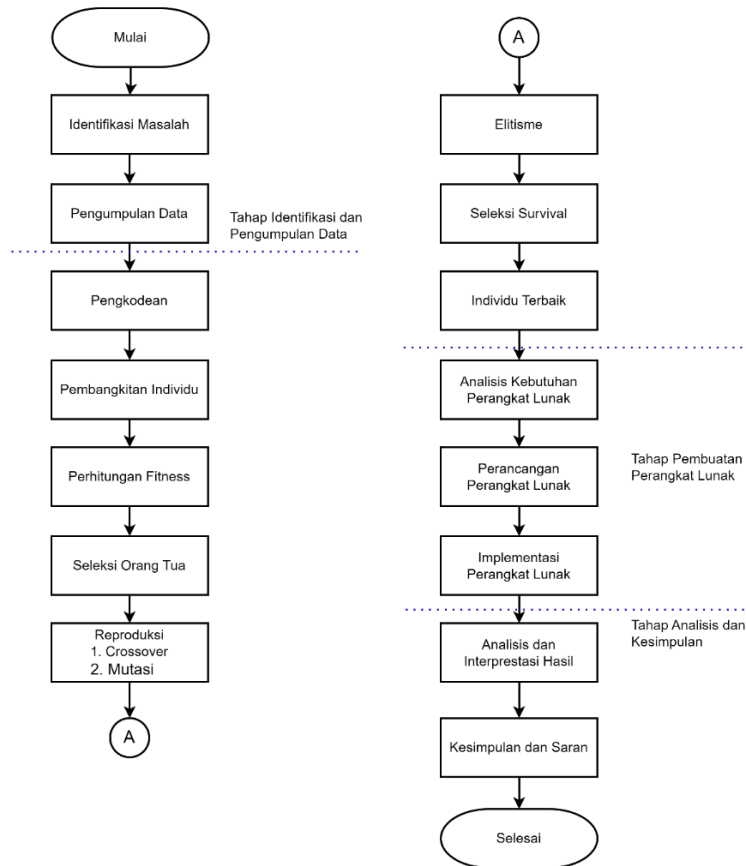
Penelitian yang menerapkan Algoritma genetika telah banyak dilakukan dengan menerapkan berbagai macam kombinasi parameter. Penelitian Santoso & Sanuri (2019), menerapkan algoritma genetika untuk menyelesaikan permasalahan perjalanan divisi pemasaran STMIK El Rahma. Tantangan yang dihadapi adalah kompleksitas dalam penjadwalan rute kunjungan ke sekolah-sekolah untuk kegiatan promosi, yang melibatkan perjalanan multi-destinasi dengan mempertimbangkan waktu kunjungan (TSP-TW). Untuk mengatasi tantangan ini, algoritma genetika diterapkan guna menemukan rute perjalanan optimal. Rekomendasi rute mempertimbangkan jarak antar lokasi serta waktu tempuh yang diperoleh dari *Google Maps API*. Pengujian dilakukan dengan berbagai skenario, meliputi variasi generasi, populasi, dan kombinasi probabilitas *crossover* dan mutasi. Hasil menunjukkan bahwa populasi 150 individu memberikan hasil terbaik, dengan kriteria berhenti jika nilai fitness tertinggi konstan selama 127 generasi berturut-turut. Kombinasi probabilitas *crossover* dan mutasi optimal yang ditemukan adalah {0.3: 0.7}, membuktikan efektivitas Algoritma Genetika dalam menyelesaikan masalah TSP-TW ini. Wulandari & Helmi (2019), menyelesaikan masalah pendistribusian surat kabar oleh beberapa salesman pada suatu agen, yang membutuhkan minimasi rute dari depot agen ke alamat pelanggan untuk pengantaran tepat waktu menggunakan Algoritma Genetika (AG). Dengan menggunakan AG, yang mengikuti tahapan operasi genetik dan seleksi alam, dilakukan pembentukan populasi awal, penentuan nilai *fitness*, seleksi individu berdasarkan nilai *fitness*, serta penerapan operasi genetik seperti *crossover* dan mutasi untuk menghasilkan individu baru dengan rute terbaik. Hasil simulasi menunjukkan bahwa dengan probabilitas *crossover* 0,5 dan probabilitas mutasi 0,01, rute terpendek sepanjang 64,03 km berhasil ditemukan pada generasi pertama, menandakan efisiensi dalam distribusi surat kabar dan pengiriman yang tepat waktu.

