

Optimization of Distribution Routes Using the Genetic Algorithm in the Traveling Salesman Problem

Rahmad Naufal^{1*}, Muhammad Siddik Hasibuan^{2**}

^{1,2*} Ilmu Komputer, Universitas Islam Negeri Sumatera Utara
rahmad.naufal@uinsu.ac.id¹, muhammadsiddik@uinsu.ac.id²

Article Info

Article history:

Received 2024-11-11

Revised 2024-12-03

Accepted 2025-01-20

Keyword:

Optimization,
Distribution Routes,
Genetic Algorithm,
Traveling Salesman Problem,
Greedy Algorithm.

ABSTRACT

Transportation plays a vital role in business operations, as it is essential for product distribution to maintain profitability. Optimizing distribution routes is crucial to reducing transportation costs, travel time, energy usage, and resource allocation while maximizing efficiency. Micro-entrepreneurs, particularly settled retailers, often face challenges in determining optimal travel routes, resulting in inefficiencies in product distribution. This issue is classified as a Traveling Salesman Problem (TSP), which involves finding the shortest possible route connecting several locations before returning to the starting point. To address this problem, this study applies a two-step approach: the greedy algorithm to provide an initial solution and the genetic algorithm for further optimization. The research employs both manual calculations and MATLAB 2018A software to solve the TSP. Results demonstrate that the optimized route reduces the travel distance by 1,260 meters compared to the initial solution, highlighting significant improvements in operational efficiency.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Pengoptimalan adalah suatu cara yang digunakan untuk merancang, mengelola, dan mengoptimalkan sesuatu demi memperoleh hasil yang ideal serta efektif. Untuk mengoptimalkan perjalanan transportasi dan mengurangi biaya operasional serta waktu tempuh dalam proses logistik dan manajemen rantai pasokan, maka dibutuhkan perencanaan rute distribusi yang terstruktur. Transportasi merupakan salah satu aspek terpenting dalam suatu bisnis, karena transportasi diperlukan untuk pendistribusian produk agar perusahaan tetap memperoleh keuntungan [1]. Distribusi suatu produk biasanya menggunakan proses perencanaan dan penyimpanan informasi, penyimpanan produk hingga produk dikirimkan. Pendistribusian produk dimulai secara bertahap di lokasi penyimpanan seperti pusat produksi, pusat distribusi, pedagang grosir, dan pengecer. Pengecer terbagi menjadi dua jenis yaitu pengecer menetap dan pengecer keliling. Pengecer menetap mengacu pada pengecer yang memiliki lokasi yang sudah ditentukan yaitu toko, sedangkan pengecer keliling keadaannya harus berkeliling sepanjang perjalanan dan tidak memiliki lokasi yang sudah ditentukan [2].

Mikro pengusaha menghadapi tantangan besar dalam mendistribusikan produk mereka akibat keterbatasan sumber daya, seperti jumlah armada transportasi yang terbatas, anggaran operasional yang kecil, dan alokasi waktu yang tidak efisien. Tanpa perencanaan rute distribusi yang optimal, biaya transportasi seringkali membengkak, sementara waktu pengiriman yang lebih lama dapat menurunkan tingkat kepuasan pelanggan. Selain itu, rute yang tidak efisien menyebabkan konsumsi bahan bakar lebih tinggi, yang secara langsung mengurangi profitabilitas usaha. Dalam konteks tersebut, diperlukan solusi yang dapat membantu mikro-pengusaha merancang rute distribusi yang lebih efisien dengan sumber daya yang ada, sehingga dapat memaksimalkan pendapatan mereka.

Pendekatan Traveling Salesman Problem (TSP) adalah solusi yang dapat digunakan [3]. TSP relevan dalam kasus distribusi produk karena membantu menentukan rute terpendek yang harus ditempuh oleh kendaraan distribusi untuk mengunjungi beberapa lokasi. Rute optimal untuk mikro-pengusaha berarti menghemat waktu, bahan bakar, dan sumber daya. Sebagai contoh, seorang penjual keliling yang mendistribusikan barang ke pelanggan di berbagai lokasi membutuhkan rute yang mengoptimalkan jumlah lokasi yang

dikunjungi tanpa harus kembali ke titik yang sama lebih dari sekali kecuali untuk kembali ke titik awal. Pendekatan Traveling Salesman Problem (TSP) adalah solusi yang dapat digunakan. TSP relevan dalam kasus distribusi produk karena membantu menentukan rute terpendek yang harus ditempuh oleh kendaraan distribusi untuk mengunjungi beberapa lokasi [4]. Rute optimal untuk mikro-pengusaha berarti menghemat waktu, bahan bakar, dan sumber daya. Sebagai contoh, seorang penjual keliling yang mendistribusikan barang ke pelanggan di berbagai lokasi membutuhkan rute yang mengoptimalkan jumlah lokasi yang dikunjungi tanpa harus kembali ke titik yang sama lebih dari sekali kecuali untuk kembali ke titik awal.

Traveling Salesman Problem merupakan sebuah permasalahan pengoptimalan yang bersifat klasik dan *Non-deterministic Polynomial-time Complete* (NPC) dengan cara mencoba seluruh kemungkinan penyelesaian yang ada [5]. Permasalahan ini biasa terjadi pada seorang *traveling salesman* dimana harus melakukan sekali kunjungan pada semua kota dalam sebuah lintasan sebelum kembali ke titik awal sehingga perjalanannya dapat dikatakan sempurna atau optimal. Metode *Traveling Salesman Problem* adalah sebuah metode yang digunakan dengan cara mencari jarak dan rute terdekat, waktu tercepat dan biaya yang minimal untuk meminimasi biaya distribusi [6]. Salah satu metode *traversal* untuk menyelesaikan permasalahan *Traveling Salesman Problem* adalah metode algoritma *greedy* [7].

Untuk menyelesaikan TSP, banyak algoritma telah dikembangkan, termasuk algoritma *greedy* dan algoritma genetika. Algoritma *greedy* digunakan untuk mencari solusi cepat dengan memilih langkah terbaik pada setiap tahap, meskipun solusi ini tidak selalu optimal [8]. Sementara itu, algoritma genetika, yang terinspirasi oleh prinsip evolusi, memungkinkan eksplorasi ruang solusi yang lebih luas untuk menemukan hasil yang lebih baik. Dengan kedua pendekatan ini, masalah optimasi rute pada TSP dapat diterapkan secara efektif dalam distribusi produk mikro pengusaha untuk meningkatkan efisiensi operasional mereka.

Algoritma *greedy* adalah algoritma yang dapat diselesaikan dengan melakukan langkah demi langkah dalam mengambil pilihan yang terbaik dengan istilah yang disebut optimum lokal [9]. Algoritma *greedy* memiliki prinsip pencarian jalur terpendek menggunakan fungsi seleksi yang berguna untuk memilih jalur tersingkat untuk menuju ke tempat yang sesuai dengan asumsi. Penerapan algoritma ini tidak dapat untuk solusi optimal namun dapat memberikan solusi awal. Kelebihan algoritma ini adalah kemampuannya dalam menemukan solusi dengan jumlah node sangat banyak dengan sangat cepat [10]. Algoritma *greedy* dapat digunakan untuk memilih dalam pemecahan masalah lintasan linier dimana asal dan tujuan disebut *node* yang berbeda [11]. Permasalahan *Traveling Salesman Problem* dapat dioptimalkan lebih lanjut dengan menggunakan metode algoritma genetika.

