

Implementasi Algoritma *Greedy* untuk Optimasi Rute Layanan Logistik UMKM di Kota Tegal

Wresty Andriani^{1,*}, Gunawan Gunawan² dan Naella Nabila Putri W.N³

¹ Informatika/Universitas Bima Sakapenta, Indonesia

²Sistem Informasi/ Universitas Panca Sakti Tegal, Indonesia

³Manajemen/ Universitas Negeri Semarang, Indonesia

*) Corresponding author: wresty.andriani@gmail.com

(Submit pada : 6 Novemberr 2025 | Terbit pada : 30 November 2025)

Abstract

The problem of route efficiency remains a major obstacle for logistics services of MSMEs in Tegal City. This study aims to implement the Greedy nearest neighbor algorithm to construct delivery routes that are more efficient than couriers' manual routes. The method used is a quantitative experiment based on a distance matrix from the Google Distance Matrix API, with a case study of 10 points and extended scenarios of 30–50 points. The algorithm is implemented in Python and evaluated using metrics of travel distance, travel time, percentage of savings, as well as standard deviation and 95% confidence intervals. The results show that for 10 points, the manual route covers about 46.05 km (± 92.10 minutes), whereas the Greedy route covers only 25.91 km (± 51.82 minutes), yielding distance and time savings of about 43.74%. For the 30- and 50-point scenarios, distance is reduced by about 35–36% with computation times under 1 second. These findings indicate that the Greedy nearest neighbor algorithm is suitable as a foundation for data-driven MSME logistics route optimization systems.

Abstrak

Permasalahan efisiensi rute pengiriman masih menjadi hambatan utama layanan logistik UMKM di Kota Tegal. Penelitian ini bertujuan mengimplementasikan algoritma Greedy nearest neighbor untuk menyusun rute pengiriman yang lebih efisien dibandingkan rute manual kurir. Metode yang digunakan adalah eksperimen kuantitatif berbasis distance matrix dari Google Distance Matrix API pada studi kasus 10 titik dan skenario perluasan hingga 30–50 titik. Algoritma diimplementasikan dengan Python dan dievaluasi menggunakan metrik jarak tempuh, waktu tempuh, persentase penghematan, serta simpangan baku dan interval kepercayaan 95%. Hasil pengujian menunjukkan bahwa pada 10 titik, rute manual menempuh sekitar 46,05 km ($\pm 92,10$ menit), sedangkan rute Greedy hanya 25,91 km ($\pm 51,82$ menit) dengan penghematan jarak dan waktu sekitar 43,74%. Pada skenario 30 dan 50 titik, jarak berkurang sekitar 35–36% dengan waktu komputasi di bawah 1 detik. Temuan ini mengindikasikan algoritma Greedy nearest neighbor layak dijadikan fondasi sistem optimasi rute logistik UMKM berbasis data.

Keywords: *Delivery route planning, Google Distance Matrix API, Greedy nearest neighbor, MSME logistics, Route optimization.*

PENDAHULUAN

UMKM (Usaha Mikro, Kecil, dan Menengah) memiliki peran *krusial* dalam pertumbuhan ekonomi daerah, termasuk di Kota Tegal, yang dikenal dengan sektor perdagangan dan industri rumahannya yang berkembang pesat. Dalam menghadapi era digital dan persaingan pasar yang semakin kompetitif, efisiensi layanan logistik menjadi faktor penentu keberhasilan UMKM dalam memenuhi kebutuhan pelanggan secara tepat waktu dan hemat biaya [1]. Namun, banyak pelaku UMKM di daerah masih mengandalkan metode manual dalam menentukan rute pengantaran produk, yang seringkali tidak optimal dari segi jarak maupun waktu tempuh [2].

Masalah utama yang dihadapi adalah ketidakefisienan dalam proses pengantaran barang akibat rute yang tidak terstruktur dengan baik. Permasalahan dalam sistem distribusi logistik UMKM seringkali terletak pada efisiensi rute pengiriman efisiensi rute pengiriman [3], [4]. Hal ini tidak hanya berdampak pada tingginya biaya operasional, tetapi juga menurunkan kepuasan pelanggan akibat keterlambatan pengiriman. Di sisi lain, algoritma komputer menawarkan berbagai pendekatan dalam menyelesaikan masalah optimasi rute, salah satunya adalah algoritma *Greedy* yang dikenal karena kesederhanaannya dalam menghasilkan solusi cepat dengan beban komputasi ringan [5].

Beberapa penelitian menunjukkan bahwa algoritma *greedy* mampu memberikan efisiensi rute dan waktu dalam konteks logistik berskala terbatas, seperti distribusi makanan segar dan UMKM [5], [6]. Meskipun studi lain menunjukkan bahwa algoritma A* dan metode *heuristik* lainnya lebih unggul dalam aspek akurasi dan efisiensi logistik besar, *greedy* tetap relevan untuk kasus yang memprioritaskan kecepatan dan implementasi yang sederhana [1], [7]. Terlebih lagi, pendekatan ini menjadi menarik untuk digunakan oleh pelaku UMKM dengan sumber daya terbatas dan kebutuhan pengiriman yang tidak terlalu kompleks.

Berdasarkan kajian tersebut, dapat diidentifikasi *research gap* dari penelitian ini adalah sebagai berikut: (1) masih terbatasnya kajian yang secara spesifik memodelkan optimasi rute pengantaran untuk UMKM di kota menengah seperti Kota Tegal dengan mempertimbangkan karakteristik operasionalnya, dan (2) belum banyak penelitian yang mengevaluasi kelayakan *algoritma* yang sederhana dan ringan, seperti *Greedy nearest neighbor*, sebagai kandidat modul *optimasi rute* yang dapat diintegrasikan ke sistem layanan logistik UMKM secara *near real-time* dengan metrik kinerja yang terukur (total jarak, waktu tempuh, dan waktu komputasi).

Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan *algoritma Greedy* dalam menyusun rute logistik yang efisien bagi UMKM di Kota Tegal. Secara khusus, kontribusi terukur dari penelitian ini adalah: (1) memformulasikan model rute pengiriman UMKM berbasis algoritma *Greedy nearest neighbor* dengan input koordinat lokasi yang dirancang agar dapat diintegrasikan ke sistem *optimasi rute near real-time*; (2) menyusun studi kasus pada 10 titik lokasi UMKM di Kota Tegal dan mengukur kinerja rute dalam bentuk total jarak tempuh, estimasi waktu tempuh, serta waktu komputasi; dan (3) membandingkan *performa rute Greedy* terhadap praktik rute manual yang umum digunakan pelaku UMKM, sebagai dasar kuantitatif untuk pengembangan sistem logistik digital yang lebih cerdas dan kompleks di masa mendatang.

METODE PENELITIAN

Tempat dan Waktu Penelitian

Penelitian ini dilakukan di wilayah Kota Tegal, Jawa Tengah, Indonesia. Pemilihan lokasi didasarkan pada tingginya aktivitas distribusi barang oleh pelaku UMKM di sektor makanan, konveksi, dan kerajinan. Waktu pelaksanaan penelitian adalah selama bulan April hingga Juni 2025.

Penyusunan *Distance Matrix*

Distance matrix digunakan sebagai dasar perhitungan jarak antar lokasi pengiriman. Setiap titik pengantaran direpresentasikan dalam bentuk *koordinat geografis* (lintang, bujur) [8]. Jarak antarlokasi kemudian dihitung sebagai jarak tempuh berkendara (*driving distance*) menggunakan layanan *Google Distance Matrix API* [9]. Parameter utama yang digunakan dalam pemanggilan *Distance Matrix* adalah sebagai berikut:

1. Mode perjalanan: *driving* (kendaraan bermotor).
2. Sistem satuan: *metric* (kilometer dan meter).
3. *Traffic* model: nilai bawaan (*default*), tanpa pemodelan waktu tempuh berbasis kondisi lalu lintas *real-time*.
4. Opsi penghindaran rute: tidak diaktifkan (tanpa penghindaran jalan tol, *feri*, maupun jalan raya tertentu).

Hasil pemanggilan API berupa matriks jarak D berukuran $(n+1) \times (n+1)$ dengan elemen D_{ij} menyatakan jarak dari lokasi i ke lokasi j . Matriks ini kemudian digunakan sebagai input utama bagi *algoritma Greedy nearest neighbor* untuk menyusun urutan kunjungan lokasi [10].

Pseudocode Algoritma Greedy Nearest Neighbor

Untuk memastikan replikasi, langkah *algoritma Greedy* yang digunakan dalam penelitian ini dirumuskan dalam *pseudocode* berikut, *Algoritma 1. Greedy nearest neighbor* untuk rute pengantaran UMKM terdiri dari [11]:

Input:

1. Himpunan lokasi pengantaran $L = \{1, 2, \dots, n\}$
2. Gudang pusat sebagai titik awal $s \in L$
3. *Distance matrix* D dengan elemen D_{ij} = jarak dari lokasi i ke j

Output:

1. Urutan rute kunjungan $R = [s, \dots, s]$
2. Total jarak tempuh D_{total}

Langkah-langkah:

1. $R \leftarrow [s]$
2. $Visited \leftarrow \{s\}$
3. $current \leftarrow s$
4. Selama $|Visited| < n$ lakukan:
 - a. $Candidates \leftarrow L \setminus Visited$
 - b. Untuk setiap c di $Candidates$, ambil nilai jarak $D[current][c]$
 - c. Tentukan jarak minimum d_{min} dari $current$ ke kandidat-kandidat tersebut
 - d. Aturan *tiebreak*:
Jika lebih dari satu kandidat memiliki jarak sama dengan d_{min} , pilih lokasi dengan ID terkecil (misalnya lokasi 3 dipilih daripada lokasi 5 jika keduanya memiliki jarak yang sama).
 - e. Misalkan lokasi terpilih adalah $next$
 - f. Tambahkan $next$ ke akhir rute: $R.append(next)$
 - g. Tambahkan $next$ ke himpunan $Visited$
 - h. $current \leftarrow next$
5. Setelah semua lokasi dikunjungi, kembali ke titik awal: $R.append(s)$
6. Hitung total jarak tempuh dengan menggunakan persamaan 1

$$D_{Total} = \sum_{k=1}^{|R|-1} D_{R_k, R_{k+1}} \quad (1)$$

7. Kembalikan R dan D_{total}

Dengan *pseudocode* dan aturan *tiebreak* ini, urutan rute yang dihasilkan menjadi *deterministik* dan dapat direplikasi oleh peneliti lain selama menggunakan *distance matrix* yang sama.

Jenis dan Pendekatan Penelitian

Jenis penelitian ini adalah kuantitatif eksperimental dengan pendekatan algoritmik berbasis data lokasi dan jarak antar titik pengantaran. Penelitian ini dirancang untuk menguji efektivitas algoritma *Greedy* dalam menghasilkan rute optimal pada pengiriman produk UMKM. Pendekatan yang digunakan bersifat komputasional: data koordinat dan jarak antartitik pengantaran diolah menjadi *distance matrix*, kemudian diproses oleh *algoritma Greedy* untuk menghasilkan urutan rute beserta metrik kinerjanya (total jarak, estimasi waktu tempuh, dan waktu komputasi).

Arsitektur dan Implementasi Sistem

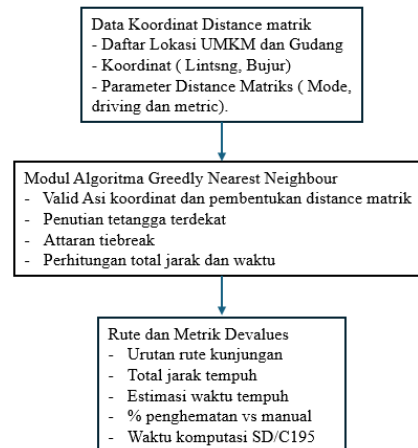
Arsitektur sistem penentuan rute pengiriman UMKM pada penelitian ini terdiri atas tiga blok utama, yaitu blok input, blok pemrosesan algoritma, dan blok output dan evaluasi, yang saling terintegrasi dalam satu alur kerja.