Widayani et al., (2023), membahas tentang inovasi dalam layanan laundry, seperti optimasi layanan penjemputan *laundry* dengan pemesanan waktu pengiriman dan penerimaan, menjadi fokus utama dalam bisnis *laundry* karena potensi keuntungan yang besar. Algoritma Genetika (*Genetic Algorithm*) digunakan untuk mengatasi permasalahan *Traveling Salesman Problem* (TSP) dalam menentukan rute terpendek dan waktu kedatangan yang tepat untuk setiap pelanggan. Proses pengoptimalan ini menggabungkan kromosom (solusi) melalui Algoritma Genetika dengan menggunakan tahapan seleksi, *crossover*, dan mutasi. Dalam penelitian ini, kombinasi terbaik dari *crossover* dan mutasi, serta ukuran populasi dan generasi, diuji untuk mencari solusi optimal. Hasil pengujian menunjukkan bahwa nilai optimal yang dicapai adalah 2000 dengan probabilitas *crossover* terbaik sebesar 0,4 dan mutasi sebesar 0,6. Hal ini membuktikan bahwa penggunaan Algoritma Genetika dapat memberikan solusi yang efektif dalam melayani pelanggan dengan jendela waktu yang ditentukan. Penelitian Pradita & Arifin (2023), menerapkan *Google Maps* dalam aplikasi modern memainkan peran penting, terutama dalam mengembangkan fitur pemetaan seperti Sistem Informasi Geografis (SIG). SIG dapat dimanfaatkan untuk menyediakan informasi lokasi serta mendukung proses pengambilan keputusan, seperti pada kasus penjualan sayur segar di Kota Mojokerto. Dengan SIG, penjual dapat mengunggah produk secara daring sehingga pembeli dapat dengan mudah menemukan dan melihat lokasi penjual sayur terdekat. Pengujian sistem menggunakan metode *blackbox* menunjukkan tingkat keberhasilan sebesar 85,45%, yang menunjukkan bahwa sistem ini efektif dan efisien dalam mendukung interaksi online antara penjual dan pembeli.

Meskipun banyak penelitian telah menggunakan algoritma untuk optimasi rute pengiriman, sebagian besar masih terbatas pada aplikasi transportasi umum atau perusahaan besar. Belum banyak penelitian yang mengimplementasikan Algoritma Genetika untuk mengoptimalkan rute pengiriman pada restoran dengan mempertimbangkan faktor waktu, biaya, dan variabel lingkungan yang dinamis, seperti kondisi lalu lintas yang didapatkan dari API *Google Maps*. Penelitian ini bertujuan untuk menerapkan Algoritma Genetika guna mengoptimalkan rute pengiriman pesanan pada restoran dengan mempertimbangkan efisiensi biaya dan waktu.

Metode seleksi orang tua yang digunakan *roulette-wheel selection*, metode *crossover* menggunakan *order crossover* dan metode mutasi menggunakan *swap-mutation*. Selanjutnya dilakukan berbagai pengujian seperti pengujian generasi, pengujian populasi, kombinasi nilai *crossover* dan mutasi untuk mendapatkan parameter algoritma genetika yang optimal.

## 2. METODE PENELITIAN



**Gambar 1.** Metode Penelitian

Tahapan penelitian ini terdiri dari 4 tahapan utama yaitu pengumpulan data, pengolahan data, pembuatan perangkat lunak serta analisis dan interpretasi hasil. Untuk penjelasan tahapan adalah sebagai berikut:

### A. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data lokasi yang sering menjadi tujuan distribusi pemesanan makanan di Resto Pak Lanjar. Data tersebut diperoleh melalui wawancara dengan pihak Resto, kemudian diverifikasi melalui *Google Maps*, dan diambil datanya menggunakan *Google Maps API*.

Penelitian ini menggunakan 6 objek dengan 5 objek sebagai sampel lokasi tujuan dan Resto Pak Lanjar yang menjadi lokasi awal perjalanan.

**Tabel 1.** Objek Penelitian

Kode	Nama Objek
1	Resto Pak Lanjar
2	Kantor BKAD Sleman
3	Samsat Sleman
4	Bank BPD DIY cab. Sleman
5	Puskesmas Ngaglik 2 Sleman
6	Dukcapil Sleman

Data perbandingan jarak 6 objek disajikan dalam Tabel 2, sedangkan data perbandingan waktu tempuh setiap objek disajikan pada Tabel 3. Pada Tabel tersebut terlihat jarak dan waktu tempuh dari Resto Pak Lanjar ke Samsat Sleman dan dari Samsat Sleman ke Resto Pak Lanjar tidak sama karena ada penerapan jalur 1 arah maka inilah yang disebut dengan *asimetris*.

#### B. Pengolahan Data

Pengolahan data pada penelitian ini bertujuan untuk mencari rute paling optimal menggunakan Algoritma Genetika. Adapun langkah-langkahnya adalah sebagai berikut (Santoso & Sanuri, 2019).