Algoritma *greedy* dipilih sebagai metode awal untuk TSP karena sifatnya yang sederhana, cepat, dan mampu memberikan solusi awal melalui prinsip optimum lokal [12].

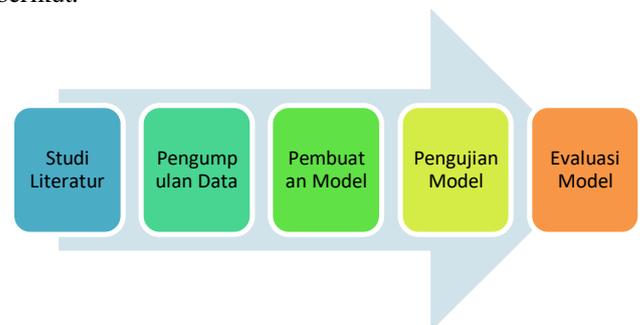
Meskipun efisien, pendekatan ini cenderung menghasilkan solusi lokal sehingga diperlukan metode tambahan untuk optimasi. Untuk menyempurnakan solusi, algoritma genetika digunakan karena kemampuannya mengeksplorasi ruang solusi secara luas melalui mekanisme reproduksi, crossover, dan mutasi. Metode ini mampu menghindari jebakan solusi lokal dan mencari solusi optimal secara global, menjadikannya cocok untuk masalah TSP yang kompleks. Kombinasi kedua algoritma ini memberikan efisiensi waktu sekaligus hasil yang lebih optimal.

Menurut Charles Darwin, “Algoritma genetika merupakan suatu algoritma yang menggunakan prinsip seleksi alam dimana individu akan terus menerus melakukan perubahan gen hingga sesuai dengan lingkungannya” [13]. Algoritma genetika merupakan suatu penerapan yang didasarkan pada genetika alami dengan cara mencari solusi yang optimal di antara beberapa alternatif sesuai fungsi probabilistik. Algoritma genetika memiliki prinsip kerja memproses solusi dengan mengadopsi konsep evolusi biologis dimana individu yang berada dalam populasi bersaing dengan melakukan reproduksi sehingga memperoleh individu baru yang memiliki gen lebih baik dari induknya. Pada proses ini seluruh individu baik induk maupun keturunannya akan melalui proses genetik yang disebut juga seleksi alam dimana individu yang kuat yang akan bertahan menuju ke generasi selanjutnya [14].

Karena sifatnya yang termasuk dalam kategori *Non-deterministic Polynomial-time Complete* (NPC), TSP merupakan masalah yang sangat kompleks dari perspektif komputasi. Seiring dengan jumlah lokasi yang harus dikunjungi, kompleksitas masalah TSP meningkat. Jumlah kemungkinan urutan perjalanan untuk n lokasi adalah $(n-1)!$. Sebagai contoh, ada 24 rute yang mungkin untuk lima lokasi, tetapi jumlah ini meningkat menjadi lebih dari 3,6 juta untuk 10 lokasi. Oleh karena itu, ketika jumlah lokasi menjadi sangat besar, TSP menjadi tantangan besar bagi komputer tradisional untuk menemukan solusi optimal dalam waktu yang tepat.

II. METODE

Kerangka penelitian ini ditunjukkan melalui gambar 1 berikut.

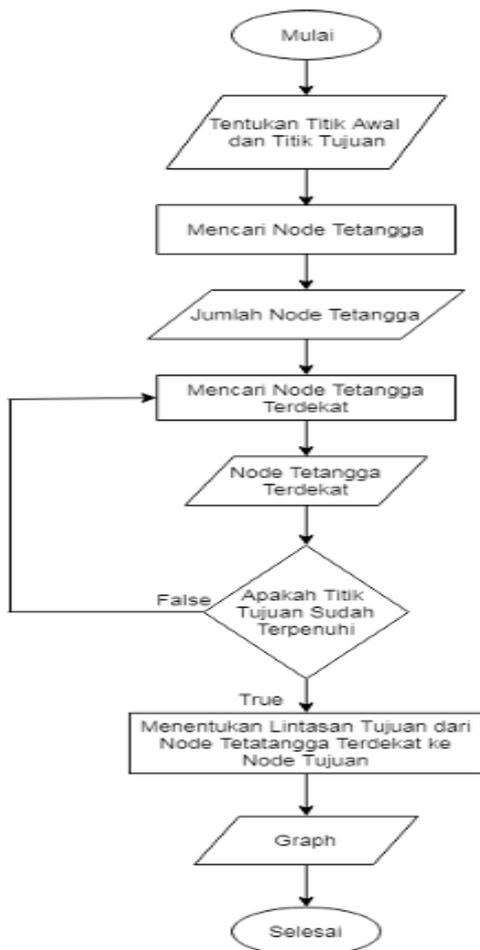


Gambar 1. Kerangka Penelitian

Berdasarkan kerangka penelitian tersebut, penelitian ini dimulai dari studi literatur. Tahapan ini merupakan proses

mencari dan mengumpulkan referensi dari buku, jurnal, maupun internet guna sebagai landasan teoritis penelitian. Proses selanjutnya adalah pengumpulan data yang dilakukan dengan empat cara yaitu observasi langsung dimana data-data yang dibutuhkan dikumpulkan, analisis data dimana data yang dikumpulkan dianalisis ke dalam *Traveling Salesman Problem* menggunakan metode algoritma *greedy* sebagai algoritma *traversal* kemudian dioptimalkan menggunakan algoritma genetika, wawancara dan diskusi dengan dosen pembimbing dimana *salesman* tempe Sultan di wawancara secara langsung mengenai rute distribusi yang dilakukan selama ini. kemudian melakukan diskusi langsung dengan dosen pembimbing mengenai hasil penelitian, dan terakhir pengambilan keputusan yaitu melakukan penarikan kesimpulan berdasarkan hasil penelitian.

Tahap yang ketiga yaitu pembuatan model. Dalam proses ini akan ditentukan rute pendistribusian produk dengan menggunakan metode algoritma *greedy* sebagai algoritma *traversal* pada kasus *Traveling Salesman Problem* dengan bantuan *software* Mathlab 2018A. proses algoritma *greedy* dapat dilihat pada gambar 2 berikut.