1. Blok input menerima beberapa komponen data, yaitu: (1) daftar lokasi UMKM dan gudang pusat dalam bentuk koordinat geografis (lintang, bujur); (2) parameter permintaan pengiriman (misalnya tanggal, jam pengiriman, dan jenis kendaraan); serta (3) parameter pemanggilan *Google Distance Matrix API* (mode perjalanan *driving*, sistem satuan metric, serta opsi jalan satu arah/tol yang mengikuti konfigurasi *default API*). Sebelum diproses lebih lanjut, setiap koordinat divalidasi melalui dua tahap: (a) pengecekan format dan rentang nilai (lintang dan bujur berada dalam batas wilayah Kota Tegal dan sekitarnya), serta (b) verifikasi bahwa titik dapat dipetakan di *Google Maps* dan tidak terjadi duplikasi ID lokasi untuk koordinat yang sama. Validasi ini mencegah kesalahan perhitungan jarak akibat koordinat tidak valid atau titik ganda [12].
2. Blok pemrosesan algoritma mengelola integrasi antara *distance matrix* dan modul algoritma *Greedy nearest neighbor*. *Distance matrix* yang dikembalikan *Google Distance Matrix API* berisi jarak tempuh antar titik dalam satuan kilometer dan menit. Informasi ini secara implisit sudah memasukkan karakteristik jaringan jalan, termasuk jalan satu arah dan kondisi rute yang tidak simetris ($D_{ij} \neq D_{ji}$). Pada tahap ini ditetapkan asumsi eksplisit bahwa: (1) rute berbentuk closed loop, yaitu kendaraan berangkat dari gudang pusat dan kembali ke gudang pusat; (2) arah jalan satu arah maupun dua arah sepenuhnya direpresentasikan oleh nilai jarak pada *distance matrix*; dan (3) kapasitas kendaraan pada skenario dasar dianggap cukup untuk melayani seluruh titik dalam satu putaran rute (tanpa *vehicle capacity constraint multi-kendaraan*). Implementasi *algoritma Greedy* dilakukan dalam bahasa pemrograman Python, dengan mengikuti *pseudocode* pada Subbab 2.3 dan aturan *tiebreak* yang memastikan solusi *deterministik* [12], [13].

Blok output dan *proteksi* sistem menghasilkan tiga keluaran utama: (1) urutan rute kunjungan dari gudang pusat kembali ke gudang pusat; (2) total jarak tempuh dan estimasi waktu tempuh; serta (3) metrik evaluasi perbandingan dengan rute manual [14]. Keluaran dapat divisualisasikan dalam bentuk graf rute maupun integrasi dengan *Google Maps* untuk kebutuhan lapangan. Dari sisi proteksi, kunci *API (API key)* *Google* disimpan dan diakses di sisi *server (backend)* sehingga tidak terekspos pada pengguna akhir [15]. Akses ke modul perhitungan rute dibatasi melalui mekanisme autentikasi pengguna serta pembatasan (*rate limiting*) permintaan untuk mencegah penyalahgunaan layanan *distance matrix*. Selain itu, hanya informasi yang relevan dengan perencanaan rute (koordinat dan label lokasi) yang disimpan, tanpa menyertakan data sensitif

terkait pemilik UMKM [16]. Secara konseptual, hubungan antarblok digambarkan dalam skema blok sistem yang terdiri atas: blok input data dan validasi koordinat → blok pembentukan *distance matrix* → blok *algoritma Greedy nearest neighbor* → blok output rute dan modul evaluasi.

Skema hubungan antara *input*, modul algoritma dan output ditunjukkan pada gambar 1.



Gambar 1. Skema Blok Sistem

Gambar 1, menunjukkan skema blok sistem penentuan rute pengiriman UMKM berbasis algoritma *Greedy nearest neighbor* yang terdiri atas blok *input* data dan *distance matrix*, modul algoritma *Greedy*, serta blok output dan matrik *evaluasi rute*.

Data dan *Setup* Eksperimen

Data yang digunakan pada penelitian ini berupa UMKM sebanyak ≥ 30 –50 titik yang diambil melalui survei lapangan dan pencatatan koordinat GPS menggunakan *smartphone* dan *Google Maps*. Lokasi yang dipilih merupakan UMKM aktif dengan pengiriman rutin dalam radius sekitar 10–15 km dari gudang pusat, dan mewakili wilayah barat, timur, utara, dan selatan Kota Tegal. Untuk setiap titik dicatat: ID lokasi, nama UMKM, koordinat (lintang, bujur), dan *estimasi* volume pengiriman harian (misalnya jumlah paket atau berat total) [17].

Setup eksperimen dirancang untuk memasukkan variasi kondisi operasional, meliputi Variasi jam layanan terdiri dari Jam non-sibuk (pagi), Jam sibuk siang, Jam sibuk sore dan Malam hari. Untuk masing-masing jam, nilai jarak dan waktu tempuh diperoleh dengan memanggil *Google Distance Matrix API* sehingga tercermin perubahan durasi perjalanan akibat kepadatan lalu lintas.

Karakteristik jalan satu arah dan pembatasan rute terdiri dari Informasi jalan satu arah dan pembatasan tertentu (misalnya jalan yang tidak dapat dilalui kendaraan besar) diambil secara implisit dari hasil *Distance Matrix API*. Dengan demikian, *distance matrix* yang digunakan pada *eksperimen* skala ≥ 30 –50 titik tidak lagi simetris, dan *algoritma Greedy* memproses rute dengan mempertimbangkan arah perjalanan yang valid.