##### 1. Pengkodean

Pada Algoritma Genetika, permasalahan terlebih dahulu perlu dikonversi menjadi individu yang direpresentasikan oleh satu atau lebih kromosom dengan skema pengkodean tertentu. Dalam konteks masalah perjalanan atau kunjungan, solusi yang dicari adalah susunan objek yang harus dikunjungi secara berurutan. Salah satu skema pengkodean yang sering digunakan adalah *permutation encoding*, di mana setiap objek digambarkan dalam bentuk bilangan bulat tipe integer. Setiap individu memiliki jumlah gen yang sama dengan jumlah objek destinasi yang dipilih dalam perjalanan.

##### 2. Inisiasi Populasi

Inisiasi adalah proses pembangkitan sejumlah individu secara acak atau melalui prosedur tertentu. Dengan memanfaatkan fungsi acak, setiap individu dibentuk dengan memilih secara acak bilangan antara 1 hingga jumlah gen yang diperlukan, hingga terbentuk satu kromosom. Kromosom-kromosom yang sudah dibangkitkan ini kemudian akan membentuk suatu populasi.

##### 3. Fungsi evaluasi

Setelah populasi terbentuk, tahap berikutnya adalah menghitung nilai *fitness* untuk setiap individu. Karena permasalahan ini adalah *Traveling Salesman Problem* (TSP) yang bertujuan meminimalkan total biaya perjalanan, fungsi *fitness* yang digunakan adalah 1 dibagi dengan total biaya ditambah pinalti.

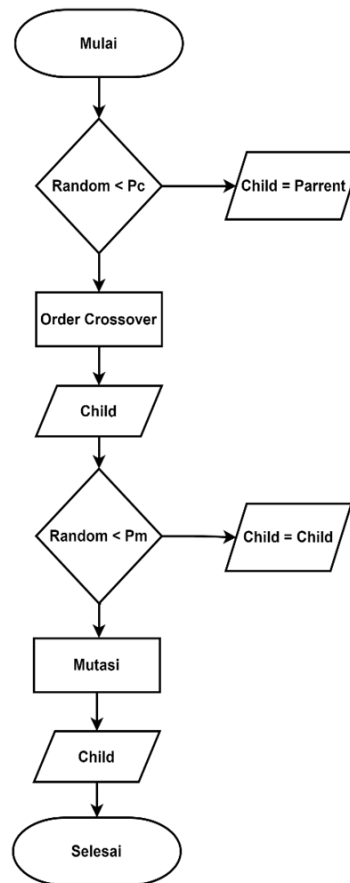
##### 4. Seleksi orang tua

Proses seleksi yang digunakan dalam penelitian ini adalah *roulette-wheel selection*, yang mudah diimplementasikan dalam pemrograman. Berikut adalah langkah-langkahnya:

- Hitung nilai *fitness* setiap individu ( $f_i$ ), di mana  $i$  adalah urutan individu dari 1 hingga  $n$ .
- Hitung total *fitness* dari semua individu dalam populasi.
- Tentukan probabilitas setiap individu dengan membagi nilai *fitness* masing-masing individu dengan total *fitness* populasi.
- Gunakan  $P_s[i]$  untuk menyatakan probabilitas individu ke- $i$  dan  $f(i)$  untuk nilai *fitness* individu ke- $i$ , di mana  $i = 1, 2, 3, \dots, n$ .
- Hitung probabilitas kumulatif setiap individu.
- Hasilkan bilangan acak  $R \in [0,1]$ , sesuai dengan jumlah populasi pada generasi tersebut.
- Pemilihan individu sebagai induk ditentukan berdasarkan nilai bilangan acak ( $R[i]$ ) yang dihasilkan. Jika bilangan acak  $R[i]$  kurang dari nilai kumulatif *fitness* pertama ( $q[1]$ ), maka kromosom pertama akan terpilih sebagai induk dan menggantikan kromosom ke- $i$ . Sebaliknya, jika  $R[i]$  memenuhi kondisi ( $q[i-1] < R[i] < q[i]$ ), maka kromosom ke- $i$  akan dipilih sebagai induk. Misalnya, jika  $R[1]$  kurang dari  $q[4]$ , maka kromosom pertama ( $K[1]$ ) akan digantikan oleh kromosom ke-4 ( $K[4]$ ) dalam populasi awal.

##### 5. Reproduksi

Penelitian ini menggunakan metode *order crossover* untuk representasi permutasi (Cia, 2024). Proses selanjutnya adalah mutasi menggunakan skema *swap mutation*. Proses *crossover* dan mutasi mengikuti skema pada Gambar 2.



