

Optimalisasi Rute Transportasi: Studi Kasus Algoritma Greedy Menggunakan Bahasa Pemrograman Python

Sovantri Putra Paskah Halawa¹ Zulfahmi Indra² Fhadillah Br Hutagalung³ Najwa Latifah Hasibuan⁴

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan, Kota Medan, Sumatera Utara, Indonesia^{1,2,3,4}

Email: sovanhalawa06@gmail.com¹ zulfahmi.indra@unimed.ac.id²
audyFhadillah26@gmail.com³ najwalatifahhasibuan@gmail.com⁴

Abstrak

Masalah optimalisasi rute transportasi merupakan tantangan penting dalam berbagai aplikasi dunia nyata, seperti logistik, pengiriman barang, dan perencanaan transportasi. Salah satu pendekatan yang sering digunakan untuk menyelesaikan masalah ini adalah algoritma Greedy, yang menawarkan efisiensi waktu komputasi dengan cara membuat keputusan lokal terbaik pada setiap langkahnya. Meskipun algoritma ini tidak selalu memberikan solusi yang optimal, kesederhanaannya menjadikan pilihan yang menarik untuk aplikasi skala besar. Penelitian ini berfokus pada penerapan algoritma Greedy dalam mengoptimalkan rute transportasi, dengan menggunakan bahasa pemrograman Python sebagai platform implementasi. Studi kasus yang digunakan dalam penelitian ini melibatkan pengoptimalan rute untuk pengiriman barang ke sejumlah titik lokasi, di mana tujuan utama adalah meminimalkan total jarak tempuh dan waktu perjalanan. Implementasi algoritma Greedy dilakukan dengan menguji berbagai ukuran dan kompleksitas data rute transportasi.

Kata Kunci: Optimalisasi Rute Transportasi, Algoritma Greedy, Python.



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

PENDAHULUAN

Dalam era globalisasi saat ini, transportasi memainkan peran yang sangat penting dalam mendukung berbagai aspek kehidupan, terutama dalam bidang logistik dan distribusi barang. Efisiensi sistem transportasi berdampak langsung pada biaya operasional perusahaan, kepuasan pelanggan, serta dampak lingkungan. Optimisasi rute transportasi menjadi semakin relevan seiring dengan meningkatnya volume pengiriman, baik itu dalam skala lokal maupun global, terutama dalam industri e-commerce dan pengiriman barang. Pengaturan rute yang baik dapat mengurangi biaya bahan bakar, meminimalkan waktu pengiriman, serta mengurangi emisi karbon, yang pada gilirannya berkontribusi pada keberlanjutan lingkungan. Salah satu pendekatan untuk mengatasi masalah optimisasi rute transportasi adalah melalui algoritma Greedy. Algoritma ini merupakan salah satu metode penyelesaian masalah yang sering digunakan karena kemampuannya untuk melakukan komputasi dengan cepat dan sederhana. Algoritma Greedy bekerja dengan prinsip membuat keputusan optimal secara lokal pada setiap langkah, dengan harapan bahwa keputusan tersebut akan mengarah pada solusi optimal secara keseluruhan. Namun, meskipun algoritma ini sering kali menghasilkan solusi yang mendekati optimal, ada kemungkinan bahwa solusi yang dihasilkan bukan yang terbaik secara global, terutama dalam masalah-masalah yang kompleks dan memiliki banyak variabel.

Penelitian ini berfokus pada penerapan algoritma Greedy dalam konteks optimalisasi rute transportasi. Masalah yang akan dianalisis adalah pengiriman barang ke sejumlah titik lokasi yang berbeda, dengan tujuan untuk meminimalkan total jarak tempuh dan waktu perjalanan. Implementasi algoritma dilakukan menggunakan bahasa pemrograman Python, yang dipilih karena kemampuannya dalam menangani dataset yang besar, serta dukungan pustaka-pustaka