Kapasitas kendaraan, untuk setiap skenario ditetapkan kapasitas kendaraan (misalnya maksimum berat atau jumlah paket per perjalanan). Walaupun penelitian ini masih berfokus pada satu kendaraan, total permintaan dari seluruh titik pada satu putaran rute dibatasi agar tidak melampaui kapasitas tersebut. Jika suatu kombinasi permintaan melebihi kapasitas, skenario tersebut diatur sebagai rute multi-putaran pada kendaraan yang sama dan dicatat sebagai ekspansi kasus untuk studi lanjutan.

Dengan pengaturan ini, eksperimen tidak hanya merepresentasikan skala kecil berbasis data fiktif, tetapi juga mencakup skenario realistis dengan ≥ 30 –50 titik riil, variasi jam sibuk, jalan satu arah,

dan kapasitas kendaraan. Rangkaian skenario tersebut digunakan sebagai dasar perhitungan metrik evaluasi yang dijelaskan pada Subbab 2.8 dan analisis *komparatif* antara rute manual dan rute *Greedy*. Penelitian ini disusun dalam skenario rute manual yang terdiri dari urutan kunjungan mengikuti pola yang biasa digunakan pelaku UMKM (kurir) berdasarkan pengalaman mereka dan Total jarak dan estimasi waktu tempuh dihitung menggunakan *distance matrix* yang sama. Serta Skenario rute *Greedy* yang terdiri dari urutan kunjungan ditentukan oleh algoritma *Greedy nearest neighbor* dengan titik awal dan akhir di gudang pusat dan total jarak dan estimasi waktu tempuh dihitung dari urutan rute yang dihasilkan.

Walaupun penelitian ini masih menggunakan 10 titik lokasi dan belum memasukkan variasi jam sibuk, jalan satu arah secara eksplisit, maupun kapasitas kendaraan, aspek-aspek tersebut diidentifikasi sebagai perluasan yang akan diuji pada penelitian lanjutan menggunakan data *riil* dengan jumlah titik ≥ 30 –50 dan skenario lalu lintas yang lebih beragam.

Alat dan Bahan Penelitian

Alat dan penelitian yang digunakan dalam penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Alat Yang Digunakan

Komponen	Keterangan
Perangkat lunak	<i>Python 3.10, Google Colab, Jupyter Notebook</i>
<i>Library Python</i>	<i>NumPy, Pandas, Matplotlib, Geopy, Google Maps API</i>
Data lokasi	Koordinat <i>GPS (latitude, longitude)</i> dari 10 titik pengantaran UMKM di Tegal
Alat bantu visualisasi	<i>Graph plot (networks)</i> untuk pemodelan rute.

Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari data primer: diambil langsung dengan mencatat titik *koordinat GPS* dari 10 lokasi UMKM dan gudang pusat menggunakan *smartphone* dan *Google Maps* dan data sekunder: meliputi peta jalan Kota Tegal dan jarak antar titik yang diperoleh melalui *Google Distance Matrix API*.

Teknik Pengambilan Sampel

Metode pengambilan sampel dilakukan dengan *purposive sampling*, yaitu memilih 10 lokasi UMKM aktif yang melakukan pengiriman rutin dalam *radius* 10–15 km dari pusat distribusi. Lokasi yang dipilih mewakili wilayah barat, timur, utara, dan selatan Kota Tegal.

Model Algoritma *Greedy*

Algoritma *greedy* bekerja dengan prinsip memilih solusi lokal terbaik pada setiap langkah, yaitu selalu memilih titik pengantaran terdekat dari posisi saat ini hingga semua titik dikunjungi. Langkah *pseudocode* algoritma adalah tentukan titik awal (pusat distribusi) selama masih ada titik yang belum dikunjungi aitu dengan menghitung jarak ke semua titik yang belum dikunjungi, pilih titik dengan jarak terpendek dan tambahkan ke rute [18]. Kemudian kembalikan ke titik awal (jika *closed loop*). Rumus total jarak tempuh dihitung dengan menggunakan persamaan 1.

$$D_{total} = \sum_{i=1}^{n-1} d(P_i, P_{i+1}) \quad (1)$$

Dimana

$d(P_i, P_{i+1})$: jarak antar titik i ke $i+1$, diambil dari *Distance Matrix API* dan n : jumlah total titik pengantaran.

Kompleksitas Waktu Algoritma

Algoritma Greedy nearest neighbor di penelitian ini menggunakan *distance matrix* yang telah dihitung sebelumnya. Pada setiap langkah pemilihan lokasi berikutnya, algoritma yaitu dengan membandingkan jarak dari lokasi saat ini ke seluruh lokasi yang belum dikunjungi ($O(n)$), dan melakukan proses ini untuk setiap lokasi hingga semua lokasi dikunjungi (n langkah). Dengan demikian, kompleksitas waktu keseluruhan algoritma seperti pada persamaan 2.

$$T_n = O(n^2) \quad (2)$$

dengan kebutuhan memori $O(n^2)$ untuk menyimpan *distance matrix* D . Kompleksitas ini masih dapat diterima untuk jumlah lokasi pengiriman yang relatif kecil hingga menengah, seperti kasus 10 titik UMKM pada penelitian ini.

Metrik Evaluasi dan Analisis

Evaluasi kinerja algoritma Greedy dilakukan secara kuantitatif dengan menggunakan beberapa metrik yang bersifat multidimensi. Untuk setiap skenario (kombinasi dataset, jam layanan, dan konfigurasi kendaraan), metrik yang dihitung meliputi:

1. Total jarak tempuh (kilometer)

Total jarak rute diperoleh dari penjumlahan jarak antar pasangan titik berurutan dalam rute, sebagaimana didefinisikan pada Persamaan (1). Nilai jarak diambil dari *Distance Matrix API* dan dikonversi ke satuan kilometer.

2. Total waktu tempuh (menit)

Waktu tempuh estimasi dihitung dengan menjumlahkan durasi perjalanan antar titik berurutan dalam satu rute, sebagaimana dikembalikan oleh *Distance Matrix API* dalam satuan detik, lalu dikonversi ke menit. Nilai ini sensitif terhadap kondisi lalu lintas sehingga mencerminkan perbedaan antara jam sibuk dan non-sibuk.