**Gambar 2.** Skema Crossover dan Mutasi

**Tabel 2.** Data Jarak Tempuh Antar Objek (Km)

Kode	1	2	3	4	5	6
1	0	4.3	5.5	3.8	1.9	4.4
2	4.3	0	2.8	1.6	6.2	0.2
3	6.3	3.6	0	2.5	8.2	3.8
4	3.7	1.1	2.3	0	5.7	1.3
5	1.9	6.5	7.7	5.9	0	6.7
6	4.5	0.2	2.9	1.8	6.4	0

**Tabel 3.** Data Waktu Tempuh Antar Objek (Menit)

Kode	1	2	3	4	5	6
1	0	9.1	10.9	8.9	4	9.2
2	8.8	0	5.9	4.1	13.3	0.6
3	12.2	7.6	0	4.9	16.7	7.6
4	8.1	3.4	5.7	0	12.6	3.5
5	5	13.5	15.4	12.8	0	13.6
6	9	0.6	6.1	4.2	13.5	0

6. Elitisme  
Agar individu dengan nilai *fitness* tertinggi tetap terjaga selama proses evolusi, diperlukan skema elitisme. Individu hasil dari skema elitisme ini akan menjadi bagian dari generasi berikutnya sebagai salah satu individu baru.
7. Seleksi *survivor*  
Seleksi *survivor* akan menggantikan N individu dalam satu generasi dengan N individu baru secara sekaligus. Setiap individu dalam jumlah N tersebut harus unik, artinya tidak ada individu yang

- identik dalam satu populasi.
8. Kriteria berhenti  
Iterasi akan berhenti jika nilai *fitness* tertinggi tidak mengalami perubahan selama beberapa generasi berturut-turut. Dengan demikian, jika perhitungan dilanjutkan hingga generasi ke-N, maka diyakini bahwa nilai *fitness* terendah akan tetap tidak berubah.
- C. Pembuatan Perangkat Lunak  
Tahap berikutnya adalah mengubah spesifikasi sistem menjadi perangkat lunak dengan menulis program menggunakan bahasa pemrograman yang telah dipilih. Implementasi dilakukan menggunakan HTML, CSS, dan PHP. Selain itu, fungsi-fungsi dari *Google Maps API* digunakan untuk keperluan seperti pengenalan objek (*geocoding*), perhitungan jarak dan waktu tempuh (*distance matrix*), serta menampilkan rute atau petunjuk arah (*direction*). Untuk menjalankan fungsi-fungsi *Google Maps API*, diperlukan bahasa pemrograman *JavaScript* (Sugiyatno et al., 2023).
- D. Analisis dan Interpretasi Hasil  
Pada tahap ini, dilakukan pembahasan dan analisis terhadap model Algoritma Genetika yang telah dirancang sesuai dengan tujuan penelitian. Analisis mencakup analisis perangkat lunak, analisis jumlah generasi, analisis jumlah populasi, serta analisis kombinasi probabilitas *crossover* dan probabilitas mutasi yang optimal. Hasil dari analisis ini diinterpretasikan dengan jelas untuk mendukung penarikan kesimpulan.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Implementasi Perangkat Lunak

Perangkat lunak ini bertujuan untuk memudahkan Resto Pak Lanjar dalam mendapatkan rekomendasi rute pengiriman yang optimal. Pada halaman utama aplikasi, terdapat menu input untuk memasukkan lokasi awal pengiriman, serta menu input untuk memasukkan lokasi tujuan pengiriman. Jika lokasi tersebut dikenali oleh aplikasi, maka akan muncul informasi berupa nama lokasi, alamat, dan koordinatnya.

Implementasi Algoritma Genetika Untuk Optimalisasi Rute Pengiriman Pesanan di Resto Pak Lanjar Sleman

Lokasi Awal

Mulai dari :

Alamat :

Latitude :

Longitude :

Lokasi Tujuan

Masukan Lokasi Tujuan :

Lokasi Tujuan	Nama di Google	Alamat	Latitude	Longitude	Hapus
kantor samsat sleman	Kantor Samsat Sleman Yogyakarta	Klt: Samsat Sleman, Krappak, Tiharjo, Kec. Sleman, Kabupaten Sleman, Daerah Istimewa Yogyakarta, Indonesia	-7.6987823,	110.3467764	<input type="button" value="Delete"/>

Gambar 3. Implementasi Menu Input Lokasi Awal dan Lokasi Tujuan

Pada menu input lokasi awal, pengguna perangkat lunak memasukkan tujuan pengirim. Jika tujuan tersebut dikenali oleh aplikasi, maka akan ditampilkan nama lokasi, alamat, serta koordinat lintang dan bujur. Implementasi menu input lokasi tujuan ditunjukkan pada Gambar 3. Dengan menggunakan *Google Maps API*, perangkat lunak menghitung jarak dan waktu tempuh dari lokasi awal ke masing-masing tujuan. Selanjutnya, aplikasi melakukan perhitungan menggunakan Algoritma Genetika dengan parameter yang telah ditetapkan, yaitu waktu kedatangan, waktu ketersediaan, jarak, waktu tempuh, jumlah populasi, jumlah generasi, nilai probabilitas *crossover* ( $P_c$ ), dan probabilitas mutasi ( $P_m$ ). Hasil perhitungan ini berupa solusi rute terbaik, nilai *fitness*, serta rekomendasi rute perjalanan lengkap dengan estimasi jarak dan waktu tempuhnya.

