

Penerapan Algoritma *Dijkstra* Dan *Greedy* Untuk Optimasi Rute Angkut Sampah Di Kecamatan Periuk

Muhammad Adam Nadjib Mahfoud¹, Taufik Hidayat^{2*}, Sukrim³, Sukisno⁴,
Asep Hardiyanto Nugroho⁵

Universitas Islam Syekh - Yusuf

Jln. Maulana Yusuf No.10 Babakan, Kota Tangerang 15118

Sur-el : adamnadjib@gmail.com¹, thidayat@unis.ac.id², sukrim@unis.ac.id³,
sukisno@unis.ac.id⁴, asep.hardiyanto@unis.ac.id⁵

^{*)} corresponding author

DOI : <https://doi.org/10.33557/jurnalatrik.v26i2.3259>

Abstract : *The increase in population in urban areas has resulted in an increase in the volume of waste generated. This creates problems in waste management, especially in determining efficient transportation routes. This study aims to optimize waste transportation routes in Periuk Subdistrict, Tangerang City, by applying Dijkstra and Greedy algorithms. Dijkstra's algorithm is used to find the shortest path based on the minimum distance traveled, while the Greedy algorithm is used to select the path that looks most profitable at each step. Geographical data in the form of coordinates of Waste Disposal Sites (TPS) and the distance between TPS are used as input in both algorithms. The results showed that Dijkstra's algorithm produced the shortest route with a distance of 16.7 km, more efficient than the route produced by the Greedy algorithm with a distance of 24.1 km*

Keywords: *Dijkstra's Algorithm, Greedy Algorithm, Route Optimization, Garbage Transport*

Abstrak : *Peningkatan jumlah penduduk di perkotaan berdampak pada peningkatan volume sampah yang dihasilkan. Hal ini menimbulkan permasalahan dalam pengelolaan sampah, terutama dalam menentukan rute pengangkutan yang efisien. Penelitian ini bertujuan untuk mengoptimalkan rute angkut sampah di Kecamatan Periuk, Kota Tangerang, dengan menerapkan algoritma Dijkstra dan Greedy. Algoritma Dijkstra digunakan untuk mencari jalur terpendek berdasarkan jarak tempuh minimum, sedangkan algoritma Greedy digunakan untuk memilih jalur yang terlihat paling menguntungkan pada setiap langkah. Data geografis berupa titik koordinat Tempat Pembuangan Sampah (TPS) dan jarak antar TPS digunakan sebagai masukan dalam kedua algoritma. Hasil penelitian menunjukkan bahwa algoritma Dijkstra menghasilkan rute terpendek dengan jarak 16,7 km, lebih efisien dibandingkan rute yang dihasilkan oleh algoritma Greedy dengan jarak 24,1 km.*

Kata kunci: *Algoritma Dijkstra, Algoritma Greedy, Optimasi Rute, Angkut Sampah*

1. PENDAHULUAN

Bahan limbah, yang dikenal sebagai sampah, adalah produk sampingan sisa yang dihasilkan dari proses manusia dan alami yang mengeras. Jumlah sampah yang bervariasi dihasilkan setiap hari oleh setiap orang. Eskalasi populasi perkotaan secara langsung berkorelasi dengan meningkatnya jumlah akumulasi limbah[1]. Hal ini disebabkan oleh peningkatan

kualitas hidup dan perubahan pola konsumsi masyarakat[2]. Oleh karena itu, pengelolaan sampah perlu ditingkatkan untuk mencegah dampak negatif yang mungkin ditimbulkan oleh sampah[3].

Kota Tangerang merupakan salah satu kota terbesar di Provinsi Banten, Indonesia. Kota ini memiliki luas wilayah sebesar 1.689,73 km² dan jumlah penduduk sebesar 2.161.073 jiwa (2022) Seiring dengan pertumbuhan penduduk yang pesat,

Kota Tangerang juga mengalami peningkatan volume sampah. Berdasarkan data Dinas Lingkungan Hidup dan Kebersihan (DLHK) Kota Tangerang, volume sampah di Kota Tangerang pada tahun 2022 mencapai 2.250 ton per hari.