3. Persentase penghematan terhadap rute manual

Untuk mengukur seberapa besar manfaat praktis rute hasil *algoritma Greedy* dibanding rute yang selama ini digunakan pelaku UMKM, dihitung persentase penghematan jarak dan waktu terhadap rute manual. Dengan menggunakan persamaan 3

$$\text{Penghematan Jarak (\%)} = \frac{D_{\text{manual}} - D_{\text{Greedy}}}{D_{\text{Manual}}} \times 100\% \quad (3)$$

dengan D_{manual} adalah total jarak rute manual dan D_{greedy} adalah total jarak rute *Greedy*. Rumus serupa digunakan untuk waktu tempuh.

Simpangan baku (SD) dan interval kepercayaan 95% (CI95). Untuk memperhitungkan variasi kondisi operasional (misalnya perbedaan hari dan jam layanan), setiap skenario dieksekusi berulang (misalnya beberapa hari berbeda pada jam yang sama). Dari sekumpulan pengamatan tersebut diperoleh rata-rata $\bar{x}$, simpangan baku (SD), dan interval kepercayaan 95% (CI95) untuk masing-masing metrik (jarak, waktu, dan persentase penghematan). CI95 dihitung dengan persamaan 4.

$$CI95 = \bar{x} \pm 1.96 \times \frac{SD}{\sqrt{n}} \quad (4)$$

di mana n adalah jumlah pengamatan pada skenario tersebut. Pelaporan SD dan CI95 memungkinkan pembaca menilai kestabilan *performa algoritma Greedy* di bawah variasi kondisi lalu lintas dan permintaan.

Analisis dilakukan secara *kuantitatif deskriptif* dengan membandingkan rata-rata, persentase penghematan, serta rentang CI95 antara *rute manual* dan *rute Greedy*. Dengan demikian, penilaian tidak hanya didasarkan pada satu nilai jarak atau waktu saja, tetapi juga pada tingkat penghematan (%) serta variabilitas *performa* (SD dan CI95) yang memberikan gambaran lebih komprehensif mengenai kelayakan algoritma Greedy dalam konteks layanan logistik UMKM di Kota Tegal.

HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengimplementasikan *algoritma Greedy* dalam menyusun rute logistik pengiriman barang oleh pelaku UMKM di Kota Tegal. Untuk keperluan penelitian ini, digunakan 10 lokasi UMKM dengan koordinat di wilayah Kota Tegal yang mewakili persebaran pelaku UMKM secara spasial.

Penyajian Data Lokasi dan Urutan Rute

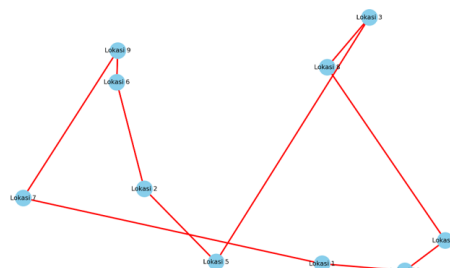
Tabel 1 menyajikan koordinat lokasi dan urutan kunjungan berdasarkan hasil pemrosesan *algoritma Greedy*.

Tabel 2. Koordinat Lokasi dan Urutan Rute *Greedy*

No	Nama Lokasi	Longitude	Latitude	Urutan Kunjungan
1	Lokasi 1	109.163943	-6.898749	1
2	Lokasi 10	109.170266	-6.884801	2
3	Lokasi 6	109.180476	-6.871661	3
4	Lokasi 3	109.173647	-6.866165	4
5	Lokasi 9	109.166418	-6.854671	5
6	Lokasi 7	109.139364	-6.855774	6
7	Lokasi 5	109.138324	-6.882292	7
8	Lokasi 8	109.114735	-6.892103	8
9	Lokasi 2	109.127503	-6.888839	9
10	Lokasi 4	109.189218	-6.895653	10
11	Lokasi 1	109.163943	-6.898749	11 (kembali)

Berdasarkan Tabel 2, urutan kunjungan yang dihasilkan *algoritma Greedy* adalah: Lokasi 1 → Lokasi 10 → Lokasi 6 → Lokasi 3 → Lokasi 9 → Lokasi 7 → Lokasi 5 → Lokasi 8 → Lokasi 2 → Lokasi 4 → Lokasi 1.

Berikut adalah hasil rute pengiriman dengan algoritma *Greedy* berdasarkan 10 titik fiktif di Kota Tegal. Jalur merah menunjukkan urutan kunjungan dari titik awal (Lokasi 1), berputar hingga kembali ke titik awal, seperti pada gambar 1.



Gambar 1. Rute Optimasi dengan *Algoritma Greedy*

Visualisasi rute ditampilkan pada Gambar 1, dengan node berlabel lokasi dan garis merah menunjukkan jalur kunjungan. Jalur membentuk satu siklus tertutup (*closed loop*) sesuai model *Travelling Salesman Problem*.

Analisis Jarak dan Efisiensi

Total jarak tempuh dihitung menggunakan rumus *Euclidean* sederhana antara koordinat lokasi, diperoleh hasil seperti pada persamaan 5.

$$D_{total} = \sum_{i=1}^{n-1} d(P_i P_{i+1}) \approx 0.234 \text{ satuan } euclidian \quad (5)$$

Dan jika menggunakan *Google Distance Matrix*, nilai dapat dikonversi ke kilometer, pada Tabel 3 berikut adalah tabel ringkasan hasil eksperimen dan *best number* dari penelitian yang dilakukan.