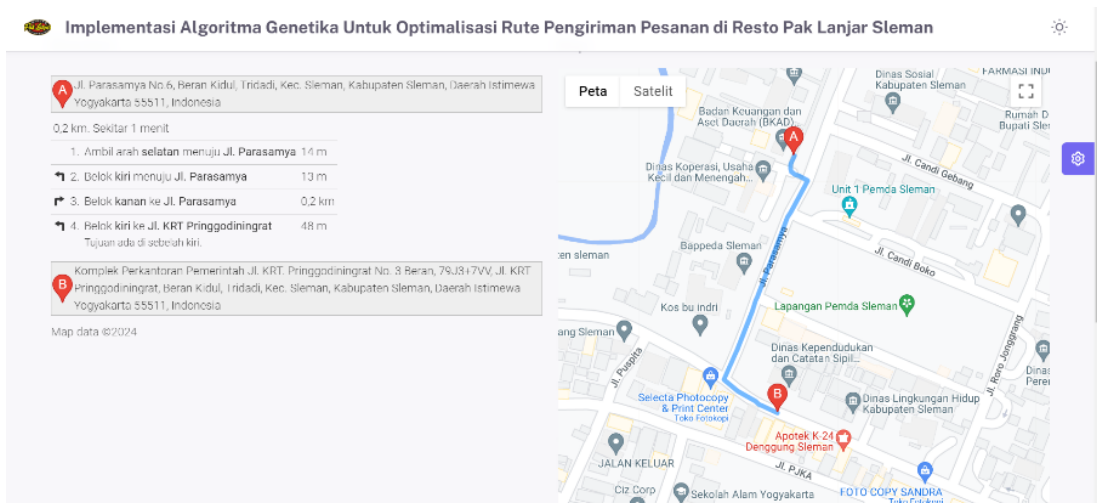
Pengguna juga dapat melihat petunjuk arah menuju lokasi tujuan. Rute yang ditampilkan merupakan rute tercepat berdasarkan kondisi lalu lintas *real-time* yang direkomendasikan dengan bantuan *Google Maps API*. Rute ini dapat berubah sesuai kondisi lalu lintas saat aplikasi digunakan. Implementasi petunjuk arah ini ditampilkan pada Gambar 5.

Solusi Terbaik Yang Ditemukan Adalah 1-3-4-2-5-6-1 dengan fitness 0.0056179775280899 dengan 31 generations.

Perjalan Ke-	Rekomendasi Tujuan	Estimasi Jarak	Estimasi Waktu	Rute
1	Warung Pak Lanjar ke kantor dukcapil sleman	4.486 Km	9.22 menit	<a href="#">Rute</a>
2	kantor dukcapil sleman ke kantorbkad sleman	2.766 Km	5.05 menit	<a href="#">Rute</a>
3	kantorbkad sleman ke kantor samsat sleman	3.12 Km	6.4 menit	<a href="#">Rute</a>
4	kantor samsat sleman ke bank bpd diy cab sleman	2.405 Km	4.17 menit	<a href="#">Rute</a>
5	bank bpd diy cab sleman ke puskesmas ngaglik 2	6.512 Km	12.6 menit	<a href="#">Rute</a>
6	puskesmas ngaglik 2 ke Warung Pak Lanjar	1.942 Km	4.97 menit	<a href="#">Rute</a>
Total		21.231 Km	42.4 menit	

© 2024

Gambar 4. Implementasi Menu Hasil Perhitungan



Gambar 5. Implementasi Menu Petunjuk Arah

### 3.2 Pengujian Generasi

Pengujian terhadap jumlah generasi dilakukan untuk menentukan nilai ambang optimal sebagai kriteria penghentian dalam Algoritma Genetika. Pengujian ini menggunakan metode seleksi elitis dengan populasi berisi 30 individu serta mengombinasikan Probabilitas *Crossover* ( $P_c$ ) dan Probabilitas Mutasi ( $P_m$ ) masing-masing sebesar 0,5 (Santoso H, 2016). Variasi jumlah gen berkisar antara 6 hingga 10, dengan lima kali percobaan untuk setiap konfigurasi gen. Jumlah generasi awal ditetapkan sebanyak 200 generasi. Selama tiap percobaan, interval perubahan nilai *fitness* maksimum dicatat sebagai nilai ambang. Sebagai contoh, dalam satu percobaan, nilai *fitness* awal adalah 0,333 pada generasi pertama. Pada generasi ke-12, nilai *fitness* meningkat menjadi 0,351, memberikan interval perubahan sebesar 7. Hingga generasi ke-200, tidak terjadi peningkatan nilai *fitness* lebih lanjut. Dengan demikian, nilai ambang yang tercatat pada akhir percobaan adalah interval perubahan nilai *fitness* maksimum, yaitu sebesar 7. Hasil percobaan pengujian generasi dapat dilihat pada Tabel 4.