yang kuat seperti NumPy dan Matplotlib untuk analisis dan visualisasi data. Python juga menyediakan fleksibilitas dalam pengembangan dan pengujian algoritma, menjadikannya alat yang sangat berguna dalam penelitian ini. Meskipun algoritma Greedy terkenal dengan efisiensinya, dalam beberapa kasus, pendekatan ini mungkin memerlukan modifikasi atau perlu digabungkan dengan metode lain untuk menghasilkan solusi yang lebih baik. Oleh karena itu, penelitian ini tidak hanya bertujuan untuk menerapkan algoritma Greedy dalam optimisasi rute transportasi, tetapi juga untuk mengevaluasi sejauh mana algoritma ini efektif dalam menyelesaikan masalah dengan skala yang lebih besar dan kompleksitas yang lebih tinggi. Melalui serangkaian pengujian pada data yang bervariasi, penelitian ini diharapkan dapat memberikan wawasan lebih lanjut mengenai keunggulan dan keterbatasan algoritma Greedy dalam konteks dunia nyata.

Rumusan Masalah

Berdasarkan pembahasan di atas, rumusan masalah dalam penelitian ini meliputi:

1. Bagaimana penerapan algoritma Greedy dapat membantu dalam optimalisasi rute transportasi, khususnya dalam hal pengiriman barang ke sejumlah lokasi yang berbeda?
2. Apakah algoritma Greedy mampu meminimalkan total jarak tempuh dan waktu perjalanan dalam skenario transportasi dengan data rute yang bervariasi dalam ukuran dan kompleksitas?
3. Seberapa efektif algoritma Greedy dalam menangani masalah optimalisasi rute transportasi pada skala besar dibandingkan dengan metode lain?
4. Apa saja kendala yang dihadapi dalam penerapan algoritma Greedy untuk masalah optimalisasi rute transportasi, dan bagaimana solusi untuk mengatasinya?

METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahap yang meliputi perancangan, implementasi, dan evaluasi dari penerapan algoritma Greedy dalam optimalisasi rute transportasi. Tahapan penelitian ini dijelaskan secara rinci sebagai berikut:

1. Pengumpulan Data. Data yang digunakan dalam penelitian ini berupa dataset lokasi yang berisi koordinat geografis sejumlah titik pengiriman barang. Dataset ini akan mencakup:
 - a. Jumlah titik pengiriman: Dataset terdiri dari berbagai ukuran, mulai dari beberapa titik lokasi hingga ratusan titik, untuk menguji performa algoritma pada skala yang berbeda.
 - b. Jarak antar titik: Jarak dihitung berdasarkan koordinat titik-titik pengiriman menggunakan rumus Euclidean untuk mencerminkan jarak fisik sebenarnya di antara titik-titik tersebut.

Data ini bisa dihasilkan secara sintetik atau diambil dari dataset rute transportasi yang sudah tersedia di sumber terbuka.

2. Perancangan Algoritma Greedy. Pada tahap ini, algoritma Greedy dirancang untuk menyelesaikan masalah optimisasi rute transportasi. Langkah-langkah dalam algoritma Greedy untuk masalah ini adalah sebagai berikut:
 - a. Inisialisasi titik awal: Memulai dari satu titik pengiriman, yaitu titik pertama dalam dataset.
 - b. Pemilihan rute berikutnya: Pada setiap langkah, algoritma memilih titik pengiriman terdekat yang belum dikunjungi.
 - c. Pengulangan hingga semua titik dikunjungi: Proses ini diulangi sampai semua titik pengiriman telah dilalui.