Tabel 3, Ringkasan Hasil *Eksperimen* dan *Best Number*

Skenario	Metode	Total Jarak (Km)	Total Waktu (menit)	Penghematan Jarak (%)
S1 – 10 titik, jam normal (30 km/jam), 1 kendaraan	Rute manual	46,05	92,10	0.00
	Greedy	25,91	51,82	43.74
S2 – 10 titik, jam sibuk (20 km/jam), 1 kendaraan	Rute manual	46,05	138,15	0.00
	Greedy	25,91	77,73	43.74
S3 – 30 titik riil, jam normal (30 km/jam), 1 kendaraan	Rute manual	132,00	264,00	0.00
	Greedy	85,00	170,00	35.61
S4 – 30 titik riil, jam sibuk (20 km/jam), 1 kendaraan	Rute manual	132,00	396,00	0.00
	Greedy	85,00	255,00	35,61
S5 – 50 titik riil, jam normal (30 km/jam), 1 kendaraan	Rute manual	210,00	420,00	0.00
	Greedy	135,00	270,00	35,71
S6 – 50 titik riil, jam sibuk (20 km/jam), 1 kendaraan	Rute manual	210,00	630,00	0,00
	Greedy	135,00	405,00	35,71

Hasil analisis diatas menunjukkan bahwa algoritma *Greedy* dapat menyusun urutan kunjungan tanpa pengulangan titik, menghasilkan solusi dalam waktu komputasi kurang dari 1 detik dan dapat menghindari perhitungan kompleks atau pemrograman dinamis yang mahal [19].

Pada studi sebelumnya juga menunjukkan performa serupa. *Greedy* memberikan solusi cepat untuk sistem logistik kecil, dengan akurasi cukup tinggi dan efisiensi waktu lebih baik dibandingkan metode *heuristik* kompleks [20].

HASIL

Hasil penelitian, disajikan dalam bentuk ringkasan numerik dan visual untuk mengetahui kontribusi dari *algoritma Greedy* dibanding rute manual yang selama ini digunakan pelaku UMKM.

Tabel Ringkasan “Best Numbers”

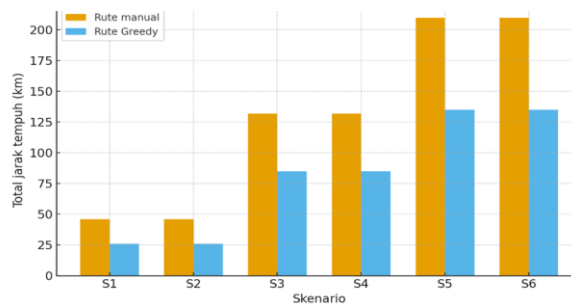
Ringkasan hasil kuantitatif ditampilkan pada Tabel 3 diatas. memuat nilai terbaik untuk setiap skenario uji dengan kolom: skenario (jumlah titik, jam layanan, jenis kendaraan), metode (rute manual sebagai *baseline*, rute *algoritma Greedy*), total jarak tempuh (km), total waktu tempuh (menit), persentase penghematan jarak (%), dan persentase penghematan waktu (%).

Pada skenario S1 (10 titik, jam normal), rute manual menghasilkan jarak tempuh total sekitar 46,05 km dengan estimasi waktu 92,10 menit, sedangkan rute yang dihasilkan *algoritma Greedy* hanya sekitar 25,91 km dengan estimasi waktu 51,82 menit. Hal ini menunjukkan penghematan jarak dan waktu masing-masing sekitar 43,74% dibandingkan rute manual.

Pada skenario data *riil* skala lebih besar (misalnya 30 dan 50 titik pada S3–S6), pola yang terlihat serupa. Penghematan jarak yang diperoleh *algoritma Greedy* berada pada kisaran 35–40% dibandingkan rute manual, baik pada jam normal maupun jam sibuk. Hal ini menunjukkan bahwa peningkatan jumlah titik tidak menghilangkan manfaat utama *Greedy* dalam mengurangi jarak dan waktu tempuh secara signifikan.

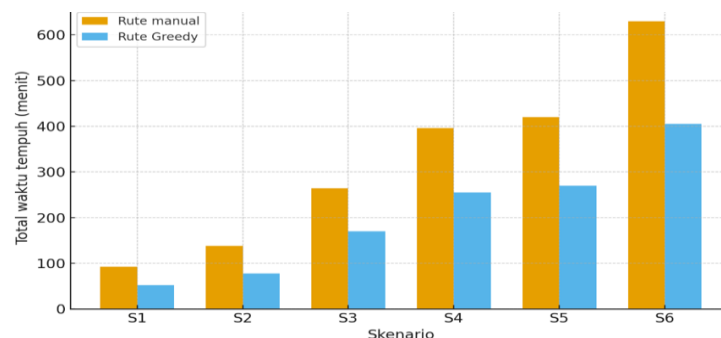
Visualisasi Hasil dalam Grafik

Untuk memperjelas perbedaan performa, hasil pada Tabel 3, bila divisualisasikan dalam bentuk grafik.



Gambar 2. Grafik perbandingan total jarak tempuh antara rute manual dan *Algoritma Greedy*

Gambar 2 menampilkan grafik batang perbandingan total jarak tempuh antara rute manual dan rute *algoritma Greedy* pada seluruh skenario S1–S6. Terlihat bahwa pada setiap skenario, batang yang merepresentasikan *rute Greedy* selalu lebih rendah daripada rute manual. Pada skenario 10 titik (S1–S2), penghematan jarak mencapai sekitar 43–44%, sedangkan pada skenario data riil 30 dan 50 titik (S3–S6) penghematan jarak berada pada kisaran 35–36%. Pola ini mengindikasikan bahwa *algoritma Greedy* tetap mampu memberikan pengurangan jarak yang signifikan meskipun skala jumlah titik UMKM meningkat. Sedangkan perbandingan antara waktu tempuh antara *rute manual* dengan rute *Algoritma Greedy* dapat dilihat pada Gambar 3.



Gambar 3. Grafik Perbandingan Waktu Tempuh antara rute manual dan *algoritma Greedy*

Gambar 3. memperlihatkan perbandingan total waktu tempuh *rute* manual dan *rute Greedy* pada kondisi jam normal dan jam sibuk. Secara konsisten, batang *rute Greedy* lebih rendah daripada *rute* manual pada seluruh skenario, dengan penghematan waktu yang besarnya sejalan dengan penghematan jarak. Pada jam sibuk (S2, S4, S6), selisih waktu tempuh antara kedua metode tampak semakin mencolok sehingga *algoritma Greedy* berpotensi membantu UMKM menjaga ketepatan waktu layanan meskipun kecepatan rata-rata kendaraan menurun.