Tabel 4. Pengujian Generasi

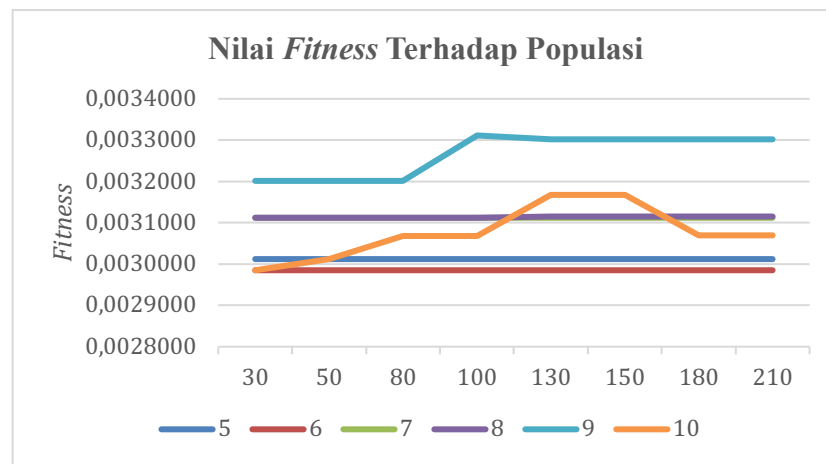
Percobaan Ke	Threshold				
	6	7	8	9	10
1	5	11	30	5	20
2	11	17	14	13	5
3	7	15	27	2	7
4	9	20	15	3	28
5	7	10	11	24	11
Maksimal	11	20	30	24	28

Jangkauan perubahan nilai *fitness* terjauh adalah 30, yang ditetapkan sebagai ambang batas optimal untuk kriteria penghentian. Jika selama 30 generasi berturut-turut nilai *fitness* tertinggi tidak mengalami perubahan, maka iterasi akan dihentikan, dan nilai *fitness* terakhir akan digunakan sebagai solusi optimal.

### 3.3 Hasil dan Analisis Pengujian Populasi

Pengujian populasi bertujuan untuk menentukan jumlah individu optimal dalam setiap generasi Algoritma Genetika, dengan Probabilitas Crossover ( $P_c$ ) dan Probabilitas Mutasi ( $P_m$ ) masing-masing 0,5. Ukuran populasi yang diuji meliputi 30, 50, 80, 100, 130, 150, 180, dan 210 individu, dengan jumlah gen bervariasi antara 5 hingga 10. Iterasi dihentikan jika setelah 30 generasi tidak ada peningkatan nilai fitness, dan nilai terakhir digunakan sebagai solusi optimal.

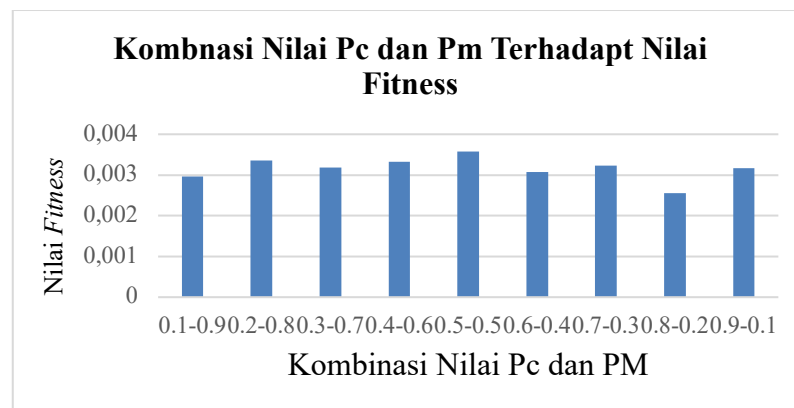
Gambar 6 menunjukkan grafik hubungan antara nilai *fitness* dan ukuran populasi. Untuk jumlah gen di bawah 8, nilai *fitness* tertinggi diperoleh pada populasi dengan 30 individu, tanpa perubahan signifikan setelahnya. Oleh karena itu, jumlah individu optimal untuk gen berjumlah 8 atau kurang adalah 30 individu. Sedangkan untuk jumlah gen lebih dari 7, nilai *fitness* tertinggi mulai bervariasi namun stabil pada populasi berjumlah 180 individu, terutama pada pengujian dengan 10 gen. Setelah ukuran populasi mencapai 180 individu, tidak terjadi perubahan signifikan dalam nilai *fitness*. Dengan demikian, untuk jumlah gen lebih dari 7, ukuran populasi optimal adalah 180 individu.