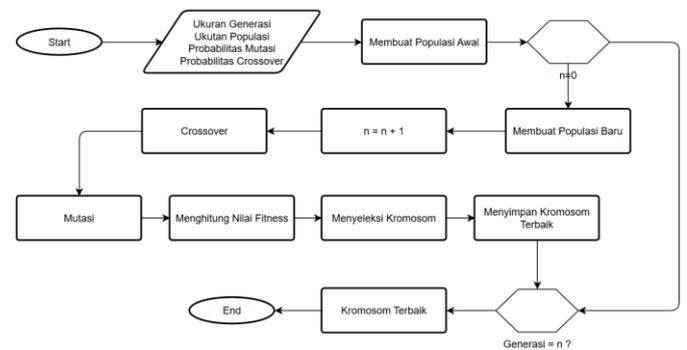


Gambar 2. Flowchart Algoritma Greedy

Kemudian rute tersebut akan dioptimalkan dengan menggunakan metode algoritma genetika dengan membuat algoritma dasar dari program yang akan menghasilkan jalur terpendek dengan menggunakan aplikasi MATLAB 2018A.

Dalam penelitian ini, MATLAB digunakan untuk mendukung proses simulasi dan perhitungan masalah *Traveling Salesman Problem* (TSP) dengan metode algoritma *greedy* dan algoritma genetika. MATLAB tidak hanya mempermudah proses komputasi kompleks seperti iterasi algoritma genetika yang mencakup mutasi, inialisasi populasi, seleksi, crossover, dan evaluasi fungsi fitness tetapi juga secara sistematis membantu algoritma *greedy* menghitung rute terpendek [15].

Penelitian ini menggunakan algoritma genetika untuk mengoptimalkan solusi dari *Traveling Salesman Problem* yang dimulai dengan menentukan variabel yang dibutuhkan berupa ukuran generasi, populasi, probabilitas mutasi, dan *crossover*. Pertama buat populasi awal dengan menentukan nilai $n = 0$ dimana populasi awal juga diambil berdasarkan hasil dari algoritma *greedy*. Kemudian untuk membuat populasi baru dan mencari nilai dari populasi, tambahkan jumlah dari setiap n untuk mencari nilai dari populasi dan mutasi digunakan sebagai alat untuk menentukan nilai *fitness*. Kromosom diseleksi melalui setiap nilai yang dimasukkan hingga memperoleh kromosom yang paling baik dengan menggunakan generasi yang berbeda. Proses ini dapat dilihat pada gambar 3 berikut.



Gambar 3. Proses Pembuatan Model Algoritma Genetika

Tahapan keempat adalah pengujian model. Proses ini merupakan tahapan melakukan *trial* dan *error* pada program yang telah dibuat sehingga diperoleh hasil yang diinginkan. Dan tahapan yang terakhir adalah evaluasi model. Proses ini merupakan proses untuk melakukan analisis dari hasil penelitian dan pengolahan data sehingga dapat memperoleh hasil penyelesaian yang dapat dijadikan referensi oleh perusahaan dalam melakukan pengoptimalan rute distribusi.

A. *Traveling Salesman Problem*

Traveling Salesman Problem (TSP) merupakan sebuah riset operasi untuk mengoptimalkan rute pendistribusian barang yang diselesaikan secara konvensional. *Traveling Salesman Problem* melibatkan aspek transportasi dimana seorang *salesman* harus mengunjungi sejumlah daerah dalam

proses pengiriman produk. Daerah yang sudah dikunjungi tidak dapat dilewati kembali atau dengan kata lain hanya dapat dilewati sekali saja lalu kembali ke titik awal [1].

Jarak tempuh transportasi dari lokasi A ke lokasi B yang dianalogikan oleh C_{ab} diartikan sebagai penentuan rute formulasi model *Traveling Salesman Problem*. Kemudian variabel keputusan X_{ab} yang akan menjelaskan bahwa jalur tersebut sudah dilalui atau belum dapat dilihat melalui persamaan berikut.

$$X_{ab} = \begin{cases} 1, & \text{jika rute transportasi dari a ke b} \\ 0, & \text{jika tidak ada rute transportasi dari a ke b} \end{cases} \dots\dots\dots(1)$$

Jika fungsi tujuan dari *Traveling Salesman Problem* adalah Z, maka fungsi Z dirumuskan dengan diminimumkan seperti pada persamaan berikut.

$$Z = \sum_{a=1}^n \sum_{b=1}^n C_{ab} X_{ab} \dots\dots\dots(2)$$

Dengan batasan kendala,

$$\sum_{a=1}^n X_{ab} = 1 \quad (i = 1,2,3, \dots, n) \dots\dots\dots(3)$$

$$\sum_{b=1}^n X_{ab} = 1 \quad (i = 1,2,3, \dots, n) \dots\dots\dots(4)$$

$$\sum_{b=1}^n X_{ab} = 0 \text{ atau } 1 \dots\dots\dots(5)$$

Batasan pertama dan kedua mendeskripsikan bahwa rute transportasi yang dituju dengan mendatangi satu kali kunjungan pada setiap kota dan meninggalkan kota tersebut satu kali [1].

B. Algoritma Greedy

Dalam memecahkan persoalan optimasi, salah satu metode yang paling populer digunakan adalah algoritma greedy. Dalam Bahasa Inggris, kata "greedy" berarti rakus, tamak, atau serakah. Algoritma ini memiliki prinsip utama yaitu "Take What You Can Get Now," yang berarti mengambil pilihan terbaik yang tersedia saat ini. Langkah-langkah dalam membentuk solusi menggunakan algoritma greedy meliputi eksplorasi berbagai pilihan pada setiap langkah solusi, di mana pilihan terbaik akan diambil sebagai keputusan. Keputusan yang telah dibuat tidak dapat diubah pada langkah selanjutnya. Algoritma greedy menggunakan pendekatan berupa pembuatan pilihan optimum lokal untuk mencapai solusi optimum global pada setiap langkahnya.

Elemen-elemen utama dalam algoritma greedy terdiri dari lima komponen. Pertama, himpunan kandidat, yang berisi elemen-elemen dengan peluang untuk membentuk solusi. Kedua, himpunan solusi, yang terdiri dari kandidat-kandidat terpilih sebagai solusi dari persoalan. Ketiga, fungsi seleksi, yang berperan dalam memilih kandidat untuk menghasilkan solusi optimal. Keempat, fungsi kelayakan, yang memastikan bahwa setiap kandidat yang dipilih memenuhi syarat untuk membentuk solusi yang layak. Terakhir, fungsi objektif, yang bertugas memberikan nilai maksimum atau minimum pada solusi sehingga hanya satu solusi terbaik yang dipilih dari himpunan solusi. Kombinasi elemen-elemen ini membuat algoritma greedy efektif untuk menyelesaikan berbagai masalah optimasi.

Cara kerja algoritma greedy dalam mencari rute dimulai dengan menentukan node awal dan node tujuan. Proses penyelesaian dilakukan secara berulang melalui beberapa langkah. Pertama, menentukan kandidat dengan memeriksa semua sisi pada node awal yang terhubung secara langsung. Selanjutnya, menentukan kandidat solusi dengan memilih sisi yang memiliki bobot terkecil dan menghitung panjang lintasan sementara. Kemudian, solusi terpilih ditentukan dengan memeriksa apakah node akhir sama dengan node tujuan. Jika tidak, maka node awal diperbarui menjadi node akhir terpilih. Proses ini diulangi hingga node tujuan ditemukan.

Untuk menemukan lintasan terpendek dalam algoritma *greedy* dapat digunakan rumusan sebagai berikut.