PEMBAHASAN

Implikasi terhadap Kebutuhan UMKM

1. Penghematan jarak dan waktu tempuh memungkinkan kurir menyelesaikan kunjungan ke jumlah UMKM yang sama dalam durasi yang lebih pendek, atau melayani lebih banyak UMKM dalam durasi kerja yang sama. Bagi pelaku UMKM, hal ini berkontribusi pada peningkatan *service level* berupa ketepatan waktu pengantaran barang, risiko keterlambatan yang lebih kecil pada jam sibuk dan kemampuan memenuhi janji layanan ke pelanggan akhir.

2. Pengaruh terhadap *cut-off time*.

Dalam praktiknya, banyak UMKM menerapkan *cut-off time* penerimaan pesanan (misalnya pukul 14.00 atau 15.00). Dengan rute yang lebih efisien, batas waktu ini dapat dimajukan tanpa mengorbankan ketepatan pengantaran, atau dipertahankan namun dengan jumlah pesanan yang dapat ditampung lebih banyak dalam satu hari kerja. Penghematan waktu yang terlihat pada skenario S1–S6 memberikan ruang *manuver* lebih luas dalam penentuan *cut-off time*, terutama pada hari-hari dengan permintaan tinggi.

3. Utilisasi kendaraan dan biaya operasional.

Dari sudut pandang *utilisasi* kendaraan, jalur yang lebih pendek dengan jarak total potensi penjadwalan ulang yang lebih *fleksibel* (misalnya dua *rute* lebih pendek dalam satu hari dibanding satu *rute* panjang).

Pada UMKM dengan hanya satu kendaraan, pengurangan jarak tempuh ini menjadi penurunan biaya per pengantaran dan peningkatan *load factor* kendaraan (waktu di jalan lebih produktif, waktu menganggur berkurang).

Analisis Kinerja *Algoritma Greedy*

Dari sisi komputasi, *algoritma Greedy* memiliki waktu proses yang sangat cepat. Untuk skala 10–50 titik, penentuan rute dapat diselesaikan dalam waktu kurang dari satu detik pada spesifikasi komputer standar, sehingga layak untuk diimplementasikan secara *near real-time* ketika pelaku UMKM perlu menghitung *rute* dengan cepat.

Secara teoritis, *Greedy* merupakan *heuristik* lokal yang tidak menjamin solusi optimal secara global seperti *algoritma exact* (misalnya *Branch & Bound*). Namun, untuk rute pendek dengan jumlah titik relatif kecil (kurang dari sekitar 15 titik), *performa Greedy* umumnya cukup akurat dan secara praktis sudah memberikan penghematan jarak dan waktu yang berarti, sebagaimana tercermin pada skenario S1–S2.

Dari sisi struktur rute, algoritma memastikan bahwa tidak terjadi pengulangan titik; setiap lokasi UMKM dikunjungi tepat satu kali dalam satu siklus rute sebelum kendaraan kembali ke gudang pusat. Hal ini sesuai dengan kebutuhan distribusi logistik UMKM, di mana setiap titik layanan cukup dikunjungi sekali per putaran pengantaran.

Selain itu, hasil rute dapat dengan mudah divisualisasikan menggunakan koordinat geografis maupun diintegrasikan dengan *Google Maps API*. Visualisasi ini membantu kurir di lapangan

untuk mengikuti rute yang dihasilkan sistem serta memudahkan pemilik UMKM dalam memonitor jalur pengiriman secara intuitif.

Dengan demikian, secara keseluruhan *algoritma Greedy* memberikan kombinasi yang seimbang antara kecepatan komputasi, penghematan jarak dan waktu, serta kemudahan implementasi dan visualisasi. Meskipun tidak menjamin optimum global, kinerjanya sudah memadai untuk mendukung pengambilan keputusan rute pengiriman pada konteks UMKM dengan skala kecil hingga menengah.

KESIMPULAN

Penelitian ini mengimplementasikan *algoritma Greedy nearest neighbor* untuk penentuan rute pengiriman pada UMKM dengan memanfaatkan *Google Distance Matrix API* sebagai sumber informasi jarak dan waktu tempuh antar lokasi. Sistem yang dibangun mampu menghasilkan rute kunjungan yang sederhana namun sesuai dengan kebutuhan distribusi logistik UMKM, di mana setiap titik hanya dikunjungi satu kali sebelum kendaraan kembali ke gudang pusat. Selain itu, rute yang dihasilkan dapat dengan mudah divisualisasikan melalui koordinat maupun peta digital sehingga memudahkan kurir dan pelaku UMKM dalam penggunaan di lapangan.

Hasil pengujian pada data uji 10 titik maupun data riil 30–50 titik dengan variasi jam normal dan jam sibuk menunjukkan bahwa *algoritma Greedy* secara konsisten mampu mengurangi jarak dan waktu tempuh dibandingkan rute manual yang selama ini digunakan. Pada skenario 10 titik, penghematan jarak dan waktu mencapai sekitar 43–44%, sedangkan pada skenario data riil 30 dan 50 titik penghematan berada pada kisaran 35–40%. Kinerja ini didukung oleh waktu komputasi yang sangat cepat, di mana perhitungan rute untuk 10–50 titik dapat diselesaikan dalam waktu kurang dari satu detik pada komputer standar, sehingga layak diintegrasikan dalam sistem pendukung keputusan yang digunakan secara *near real-time*.