Gambar 6. Grafik Nilai *Fitness* Terhadap Populasi

### 3.4 Pengujian Probabilitas *Crossover* dan Probabilitas Mutasi

Dalam penelitian yang dilakukan oleh (Santoso & Sanuri, 2019), kombinasi nilai  $P_c$  dan  $P_m$  diuji dengan penetapan nilai dalam kelipatan 0,1, berkisar dari 0,1 hingga 0,9. Total nilai untuk kombinasi  $P_c$  dan  $P_m$  tetap 1, sehingga nilai bervariasi dari {0,1:0,9} hingga {0,9:0,1}, dengan penambahan atau pengurangan sebesar 0,1 pada setiap langkahnya. Setiap individu dalam populasi memiliki 8 gen, dengan total populasi berjumlah 180. Penelitian ini menggunakan ambang batas 30, yang artinya iterasi akan berhenti setelah 30 generasi tanpa adanya peningkatan signifikan dalam nilai fitness. Setiap eksperimen diulang sebanyak 5 kali, dan rata-rata dari hasil tersebut diambil sebagai hasil akhir.



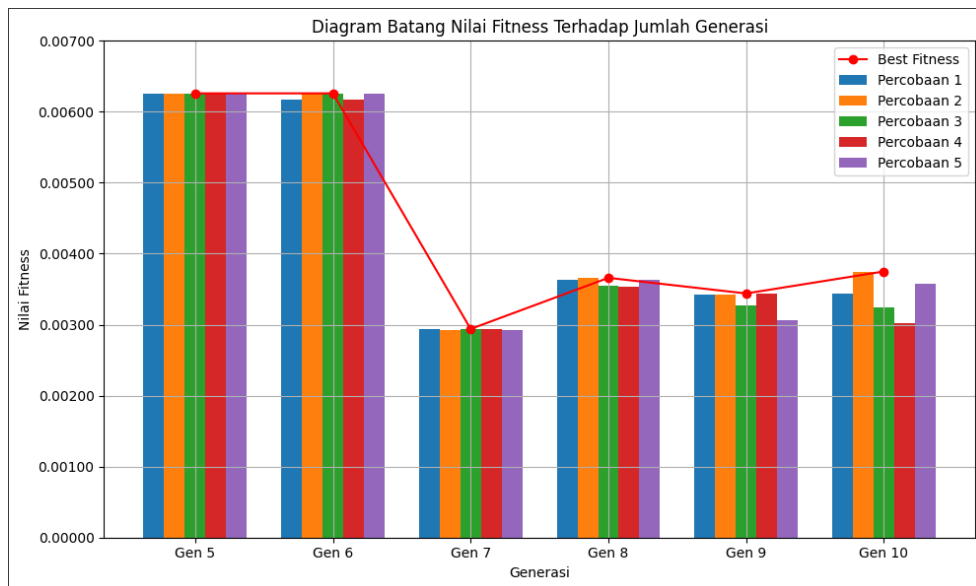
Gambar 7. Grafik Nilai *Fitness* Terhadap  $P_c$  dan  $P_m$



Dari hasil yang ditunjukkan pada Gambar 5.18, kombinasi nilai Pc dan Pm {0.5:0.5} menghasilkan rata-rata nilai fitness terbaik sebesar 0,0035686. Sebaliknya, kombinasi Pc dan Pm {0.8:0.2} memberikan rata-rata nilai fitness terendah, yaitu 0,0025584. Ini mengindikasikan bahwa penggunaan algoritma genetika untuk optimalisasi rute pengiriman pesanan di Resto Pak Lanjar Sleman mencapai tingkat optimal ketika kombinasi Pc dan Pm diatur pada {0.5:0.5}.

### 3.5 Pengujian Konsistensi Solusi

Nilai fitness konsisten dan menunjukkan hasil yang optimal untuk jumlah objek atau gen kurang dari 8. Sistem dapat memberikan rekomendasi rute dengan biaya (jarak dan waktu) yang paling murah. Nilai fitness pada jumlah gen 8 dan 10 terdapat variasi, walaupun demikian selisih variasi dengan nilai fitness terbaik sangat kecil, sehingga rute yang disarankan mendekati optimal dan menjadi rute alternatif. Perbedaan nilai fitness juga terjadi karena parameter biaya merupakan kombinasi antara jarak dan waktu perjalanan. Waktu yang diberikan memperhatikan kepadatan lalu lintas dan merupakan waktu perjalanan prediktif berdasarkan data waktu historis setiap harinya, sehingga nilainya bisa berbeda setiap waktunya