- 1) Periksa semua tepi yang berbatasan langsung dengan tepi A dan pilih tepi dengan bobot terkecil. Tepi ini akan menjadi lintasan terpendek atau biasa disebut dengan L(1).
- 2) Lintasan terpendek kedua dihitung dengan cara sebagai berikut:
 - a. Hitung: $d(i) = \text{panjang } L(1) + \text{bobot tepi dari simpul terakhir } L(1) \text{ ke simpul } i \text{ yang lain.}$
 - b. Pilih $d(i)$ yang terkecil. Bandingkan $d(i)$ dengan bobot tepi (a,i). Jika bobot tepi (a,i) kurang dari $d(i)$, maka $L(2) = L(1) \cup (\text{tepi simpul akhir } L(1) \text{ ke simpul } i)$.
- 3) Untuk menentukan lintasan terpendek berikutnya, ulangi langkah kedua.

C. Algoritma Genetika

Dalam penelitian ini, algoritma genetika berfungsi untuk pengoptimalan rute pendistribusian pada produk tempe Sultan. Tahapan dari algoritma genetika sebagai berikut.

1) Representasi Kromosom

Representasi solusi dari suatu permasalahan algoritma genetika disebut dengan istilah kromosom [19]. Untuk mengodekan solusi dari masalah yang akan terselesaikan dengan bentuk kromosom maka digunakan representasi kromosom. Berbagai jenis representasi kromosom yang berbeda-beda dapat dikodekan pada setiap permasalahan, seperti *real-code*, permutasi, representasi *integer*, dan *biner*. Permasalahan kombinatorial seperti *Traveling Salesman Problem* (TSP) bisa juga menggunakan representasi permutasi [14].

2) Inisialisasi Populasi Awal

Tahapan awal pembangkitan *parent* pada sejumlah *popsize* yang telah ditentukan secara acak disebut sebagai proses inisialisasi [19]. Inisialisasi populasi awal merupakan sebuah tahap awal yang terdiri dari beberapa kromosom yang diacak sehingga memperoleh solusi baru [20].

3) Proses reproduksi dengan cara crossover dan mutasi

Untuk memperoleh kromosom atau individu baru berdasarkan individu yang sudah ada dengan mempertahankan sifat baik dari *parent* maka dilakukan proses reproduksi pada algoritma genetika. Terdapat dua proses reproduksi pada algoritma genetika yaitu *crossover* dan mutasi [20].

Salah satu proses reproduksi pada algoritma genetika yang menggunakan metode *one-cut point* adalah *crossover*. Prinsip kerja dari metode ini adalah dengan cara memilih dua kromosom secara acak yang digunakan sebagai *parent*. Kedua kromosom *parent* akan berubah menjadi dua buah segmen dengan cara memilih satu titik potong. Kemudian segmen kromosom *parent* akan melakukan penukaran sehingga menghasilkan individu baru atau *offspring* [14].

Suatu proses reproduksi yang bertujuan menghasilkan perubahan secara acak pada satu kromosom disebut sebagai mutasi [14]. Metode mutasi dilakukan dengan cara menukar dua titik secara acak sehingga dalam sekali proses mutasi dapat menghasilkan satu *child* dengan satu *parent* [20]. Mutasi menggunakan teknik *insertion* dimana setiap gen dalam kromosom bergeser ke kiri dan gen pertama akan pindah ke paling akhir [19].

4) Evaluasi Fungsi Fitness

Evaluasi adalah suatu proses seleksi yang dilakukan dengan cara menilai individu terbaik melalui hasil perhitungan nilai *fitness* [20]. Proses ini dapat menilai baik buruknya sebuah kromosom [19]. Untuk mengukur seberapa baik kromosom solusi pada sebuah nilai maka digunakan nilai *fitness* dimana algoritma genetika akan mencari kromosom solusi dengan nilai *fitness* yang maksimal [14]. *Fitness* adalah nilai dari setiap individu dalam kromosom yang berfungsi untuk menentukan konsistensi individu terhadap kriteria atau tujuan masalah yang ingin dicapai [13].

5) Seleksi untuk mendapatkan individu baru di generasi selanjutnya

Proses seleksi adalah suatu proses yang berfungsi untuk memilih calon keturunan yang baru. *Parent* yang baik akan menghasilkan generasi yang baik pula. Proses seleksi individu akan berbanding lurus dengan nilai *fitness* dimana semakin besar nilai *fitness* maka semakin banyak pula peluang terpilihnya [13]. Proses seleksi digunakan untuk mendapatkan populasi baru pada setiap generasinya dimana proses ini akan bekerja dengan cara mengurutkan kumpulan individu dan semua keturunannya berdasarkan nilai *fitness* yang kemudian akan dipilih berdasarkan nilai *fitness* terbaik sehingga individu-individu terbaik yang akan lolos ke generasi selanjutnya [14]. *Offspring* atau keturunan yang berada pada proses seleksi akan menjadi *parent* pada generasi berikutnya [20]. Pada penelitian ini proses seleksi yang digunakan adalah metode seleksi *elitism* yakni memilih kromosom terbaik pada sejumlah *popSize* untuk dijadikan *parent* pada generasi berikutnya [19]. *Elitism* dilaksanakan dengan menduplikasi kromosom terbaik yang ada pada tahap evaluasi supaya dapat menggantikan kromosom dengan *fitness* terendah [20].

III. HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini mencakup 14 titik distribusi, yang terdiri dari satu titik awal sekaligus akhir perjalanan (Pabrik Pembuatan Tempe Sultan) dan 13 titik lainnya yang merupakan lokasi tujuan, seperti kedai sayuran, toko makanan, dan toko kelontong. Data ini diambil dari kasus nyata yang relevan dengan distribusi mikro-pengusaha di wilayah tertentu, termasuk alamat spesifik untuk setiap lokasi. Matriks jarak antar titik, yang diukur dalam meter dan kilometer, digunakan untuk menghitung jarak perjalanan antara setiap lokasi. Penelitian ini berfokus pada pengoptimalan rute distribusi untuk meminimalkan jarak tempuh dan biaya operasional, yang menjadi tantangan utama bagi mikro-pengusaha dalam mendistribusikan produk secara efisien. Daftar nama pelanggan tetap Tempe Sultan beserta alamat dan jaraknya dapat dilihat pada tabel 1 berikut.