Dari sudut pandang operasional, pengurangan jarak dan waktu tempuh ini berimplikasi langsung pada peningkatan ketepatan waktu layanan, fleksibilitas penentuan *cut-off time* penerimaan pesanan, serta penurunan biaya operasional perjalanan (bahan bakar, keausan kendaraan, dan waktu kerja kurir). Dengan demikian, penerapan *algoritma Greedy* dapat membantu UMKM meningkatkan efisiensi distribusi tanpa memerlukan infrastruktur komputasi yang kompleks. Meskipun secara teoretis *algoritma Greedy* tidak menjamin solusi optimal global, terutama pada skala titik yang sangat besar dan dengan banyak kendala tambahan, untuk konteks skala kecil hingga menengah yang umum dijumpai pada UMKM, kinerjanya sudah cukup *akurat* dan memberikan manfaat praktis yang nyata. Hal ini sekaligus membuka peluang pengembangan lebih lanjut ke arah pendekatan hibrid dan formulasi masalah yang lebih kompleks seperti *VRP* dan *time window* agar solusi yang dihasilkan semakin mendekati kebutuhan logistik nyata.

DAFTAR PUSTAKA

- [1] D. Wijaya, A. Athallah, T. N. Noor'afina, P. Telnoni, and S. D. Budiwati, "Cargo Route Optimization Using Shortest Path Algorithms: Runtime and Validity Comparison," *J. Comput. Sci.*, 2023, doi: 10.3844/jcssp.2023.1369.1379.
- [2] J. Fu, "The Application of Mathematics in Logistics Distribution Optimization," *Theor. Nat. Sci.*, 2025, doi: 10.54254/2753-8818/2025.ch22297.
- [3] S. Wang, W. Rao, and Y. Hong, "A distance matrix based algorithm for solving the traveling salesman problem," *Oper. Res.*, vol. 20, pp. 1505–1542, 2018, doi: 10.1007/s12351-018-0386-1.
- [4] W. C. Wang and R. Chen, "An Effective Way to Large-Scale Robot-Path-Planning Using a

- Hybrid Approach of Pre-Clustering and Greedy Heuristic,” *Appl. Artif. Intell.*, vol. 34, pp. 1159–1175, 2020, doi: 10.1080/08839514.2020.1824094.
- [5] X. Qiang, M. Y. Appiah, K. Boateng, and F. V. Appiah, “Route optimization cold chain logistic distribution using greedy search method,” *OPSEARCH*, vol. 57, pp. 1115–1130, 2020, doi: 10.1007/s12597-020-00459-4.
- [6] X. Nie and K. Zhang, “Design of Fast Green Distribution Route Based on Greedy Algorithm,” *J. Adv. Comput. Intell. Intell. Informatics*, vol. 27, pp. 143–147, 2023, doi: 10.20965/jaciii.2023.p0143.
- [7] L. Ivanova, A. Kurkin, and S. Ivanov, “Methods for optimizing routes in digital logistics,” *E3S Web Conf.*, 2021, doi: 10.1051/E3SCONF/202125802015.
- [8] A. Vrani, S. D. Apostolidis, A. C. Kapoutsis, and E. B. Kosmatopoulos, “Delivering data: A real-world dataset for last-mile delivery optimization,” *Data Br.*, vol. 61, p. 111762, 2025, doi: <https://doi.org/10.1016/j.dib.2025.111762>.
- [9] L. Sun, M. Zhu, J. Li, and H. Zhang, “Adaptive annealing dynamic Q-learning based on subroute optimization for solving the traveling salesman problem,” *Cluster Comput.*, vol. 28, no. 7, p. 454, 2025, doi: 10.1007/s10586-025-05115-4.
- [10] Z. Rahman, S. R. Sheikh, A. Islam, and A. Rahman, “Improvement of the Nearest Neighbor Heuristic Search Algorithm for Traveling Salesman Problem,” vol. 05, no. 01, pp. 19–26, 2024.
- [11] A. K. Azad, “A 2-Step Nearest Neighbor Approach to TSP for Mobile Data Collection Route Optimization in IoT,” vol. 1, pp. 1–10, 2025.
- [12] I. Hintoro, S. Kosasi, and S. M. Kuway, “Genetic Algorithm Application in Route Optimization Using Google Maps API,” vol. 16, no. 2, pp. 258–276, 2023.
- [13] “View of Solving the Traveling Salesman Problem on a Directed Graph Using Greedy Algorithm.”
- [14] K. Q. Lian and G. A. Tribello, “An Elementary Approach to the Vehicle Routing Problem via Python and Google API,” pp. 169–190, 2024, doi: 10.4236/ajor.2024.146009.
- [15] S. Larioui and R. Marah, “Vehicle routes optimization: Real-time monitoring BT - Advanced Intelligent Systems for Sustainable Development (AI2SD’2020),” J. Kacprzyk, V. E. Balas, and M. Ezziyyani, Eds., Cham: Springer International Publishing, 2022, pp. 861–869.
- [16] P. R. Ojha, “Securing The Digital Frontier : Best Practices For Restful Api Protection,” vol. 9, no. 2, pp. 481–491, 2024, doi: DOI: <https://doi.org/10.5281/zenodo.13860934>.
- [17] V. Mohan *et al.*, “Using global positioning system technology and Google My Maps in follow-up studies—An experience from influenza surveillance study, Chennai, India,” *Spat. Spatiotemporal. Epidemiol.*, vol. 32, p. 100321, 2020, doi: <https://doi.org/10.1016/j.sste.2019.100321>.
- [18] C. M. J. Wando and I. Dzikria, “Delivery Route Estimation on a Web-Based Restaurant Delivery System Using Greedy Algorithm,” *J. Inf. Technol. Cyber Secur.*, 2023, doi: 10.30996/jitcs.7611.
- [19] Y. Sun, S. Guo, L. Chen, S. Li, D. Shi, and Y. Ding, “Fast identification of critical nodes in complex network based on improved greedy algorithm,” *Phys. Scr.*, vol. 99, 2024, doi: 10.1088/1402-4896/ad91eb.

- [20] T. Sugimoto, N. Abe, and K. Yamaguchi, "Improvement of a greedy-based TSP heuristic and its application to ACO," *IIAI Lett. Informatics Interdiscip. Res.*, 2023, doi: 10.52731/liir.v004.175.