- d. Kembali ke titik awal: Setelah semua titik dikunjungi, kendaraan akan kembali ke titik awal, menyelesaikan perjalanan.
3. Implementasi Algoritma dalam Python. Algoritma Greedy diimplementasikan menggunakan bahasa pemrograman Python. Berikut ini beberapa pustaka yang digunakan dalam implementasi: NumPy, untuk menangani operasi matematika, seperti perhitungan jarak antara titik-titik pengiriman. Matplotlib, untuk visualisasi rute yang dihasilkan dari implementasi algoritma, dan Pandas untuk pengelolaan dataset.
4. Analisis Hasil Simulasi. Pengujian dilakukan dengan menggunakan dataset lokasi yang beragam untuk memeriksa efektivitas algoritma Greedy dalam meminimalkan total jarak tempuh dan waktu perjalanan. Tahap pengujian meliputi Pengukuran performa: Performa algoritma diukur berdasarkan total jarak tempuh dan waktu komputasi. Semakin pendek jarak yang dihasilkan dan semakin cepat waktu komputasi, maka semakin efektif algoritma. Analisis error: Dilakukan perbandingan hasil algoritma Greedy dengan hasil optimal (bila diketahui), atau solusi dari metode lain seperti Algoritma Dijkstra atau Algoritma Brute Force, untuk mengukur seberapa jauh hasil Greedy dari solusi optimal.
5. Evaluasi Efektivitas. Pada tahap evaluasi, efektivitas algoritma Greedy dinilai dengan mempertimbangkan beberapa faktor: Efisiensi waktu komputasi, Waktu yang diperlukan oleh algoritma untuk menyelesaikan rute pengiriman, terutama saat dataset bertambah besar. Akurasi solusi: Kemampuan algoritma dalam mendekati solusi optimal (minimal total jarak tempuh) pada berbagai kondisi skenario pengiriman. Skalabilitas: Kemampuan algoritma untuk menangani dataset dengan skala besar.

HASIL PENELITIAN DAN PEMBAHASAN

Pada penelitian ini, kami menggunakan algoritma Greedy untuk menyelesaikan masalah optimalisasi rute transportasi pada dataset yang terdiri dari 5 lokasi pengiriman. Algoritma Greedy dipilih karena mampu memberikan solusi yang cepat dan sederhana untuk menemukan rute dengan total jarak tempuh terendah, meskipun tidak selalu optimal secara global. Dalam simulasi ini, kami berfokus pada minimisasi jarak tempuh dalam skenario pengiriman barang dari satu titik awal (depot) ke beberapa titik tujuan, kemudian kembali lagi ke titik awal.

Dataset Pengiriman Barang

Kami menggunakan dataset yang terdiri dari 5 koordinat 5 lokasi pengiriman. Lokasi-lokasi tersebut diatur dalam sistem koordinat kartesian (x,y) sebagai berikut:

Tabel 1.	
Lokasi	Koordinat (x,y)
0 (depot)	(0,0)
1	(2,3)
2	(5,4)
3	(7,1)
4	(6,3)
5	(1,6)

Simulasi Algoritma Greedy

Algoritma Greedy digunakan untuk mencari rute terdekat yang harus ditempuh oleh kendaraan pengirim barang. Pada setiap langkah, algoritma memilih lokasi terdekat yang belum dikunjungi hingga semua lokasi telah dikunjungi, lalu kembali ke depot. Total jarak tempuh dihitung dari hasil rute yang diperoleh.