TABEL I
DATA LOKASI TEMPE SULTAN

No.	Naina Tujuan Lokasi	Simbol	Alamat Lokasi	Jarak lokasi
1	Pabrik Pembuatan Tempe Sultan	A	Jl. Karya Tani Gg. Pinang No. 91	0m
2	Kedai Sayur Melinjo	B	Jl. Karya Tani Gg. Melinjo	350 m
3	Kedai Sayuran	C	Jl. Karya Tani Ujung Depan Mesjid Al-Ikhlas	600 m
4	Kedai Sayuran	D	Jl. Karya Bakti No. S3	550 ni
5	Kedai Sayuran	E	Jl. Karya Jaya No. 97	900 m
6	Kedai Sayuran	F	Jl. Karya Kasih No. 104A	130 m
7	Kedai Sayur El Fateeh	G	Jl. Karya Kasili No. 49	280 m
8	Kedai Sayu- Barona	H	Jl. Karya Kasili No. 48	350 m
9	Kedai Azizi Sayu-	I	Jl. Karya Bakti No. 84 D/1A	800 m
10	Toko Barokah Zhuha 2	J	Jl. Karya Jaya No. 133	650 m
11	Kedai Azizi Sayur 3	K	Jl. Eka Rasmi No. 13	850 ni
12	Toko Rizki	L	Jl. Eka Suka No. 7	350 ni
13	Kedai Isput	M	Jl. Eka Rasmi No. 91	1400 m
14	Pabrik Pembuatan Tempe Sultan	A	Jl. Karya Tani Gg. Pinang No. 91	2200 m

TABEL II
DATA TITIK PERSIMPANGAN LOKASI

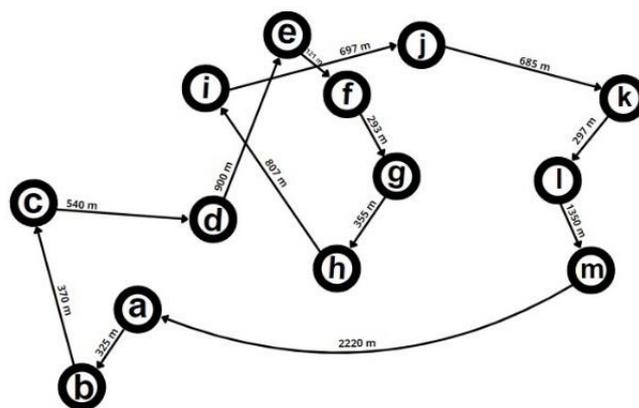
No	Nama Lokasi yang dilalui	Simbol	Alaniat
1	Depot Air Isi ulang Bella Water	a1	Jl. Karya Tani No.98
2	Apotik Gina	a2	Jl. Karya Bakti No.95C
3	Demanta Service Tambal Ban	a3	Jl. Kaiya Jaya No.83
4	Ayara Penyet Podo Solo	a4	Jl. Kaiya Kasih No.37
5	Toko Rokok Elektrik Rosemary Vaporizer	a5	Jl. Kaiya Sehati No.25
6	Waning Lontong Mbak Yuli	a6	Gg. Eka Suka II

TABEL III
DATA MATRIKS JARAK ANTAR LOKASI

Titik 0	A	B	C	D	E	...	M
A	0	350 m	400 m	750 m	1600 m (1,6 bn)	...	2200 m (2,2 bn)
B	350 m	0	600 m	900 m	1800 m (1,8 bn)	...	1900 m (12 bn)
C	400 m	600 m	0	550 m	1200 m (1,2 bn)	...	2500 m (2,5 bn)
D	750 m	900 m	550 m	0	900 m	...	2100 m (2,1 bn)
E	1600 m (1,6 kin)	1800 m (1,8 kin)	1200 m (1,2 km)	900 m	0	...	2700 m (22 bn)
F	1700 m (1,7 kin)	1900 m (12 km)	1300 m (1,3 km)	1000 m (1,0 bn)	130 m	...	2600 m (2,6 bn)
G	1400 m (1,4 kin)	1600 m (1,6 km)	1300 m (1,3 bn)	700 m	400 m	...	2400 m (2,4 bn)
H	1100 m (1,1 kin)	1200 m (1,2 km)	900 m	300 m	800 m	...	2100 m (2,1 bn)
I	1500 m (1,5)	1700 m (0,7)	1100 m (0,1)	800 m	140 m	...	2800 m (2,8)

Algoritma genetika digunakan untuk menyelesaikan *Traveling Salesman Problem (TSP)* dengan parameter sebagai berikut: ukuran populasi sebanyak 50 individu yang merepresentasikan berbagai kemungkinan rute; jumlah generasi sebanyak 100 untuk memberikan ruang iterasi yang cukup dalam menemukan solusi optimal; probabilitas mutasi sebesar 0,05 (5%) untuk menghindari solusi terjebak di optimum lokal; probabilitas crossover sebesar 0,8 (80%) untuk menghasilkan kombinasi rute baru dari dua induk; metode seleksi menggunakan *roulette wheel*, sehingga individu dengan performa lebih baik memiliki peluang lebih besar untuk direproduksi; serta fungsi fitness berdasarkan total jarak perjalanan, yang dihitung berdasarkan matriks jarak yang tersedia. Pendekatan ini bertujuan menemukan rute terpendek yang memberikan efisiensi maksimum.

Pada penelitian ini, perhitungan jarak tempuh sementara dilakukan dengan bantuan *Google Maps*. Berikut gambar graph dari penelitian ini adalah sebagai berikut.



Gambar 4. Graph rute pendistribusian produk tempe Sultan

A. Perhitungan Manual Menggunakan Algoritma Greedy

Proses perhitungan manual dilakukan dengan cara menentukan titik awal dan hasil titik terdekat. Kemudian menentukan himpunan kandidat dan himpunan solusi. Selanjutnya menentukan fungsi seleksi, fungsi kelayakan, dan fungsi objektif. Setelah itu melakukan penyelesaian perhitungan dengan rumus berikut.

$$L(n) = L(n-1) + d(i)$$

Contoh perhitungan pada penelitian ini adalah sebagai berikut.

Proses perhitungan 1

Titik Awal : Pabrik Pembuatan Tempe Sultan (A)

Hasil Kedai Terdekat : Kedai Sayur Melinjo (B)

Himpunan Kandidat : Pabrik Pembuatan Tempe Sultan (A) – Kedai Sayur Melinjo (B)

Himpunan Solusi:

- Node A – a1 – B = 350 m
- Node A – a1 – C = 400 m
- Node A – a1 – M = 2200 m

Fungsi Seleksi : Node A – a1 – B

Fungsi Kelayakan : Node A – a1 – B

Fungsi Objektif : nilai minimum adalah Node A – a1 – B

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 75 + 275 = 350$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma Greedy adalah melalui node A–a1–B dengan total jarak 350 m ditunjukkan pada gambar 5 berikut.



Gambar 5. Graf rute dari vertex A ke vertex B

Setelah melakukan proses perhitungan seperti pada di atas hingga uite dari vertex M ke vertex A maka diperoleh rute terpendek menggunakan algoritma *greedy* adalah $A \rightarrow B \rightarrow C \rightarrow D \rightarrow I \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow J \rightarrow L \rightarrow K \rightarrow M \rightarrow A$ dengan total jarak 9200 m.

B. Perhitungan Manual Menggunakan Algoritma Genetika

Perhitungan manual menggunakan algoritma genetika dapat dilakukan dengan langkah-langkah berikut.

1) Membangkitkan Populasi Awal

Berdasarkan tabel 4 populasi awal dibangkitkan secara acak dari sejumlah individu atau kromosom. Penelitian ini diambil 10 kromosom (Kr) secara acak untuk pembangkitan populasi awal.