Peningkatan volume sampah tersebut menimbulkan berbagai permasalahan, salah satunya adalah masalah pengelolaan sampah. Pengelolaan sampah di Kecamatan perbukit saat ini masih menggunakan metode konvensional, yaitu dengan mengumpulkan sampah dari rumah-rumah warga kemudian dibawa ke tempat pembuangan akhir (TPA). Metode ini memiliki dampak yaitu sampah yang menumpuk pada setiap Tempat Pembuangan Sampah (TPS) yang ada dan juga sampah yang berceceran di jalanan. Pendistribusian sampah yang terkoordinasi dengan baik dapat meningkatkan kualitas pengelolaan sampah. Pengelolaan sampah melibatkan beberapa faktor, seperti jalur kendaraan, jenis kendaraan, dan biaya pengelolaan. Faktor-faktor ini bertujuan untuk menjangkau wilayah pengambilan sampah yang luas dengan armada yang terbatas. Salah satu tantangan dalam pengelolaan sampah adalah pemilihan jalur pengambilan sampah. Pemilihan jalur yang tepat akan menentukan jarak tempuh armada secara keseluruhan[4].

Untuk mengatasi permasalahan pengelolaan sampah di Kota Tangerang, diperlukan suatu metode pengelolaan sampah yang lebih efisien, efektif, dan ramah lingkungan. Salah satu metode yang dapat digunakan adalah metode optimasi rute angkut sampah. Optimasi rute membantu menentukan jalur pengumpulan sampah terpendek atau tercepat, sehingga mengurangi jarak tempuh kendaraan pengangkut sampah. Hal ini berdampak pada penghematan bahan bakar, penurunan emisi gas

rumah kaca, dan pengurangan biaya operasional. Optimasi rute angkut sampah merupakan suatu metode yang bertujuan untuk menentukan rute angkut sampah yang paling efisien, efektif, dan ramah lingkungan. Metode ini dapat dilakukan dengan menggunakan algoritma-algoritma optimasi, seperti algoritma *dijkstra* dan algoritma *greedy*[5].

Algoritma *dijkstra* adalah metode yang digunakan untuk mencari jarak terpendek pada graf berarah[6]. Algoritma ini bekerja dengan membangun pohon jalur terpendek dari titik awal ke titik akhir[7]. Algoritma *greedy* adalah algoritma heuristik yang bekerja dengan langkah-langkah yang disusun secara sistematis. Algoritma ini membuat solusi langkah demi langkah, dengan memilih opsi yang paling menguntungkan pada setiap langkah. Pada langkah-langkah tertentu dalam algoritma *greedy*, keputusan yang telah diambil tidak dapat diubah lagi. Algoritma *greedy* adalah metode yang memecahkan masalah dengan mengambil keputusan terbaik pada setiap langkah, dan merupakan salah satu metode yang digunakan dalam masalah optimasi[8].

Berdasarkan penelitian yang telah dilakukan beberapa peneliti tentang optimasi rute untuk penentuan rute terpendek seperti penelitian yang dilakukan oleh Luluk Suryani dan Ery Murniyasih tentang “Pencarian Rute Terpendek Pada Aplikasi Ojek Sampah Dengan Menggunakan Algoritma Dijkstra” [9]. Berdasarkan penelitian tersebut, Penggunaan algoritma *dijkstra* ketika awal pemrosesan pada penentuan jarak terdekat antar titik konsumen dapat bekerja dengan baik. Selanjutnya dalam penelitian yang dilakukan oleh M. Rhifky

Wayahdi, dkk. Tentang “*Greedy, A-Star, and Dijkstra’s Algorithms in Finding Shortest Path*” [10] Telah ditentukan bahwa pemilihan algoritma yang berkaitan dengan masalah jalur terpendek bergantung pada persyaratan khusus dari aplikasi yang bersangkutan. Algoritma *Dijkstra* mewakili pilihan optimal ketika optimasi adalah yang paling penting; Namun, itu membutuhkan jumlah sumber daya komputasi yang lebih besar. Algoritma *Greedy* memberikan kecepatan yang unggul; Namun, itu tidak memastikan solusi optimal, sehingga membuatnya sesuai untuk skenario yang memprioritaskan kecepatan daripada akurasi absolut. Algoritma *A-Star* muncul sebagai opsi yang paling menguntungkan mengenai keseimbangan antara kecepatan dan presisi, memosisikannya sebagai solusi yang sangat efektif untuk beragam aplikasi praktis yang menuntut efisiensi dan hasil optimal. Penerapan algoritma *dijkstra* dan algoritma *greedy* untuk optimasi rute angkut sampah di Kecamatan Periuk belum pernah dilakukan. Oleh karena itu, penelitian ini bertujuan untuk menerapkan algoritma *dijkstra* dan algoritma *greedy* untuk optimasi rute angkut sampah di Kecamatan Periuk.