Implementasi Algoritma Greedy

Berikut adalah implementasi kode Python yang digunakan dalam simulasi ini:

```
import numpy as np
# Definisikan koordinat lokasi pengiriman (x, y)
locations = np.array([[0, 0], # Lokasi 0 (depot)
[2, 3], # Lokasi 1
[5, 4], # Lokasi 2
[7, 1], # Lokasi 3
[6, 3], # Lokasi 4
[1, 6]]) # Lokasi 5

# Fungsi untuk menghitung jarak Euclidean antara dua titik
def calculate_distance(p1, p2):
    return np.sqrt(np.sum((p1 - p2)**2))

# Fungsi untuk menemukan rute terdekat menggunakan algoritma Greedy
def greedy_route(locations):
    n = len(locations)
    visited = [False] * n
    route = [0] # Memulai dari lokasi pertama (index 0)
    visited[0] = True
    total_distance = 0

    current_location = 0
    for i in range(n - 1):
        min_distance = float('inf')
        next_location = -1
        # Cari lokasi terdekat yang belum dikunjungi
        for j in range(n):
            if not visited[j]:
                distance = calculate_distance(locations[current_location], locations[j])
                if distance < min_distance:
                    min_distance = distance
                    next_location = j
        # Kunjungi lokasi terdekat
        visited[next_location] = True
        route.append(next_location)
        total_distance += min_distance
        current_location = next_location

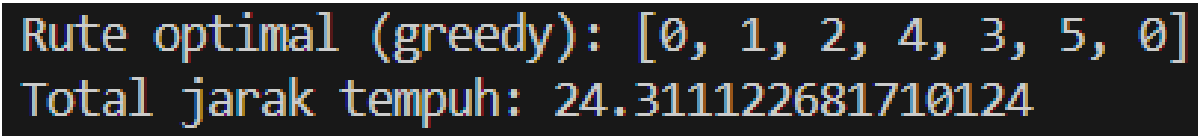
    # Kembali ke titik awal setelah semua lokasi dikunjungi
    total_distance += calculate_distance(locations[current_location], locations[0])
    route.append(0)

    return route, total_distance

# Eksekusi algoritma Greedy
route, total_distance = greedy_route(locations)
```

```
# Cetak hasil rute dan total jarak tempuh  
print("Rute optimal (greedy):", route)  
print("Total jarak tempuh:", total_distance)
```

Setelah menjalankan kode di atas, didapatkan rute optimal dan total jarak tempuh yang harus dilalui oleh kendaraan pengiriman barang. Rute yang dihasilkan algoritma Greedy adalah sebagai berikut:



```
Rute optimal (greedy): [0, 1, 2, 4, 3, 5, 0]  
Total jarak tempuh: 24.311122681710124
```

Gambar 1. Output Implementasi Python

KESIMPULAN

Penelitian ini berhasil menerapkan algoritma Greedy untuk optimisasi rute transportasi dalam pengiriman barang ke berbagai lokasi. Hasil implementasi menggunakan Python menunjukkan bahwa algoritma ini mampu menghasilkan rute yang efisien dengan total jarak tempuh sebesar 24.31 unit. Meskipun tidak selalu memberikan solusi optimal secara global, algoritma Greedy tetap menjadi metode yang menarik untuk diterapkan dalam skenario pengiriman. Penelitian ini menunjukkan bahwa meskipun terdapat beberapa keterbatasan, algoritma ini dapat meningkatkan efisiensi dalam proses pengiriman barang. Ke depannya, kombinasi dengan metode lain dapat menjadi langkah yang baik untuk mencapai hasil yang lebih optimal.

DAFTAR PUSTAKA

- Asmiati, A., Sulastriani, S., & Citta, A. B. (2023). Pengembangan Sumber Daya Manusia Dalam Mendukung Transformasi Transportasi Laut Dalam Era Revolusi Industri 4.0. *Innovative: Journal Of Social Science Research*, 3(4), 6184-6197.
- Ginting, V. S., Kusriani, K., & Taufiq, E. (2020). Implementasi Algoritma C4. 5 untuk Memprediksi Keterlambatan Pembayaran Sumbangan Pembangunan Pendidikan Sekolah Menggunakan Python. *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, 10(1), 36-44.
- Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web. *J. Tek. mesin*, 6(2), 8.
- Ranjani, J., Sheela, A., & Meena, K. P. (2019, April). Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB-A Comparative analysis. In 2019 1st international conference on innovations in information and communication technology (ICIICT) (pp. 1-5). IEEE.
- Ribangun, B. J., & Rosid, M. A. (2017). Implementasi Algoritma Greedy Pada Metode Transportasi Dengan Menggunakan Vam Dalam Pendistribusian Produk. *Spektrum Industri*, 15(1), 51.
- Rumetna, M. S., Lina, T. N., Santoso, A. B., Komansilan, R., & Karay, J. (2023). Implementasi Algoritma Depth First Search Dalam Penyelesaian Permasalahan Lintasan dan Sirkuit Euler. *Jurnal Komtika (Komputasi Dan Informatika)*, 7(1), 12-21.