TABEL IV
PEMBANGKITAN POPULASI AWAL

Kr	Rute													
1	A	B	C	D	I	E	F	G	H	J	L	K	M	A
2	A	B	C	D	H	G	F	E	L	I	M	K	J	A
3	A	E	G	B	D	F	C	H	I	J	K	L	M	A
4	A	M	F	J	I	E	C	D	B	L	G	K	H	A
5	A	I	D	L	C	G	M	E	J	H	K	B	F	A
6	A	J	M	E	C	L	F	H	I	D	G	B	K	A
7	A	B	M	E	H	J	I	K	G	F	C	D	L	A
8	A	I	K	L	M	B	H	D	F	J	G	E	C	A
9	A	H	D	J	G	B	L	I	F	M	E	C	K	A
10	A	C	L	H	B	D	M	F	I	J	K	E	G	A

2) Evaluasi Kromosom

Fungsi *fitness* ditentukan berdasarkan fungsi objektifnya sehingga dapat menentukan tingkat kualitas kromosom yang merepresentasikan penyelesaian masalah. Untuk menentukan fungsi *fitness*, hitung lebih dahulu besar jarak tiap kromosom seperti berikut.

- Kromosom 1 :** $A + B + C + D + I + E + F + G + H + J + L + K + M + A$
 $= 350 + 600 + 550 + 800 + 140 + 130 + 280 + 350 + 1100 + 1000 + 350 + 1300 + 2200 = 9150$
- Kromosom 2 :** $A + B + C + D + H + G + F + E + L + I + M + K + J + A$
 $= 350 + 600 + 550 + 300 + 350 + 280 + 130 + 1400 + 1600 + 2800 + 1300 + 850 + 2200 = 12710$
- Kromosom 3 :** $A + E + G + B + D + F + C + H + I + J + K + L + M + A$
 $= 1600 + 400 + 1600 + 900 + 1000 + 1300 + 900 + 800 + 650 + 850 + 350 + 1400 + 2200 = 13950$
- Kromosom 4 :** $A + M + F + J + I + E + C + D + B + L + G + K + H + A$
 $= 2200 + 2600 + 450 + 650 + 140 + 1200 + 550 + 900 + 2400 + 1700 + 1600 + 1600 + 1100 = 17090$
- Kromosom 5 :** $A + I + D + L + C + G + M + E + J + H + K + B + F + A$
 $= 1500 + 800 + 1500 + 2000 + 1300 + 2400 + 1700 + 500 + 1100 + 1600 + 2700 + 1900 + 1700 = 21700$
- Kromosom 6 :** $A + J + M + E + C + L + F + H + I + D + G + B + K + A$
 $= 2200 + 2200 + 2700 + 1200 + 2000 + 1400 + 650 + 800 + 800 + 700 + 1600 + 2700 + 2500 = 21450$
- Kromosom 7 :** $A + B + M + E + H + J + I + K + G + F + C + D + L + A$
 $= 350 + 1900 + 2700 + 800 + 1100 + 650 + 1500 + 1600 + 280 + 1300 + 550 + 1500 + 2200 = 16430$
- Kromosom 8 :** $A + I + K + L + M + B + H + D + F + J + G + E + C + A$
 $= 1500 + 1500 + 350 + 1400 + 1900 + 1200 + 300 + 1000 + 450 + 750 + 400 + 1200 + 400 = 12350$
- Kromosom 9 :** $A + H + D + J + G + B + L + I + F + M + E + C + K + A$
 $= 1100 + 300 + 1400 + 750 + 1600 + 2400 + 1600 + 270 + 2600 + 2700 + 1200 + 2400 + 2500 = 20820$
- Kromosom 10 :** $A + C + L + H + B + D + M + F + I + J + K + E + G + A$
 $= 400 + 200 + 1200 + 1200 + 900 + 2100 + 2600 + 270 + 650 + 850 + 1400 + 400 + 1400 = 15370$

Besar nilai rute pada tiap-tiap kromosom dapat dilihat pada tabel 5 berikut.

TABEL V
TOTAL RUTE TIAP KROMOSOM

F _x (X)	
V1	9.150
V2	12.710
V3	13.950
V4	17.090
V5	21.700
V6	21.450
V7	16.450
V8	12.350
V9	20.820
V10	15.370
TOTAL	161.020

Nilai f(x) diperoleh melalui penjumlahan dari rute-rute tiap kromosom dimana setiap representasi kromosom v_i menjadi f_i untuk masing-masing lokasi.

3) Seleksi

Pada penelitian ini, proses seleksi populasi dilakukan dengan menggunakan metode Roda Rollet (*Roulette Wheel*) dimana orang tua dipilih berdasarkan besar nilai *fitness*-nya. Semakin baik nilai *fitness*-nya maka semakin besar kemungkinan terpilihnya. Untuk mencari besar nilai *fitness* pada tiap-tiap kromosom, maka digunakan rumus:

$$Eval(V_i) = Total F(x) - V_i$$

Besar nilai *fitness* pada tiap-tiap kromosom dapat dilihat pada tabel 6 berikut.

TABEL VI
NILAI *FITNESS* TIAP KROMOSOM

Eval (V1)	= 161.020-9.150	= 151.870
Eval (V2)	= 161.020- 12.710	= 148.310
Eval (V3)	= 161.020- 13.950	= 147.070
Eval (V4)	= 161.020- 17.090	= 143.930
Eval (V5)	= 161.020-21.700	= 139.320
Eval (V6)	= 161.020-21.450	= 139.570
Eval (V7)	= 161.020- 16.450	= 144.590
Eval (V8)	= 161.020- 12.350	= 148.670
Eval (V9)	= 161.020-20.820	= 140.200
Eval (V10)	= 161.020- 15.370	= 145.650
Total <i>fitness</i> (F)		1.449.180

4) *Fitness Relatif (P_i) dan Kumulatif (q_i)*

Setelah diperoleh besar nilai total *fitness*, selanjutnya dihitung besar *fitness* relatif dan kumulatif pada tiap-tiap kromosom dengan rumus :

$$p_i = \frac{Eval(V_i)}{F} \text{ dan } q_i = q_{i-1} + p_i$$

Besar nilai *fitness* relatif dan kumulatif pada tiap-tiap kromosom dapat dilihat pada tabel 7 berikut.

TABEL VII
FITNESS RELATIF DAN KUMULATIF

	P _i	q _i
V1	$= \frac{151870}{1.449.180} = 0,10479719565$	0,10479719565
V2	$= \frac{148310}{1.449.180} = 0,10234063401$	0,20713782966
V3	$= \frac{147.070}{1.449.180} = 0,10148497771$	0,30862280737
V4	$= \frac{143.930}{1.449.180} = 0,09931823513$	0,4079410425
V5	$= \frac{139.320}{1.449.180} = 0,09613712582$	0,50407816832
V6	$= \frac{139570}{1.449.180} = 0,09630963717$	0,60038780549
V7	$= \frac{144.590}{1.449.180} = 0,0997736651$	0,70016147059
V8	$= 148670 = 0,10258905036$	0,80275052095
V9	$= \frac{200}{1.449.180} = 0,09674436577$	0,89949488672
V10	$= \frac{145.650}{1.449.180} = 0,10050511323$	1

5) *Calon Induk Crossover (< 0,25)*

Operator penyilangan dilakukan berdasarkan hasil dari proses seleksi yang diawali dengan pemilihan pasangan induk kromosom melalui penentuan probabilitas penyilangan dan pembangkitan bilangan acak. Apabila bilangan acak kurang dari probabilitas penyilangan maka kromosom tersebut dinyatakan sebagai kromosom induk. Kromosom calon induk dipilih berdasarkan hasil probabilitas penyilangan sebesar 25% seperti yang terlihat pada tabel 8 berikut.