Gambar 8. Grafik Konsistensi Nilai *Fitness* Terhadap Jumlah Gen

## 4. KESIMPULAN

Berdasarkan penelitian dan pengujian yang dilakukan, penelitian ini berhasil mencapai tujuan utamanya dengan menganalisis berbagai parameter algoritma genetika serta merancang dan mengimplementasikan algoritma tersebut sebagai prototipe aplikasi untuk menyelesaikan permasalahan optimasi rute pengiriman pesanan di Resto Pak Lanjar, Sleman. Hasil pengujian menunjukkan bahwa ukuran populasi optimal untuk jumlah gen lebih dari 7 adalah 180 individu, yang menghasilkan nilai fitness terbaik. Nilai ini mengindikasikan bahwa dengan jumlah individu yang lebih besar, sistem mampu lebih efektif mengeksplorasi solusi dan mencapai hasil optimal dalam waktu yang lebih singkat. Selain itu, kombinasi probabilitas crossover dan mutasi sebesar {0.5:0.5} memberikan kinerja terbaik dalam meningkatkan kualitas solusi, dengan rata-rata fitness sebesar 0,0035686.

## UCAPAN TERIMA KASIH

Dengan kerendahan hati dan penuh rasa hormat, penulis mengucapkan terima kasih kepada:

1. Bapak Eko Riswanto, S.T., M.Cs., Ketua STMIK El Rahma Yogyakarta.
2. Bapak Yuli Praptomo PHS, S.Kom., M.Cs., Ketua Program Studi Informatika.
3. Bapak Herdiesel Santoso, S.T., S.Kom., M.Cs., dosen pembimbing yang senantiasa memberikan arahan dan bimbingan.
4. Kedua orang tua saya atas dukungan dan doanya.
5. Seluruh teman kampus STMIK El Rahma Yogyakarta yang telah memberikan bantuan dan dukungan dalam penyusunan ini.

## DAFTAR PUSTAKA

Aradiansyah, H., & Junianto, M. B. S. (2022). Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran. *Jurnal Media Informatika Budidarma*, 6(1), 329. <https://doi.org/10.30865/mib.v6i1.3418>

- Cia, N. A. (2024). IMPLEMENTASI ALGORITMA GENETIKA DALAM REKOMENDASI MAKANAN UNTUK PENDERITA OBESITAS. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(2). <https://doi.org/10.23960/jitet.v12i2.3993>
- Denny Hermawanto. (2003). *Algoritma Genetika dan Contoh Aplikasinya*. <http://dennyhermawanto.webhop.org>
- Santoso H. (2016). *Algoritma Genetika Untuk Memberikan Rekomendasi Rute Perjalanan Multi Destinasi di Daerah Istimewa Yogyakarta*. STMik El Rahma Yogyakarta.
- Pradita, H., & Arifin, J. (2023). Sistem Informasi Geografis Pemetaan Penjual Sayur Segar di Mojokerto Menggunakan Google Map API. *Julyxxxx, x, No.x*, 1–5.
- Santoso, H., & Sanuri, R. (2019). Implementasi Algoritma Genetika dan Google Maps API Dalam Penyelesaian Traveling Salesman Problem with Time Window (TSP-TW) Pada Penjadwalan Rute Perjalanan Divisi Pemasaran STMik El Rahma. *Teknika*, 8(2), 110–118. <https://doi.org/10.34148/teknika.v8i2.187>
- Sofia Tussoliha. (2019). *Optimasi Multi Traveling Salesman Problem Menggunakan Algoritma Genetika pada Distribusi Keripik Tempe "Putra Ridhlo" Di Malang*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Sugiyatno, Syafrianto, A., & Falahi, Z. (2023). Sistem Informasi Manajemen Penelitian Dan Pengabdian Masyarakat menggunakan frame work Laravel di Stmik El Rahma. *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, 21(1), 69–79. <https://doi.org/10.61805/fahma.v21i1.26>
- Suyanto. (2011). *Artificial Intelligence (Revisi Edisi)*. Informatika Bandung.
- Widayani, W., Pratama, D. A., Pratama, R. D., Kusumajaya, E. T., & Dharma, A. (2023). Pemanfaatan Metode Heuristik Travelling Salesman Problem With Time Windows Pada Rute Antar Jemput Laundry Dengan Algoritma Genetika. *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, 17(1), 1–10. <https://doi.org/10.61805/fahma.v17i1.76>
- Wulandari, S., & Helmi, Y. (2019). Penyelesaian Multiple Travelling Salesman Problem (Multi-Tsp) Dengan Metode Order Crossover Dalam Algoritma Genetika. In *Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster)* (Vol. 08, Issue 2).