TABEL VIII
CALON INDUK *CROSSOVER*

Random Ke-	Bilangan acak 0-1	Calon induk <i>Crossover</i>
1	0,1237	V ₂
2	0,1869	V ₂
3	0,2732	V ₃
4	0,4512	V ₅
5	0,5223	V ₆
6	0,6372	V ₇
7	0,7999	V ₈
8	0,2633	V ₃
9	0,9632	V ₁₀
10	0,2231	V ₃

6) *Induk Crossover (< 0,5)*

Berdasarkan tabel 8 calon kromosom induk yang terpilih yaitu kromosom 2,3,5,6,7,8, dan 10. Kemudian bangkitkan kembali 2 bilangan acak sehingga diperoleh 2 induk *crossover* untuk disilangkan. Kromosom induk dipilih berdasarkan hasil probabilitas penyilangan sebesar 50% seperti yang terlihat pada tabel 9 berikut.

TABEL IX
INDUK CROSSOVER

Random ke-	Calon induk	Bilangan acak	Induk Crossover
1	V ₂	0,632	
2	V ₂	0,412	V ₂
3	V ₃	0,777	
4	V ₅	0,873	
5	V ₆	0,641	
6	V ₇	0,932	
7	V ₈	0,552	
8	V ₃	0,512	
9	V ₁₀	0,732	
10	V ₃	0,212	V ₃

7) *Penyilangan Induk Crossover V2 dengan V3*

V ₂	A B C D H G F	E L I M K J
V ₃	A E G B D F C	H I J K L M

Tukar Sub V₂ Dengan V₃

V ₂	A B C D H G F	H I J K L M
V ₃	A E G B D F C	E L I M K J

- Pemetaan sub V₂ Dengan V₃
H I J K L M

E L I M K J

- Aturan = $H \Leftrightarrow E$
 $I \Leftrightarrow L \Leftrightarrow K \Leftrightarrow M \Leftrightarrow J \Leftrightarrow I$

- Anak hasil pemetaan

V ₂	A B C D E G F H I J K L M
V ₃	A H G B D F C E L I M K J

8) *Mutasi*

Kromosom yang akan mengalami mutasi dipilih secara acak berdasarkan bilangan yang dibangkitkan secara acak. Pada penelitian ini, tukar dua gen dari induk kromosom V₂ secara acak.

V ₂	A B C D E G F H I J K L M
V ₁₁	A B C D H G F E I J K L M

9) *Evaluasi Anak*

Setelah kromosom mengalami mutasi, selanjutnya bangkitkan kembali beberapa kromosom baru secara acak. Hitung kembali besar total jarak pada kromosom hasil mutasi dengan kromosom baru yang dibangkitkan secara acak.

- Kromosom 11 : $A + B + C + D + H + G + F + E + I + J + K + L + M + A$
 $= 350 + 600 + 550 + 300 + 350 + 280 + 130 + 140 + 650 + 850 + 350 + 1400 + 2200$
 $= 8.150$

- Kromosom 12 : $A + B + C + D + E + G + F + H + I + J + K + L + M + A$
 $= 350 + 600 + 550 + 900 + 400 + 280 + 650 + 800 + 650 + 850 + 350 + 1400 + 2200$
 $= 9.980$
- Kromosom 13 : $A + H + G + B + D + F + C + E + L + I + M + K + J + A$
 $= 1100 + 350 + 1600 + 900 + 1000 + 1300 + 1200 + 1500 + 1600 + 2800 + 1300 + 850 + 2200$
 $= 17.700$

10) *Populasi Baru*

Setelah semua proses dilakukan maka akan diperoleh beberapa kromosom baru. Selanjutnya lakukan kembali proses perhitungan algoritma genetika mulai dari awal hingga diperoleh jarak terpendek seperti terlihat pada tabel 10 berikut.

TABEL X
POPULASI BARU

V ₁₁	A B C D H G F E I J K L M A	8.150
V ₁	A B C D I E F G H J L K M A	9.150
V ₁₂	A B C D E G F H I J K L M A	9.980
V ₈	A I K L M B H D F J G E C A	12.350
V ₂	A B C D H G F E L I M K J A	12.710
V ₃	A E G B D F C H I J K L M A	13.950
V ₁₀	A C L H B D M F I J K E G A	15.370
V ₇	A B M E H J I K G F C D L A	16.430
V ₄	A M F J I E C D B L G K H A	17.090
V ₁₃	A H G B D F C E L I M K J A	17.700
v ₉	A H D J G B L I F M E C K A	20.820
V ₆	A J M E C L F H I D G B K A	21.450
V ₅	A I D L C G M E J H K B F A	21.700

C. *Perhitungan Menggunakan Matlab 2018A dengan Metode Algoritma Greedy dan Algoritma Genetika*

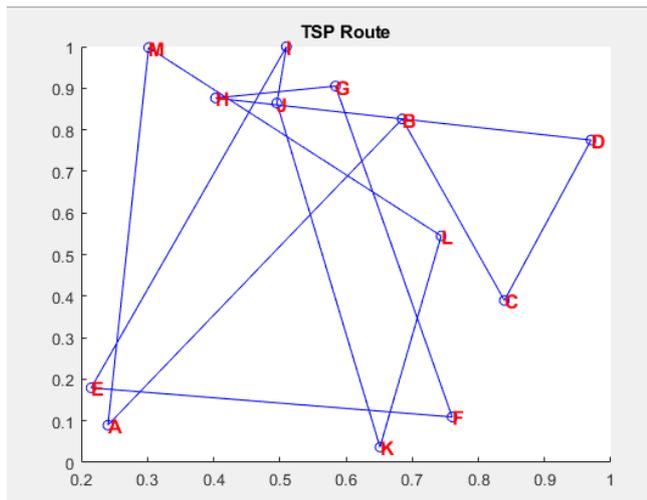
Setelah dilakukan perhitungan secara manual menggunakan metode algoritma *greedy* dan algoritma genetika, selanjutnya melakukan perhitungan menggunakan Matlab 2018A dan membandingkan hasilnya.

```

Command Window
>> GG
Best Route:
A B C D H G F E I J K L M
Total Distance:
      8150

```

Gambar 6. Command window Matlab 2018A



Gambar 7. Hasil rute menggunakan Matlab 2018A

IV. KESIMPULAN

Berdasarkan hasil perhitungan dapat disimpulkan bahwa hasil perbandingan rute pendistribusian sebelum dan sesudah dioptimalkan adalah rute pendistribusian sebelum dioptimalkan yaitu $A - B - C - D - E - F - G - H - I - J - K - L - M - A = 9.410$ m. sedangkan rute pendistribusian setelah dioptimalkan yaitu $A - B - C - D - H - G - F - E - I - J - K - L - M - A = 8.150$ m. setelah dioptimalkan jarak tempuh berkurang sebesar 1.260 m yang menunjukkan efisiensi yang signifikan. Optimasi ini memberikan manfaat praktis yang signifikan bagi mikro-pengusaha, terutama dalam distribusi produk. Dengan pengurangan jarak tempuh, biaya transportasi dapat ditekan, termasuk penghematan bahan bakar hingga waktu dan pengiriman yang lebih singkat. Hal ini tidak hanya meningkatkan efisiensi operasional tetapi juga kepuasan pelanggan, yang berkontribusi pada peningkatan daya saing usaha mikro di pasar yang semakin kompetitif.

DAFTAR PUSTAKA

- [1] M. V. Ardiansyah, R. A. Darajatun, and D. N. Rinaldi, "Optimalisasi Pendistribusian Dengan Metode Travelling Salesman Problem Untuk Menentukan Rute Terpendek Di Pt Xyz," *Tekmapro J. Ind. Eng. Manag.*, vol. 16, no. 2, pp. 84–95, 2021, doi: 10.33005/tekmapro.v16i2.264.
- [2] K. Auliasari, M. Kertaningtyas, and D. W. L. Basuki, "Optimalisasi Rute Distribusi Produk Menggunakan Metode Traveling Salesman Problem," *J. Sains, Teknol. dan Ind.*, vol. 16, no. 1, p. 15, 2018, doi: 10.24014/sitekin.v16i1.6109.
- [3] Yendrizal, "Distribusi Produk Menggunakan Metode Travelling Salesman Problem (TSP) Dengan Konsep Algoritma Heuristik," *J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 3, pp. 818–824, 2024.
- [4] A. Setiawan, *Perbandingan Algoritma Greedy Dan Cheapest Insertion Heuristics Dalam Menentukan Rute Pendistribusian Barang Dari Gateway Ke Drop Point (Studi Kasus : J & T Express Kota Jambi)*. 2024.
- [5] Nasywa Shafa Azzahra, "Analisis Optimasi Jalur Distribusi Menggunakan Pendekatan Tsp (Traveling Salesman Problem) Untuk Meningkatkan Efisiensi Biaya Distribusi Pada Toko Uthe Grosir," *J. Humaniora, Sos. dan Bisnis*, vol. 2, no. 6, pp. 542–553, 2024.
- [6] Sartika, "Pengoimalan Saluran Distribusi Kue Dengan Metode Travelling Salesman Problem (Tsp) Untuk Minimasi Jarak Dengan Rute Terpendek," *Jur. Tek. Ind. Fak. Tek. Univ. Tanjungpura, Pontianak*, vol. 3 no 2, pp. 82–88, 2019.
- [7] Irwin Supriadi, A. Mauluddin, Ichsan Ibrahim, and Pahlawan Sagala, "jurnal Implementasi Algoritma Simple Hill Climbing Sebagai Optimasi Travelling Salesman Problem Dalam Penentuan Rute Pengiriman Barang (Studi Kasus: Jasa Pengiriman Barang J&T)," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 15, no. 1, pp. 70–80, 2023, doi: 10.37424/informasi.v15i1.220.
- [8] T. Apriliani, A. S. D. Putri, J. Feranita, and M. P. Mentari, "Implementasi Algoritma Greedy Dalam Penukaran Uang Di Alfamart Di Kota Dan Algoritma String Matching Untuk Pencarian Cabang Alfamart Di Kota-Kota Di Indonesia," *SABER J. Tek. Inform. Sains dan Ilmu Komun.*, vol. 2, no. 2, pp. 283–294, 2024.
- [9] R. D. Saktia Purnama *et al.*, "Implementasi Penggunaan Algoritma Greedy Best First Search Untuk Menentukan Rute Terpendek Dari Cilacap Ke Yogyakarta," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4068.
- [10] E. R. Ningrum, A. Sanwidi, R. Akbarita, and M. N. H. Qomaruddin, "Optimasi Rute Pendistribusian Gas Elpiji Menggunakan Algoritma Floyd Warshall Dan Algoritma Greedy," *J. Ilm. Mat. Dan Terap.*, vol. 20, no. 1, pp. 1–14, 2023, doi: 10.22487/2540766x.2023.v20.i1.15568.
- [11] A. Lukman and D. T. Informatika, "Penyelesaian Travelling Salesman Problem dengan Algoritma Greedy".
- [12] F. D. Sianipar *et al.*, "Estimasi Rute Terdekat Dari Universitas Negeri Medan Ke Spbu Terdekat Menggunakan Algoritma Greedy," vol. 8, no. 6, pp. 12218–12225, 2024.
- [13] A. Yusron Mubarak and U. Chotijah, "Penerapan Algoritma Genetika Untuk Mencari Optimasi Kombinasi Jalur Terpendek Dalam Kasus Travelling Salesman Problem," *J. Teknol. Terpadu*, vol. 7, no. 2, pp. 77–82, 2021, doi: 10.54914/jtt.v7i2.424.
- [14] F. Ramadhani, F. A. Fathurrachman, R. Fitriawanti, A. C. Rongre, and V. N. Wijayaningrum, "Optimasi Pendistribusian Barang Farmasi Menggunakan Algoritma Genetika," *Klik - Kumpul. J. Ilmu Komput.*, vol. 5, no. 2, p. 159, 2018, doi: 10.20527/klik.v5i2.151.
- [15] A. S. Jingo Pratama, Abdul Khamid, and Yesy Diah Rosita, "Pencarian Rute Optimal Wisata Mojokerto Dalam Kasus Traveling Salesman Problem Menggunakan Algoritma Genetika," *J. Inform. Teknol. dan Sains*, vol. 5, no. 2, pp. 283–288, 2023, doi: 10.51401/jinteks.v5i2.2447.
- [16] B. S. Utomo, "Implementasi Algoritma Greedy Untuk Mencari Jalur Terpendek Pada Sebuah Lokasi Wisata Kota Ternate," vol. 16, no. 3, pp. 13–22.
- [17] D. Silvi and P. Dwiza, "Promosi Marketing Menggunakan Algoritma Genetika Dan," vol. 3, no. September, pp. 299–313, 2016.
- [18] T. S. Nurdianti, "Penerapan algoritma greedy untuk menentukan rute terpendek antar klinik gigi di kota medan berbasis mobile," 2020.
- [19] G. E. Yuliasuti, W. F. Mahmudy, and A. M. Rizki, "Penanganan Fuzzy Time Window pada Travelling Salesman Problem (TSP) dengan Penerapan Algoritma Genetika," *Matics*, vol. 9, no. 1, p. 38, 2017, doi: 10.18860/mat.v9i1.4072.
- [20] A. S. Fauziah, I. Cholissodin, and B. Rahayudi, "Optimasi Pendistribusian Air Mineral menggunakan Algoritma Genetika," vol. 6, no. 2, pp. 966–972, 2022.