2. METODOLOGI PENELITIAN

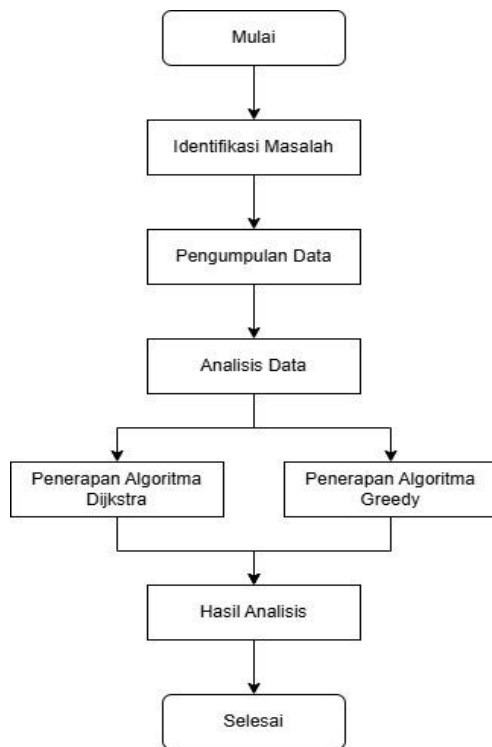
Penelitian ini menggunakan metode penelitian kuantitatif. Metode penelitian kuantitatif adalah metode penelitian yang menggunakan data numerik untuk menganalisis fenomena yang diteliti. Penggunaan metode ini bertujuan untuk mengoptimalkan rute angkutan

sampah di Kecamatan Periuk. Untuk mengoptimalkan rute angkutan sampah, perlu mengumpulkan data kuantitatif, seperti data geografis.

Pada metode studi literatur ini diharapkan dapat membantu Penulis dalam proses penelitian. Studi literatur dapat membantu Penulis dalam mengembangkan hipotesis serta mendapat referensi yang relevan. Proses ini dilakukan dengan membaca dan mereview dari berbagai sumber, seperti buku, jurnal dan artikel dengan kurun waktu 5 tahun kebelakang dan menyesuaikan dengan judul yang teliti.

Proses wawancara merupakan suatu proses memperoleh informasi untuk keperluan penelitian melalui tanya jawab secara langsung kepada responden. Proses wawancara ini di gunakan untuk mendapatkan informasi lebih lanjut mengenai proses pada sistem angkut sampah di Kecamatan Periuk. Wawancara dilakukan di Dinas Lingkungan Hidup Kota Tangerang dan dilakukan dengan bapak Meidi Nugroho Adi, S.Si. selaku analis pengembangan infrastruktur bidang kebersihan dan pengelolaan sampah, pada hasil wawancara awal penulis mendapatkan beberapa informasi mengenai jumlah armada dan juga beberapa jumlah TPS di Kecamatan tersebut.

Dalam Penelitian ini menggunakan data berupa titik geografis yang merupakan jarak antara lokasi tiap TPS. Data yang dimasukkan berupa data dalam satuan jarak kilometer (km) dapat diperoleh dari Dinas Lingkungan Hidup Kota Tangerang. Kemudian data tersebut dilakukan proses tahap analisis data menggunakan algoritma *dijkstra* dan *greedy*.



Gambar 1. Diagram Alur Penelitian

Pada tahapan diatas, ada beberapa hal yang dilakukan, diantaranya adalah :

1. Identifikasi Masalah

hal yang pertama dilakukan identifikasi masalah yang ada di Kecamatan Periuk. Dengan mengidentifikasi masalah diharapkan dapat memahasi seara jelas permasalahan pengangkutan sampah yang terjadi.

2. Pengumpulan Data

Lalu pada tahap selanjutnya dilakukan pengumpulan data, data yang dikumpulkan berupa data geografis. Data yang digunakan berupa satuan jarak dalam kilometer (km), serta data geografis berupa titik koordinat setiap Tempat Pembuangan Sampah (TPS).

3. Analisis Data

Setelah proses pengumpulan data selesai dilakukan proses analisis data. Analisis data bertujuan untuk menentukan parameter yang akan digunakan dalam algoritma *dijkstra* dan *greedy*.

4. Penerapan Algoritma *Dijkstra* dan *Greedy*

Pada tahapan ini dilakukan untuk membuat rute terpendek sistem angkutan sampah dengan menggunakan algoritma *dijkstra* dan *greedy*. Rute yang dibuat harus dapat menghasilkan rute terpendek.

5. Hasil Analisis

Selanjutnya adalah melihat hasil analisis apakah kinerja model tersebut dapat mengoptimalkan rute angkutan sampah. Hasil analisis berupa perbandingan rute terpendek dari titik awal ke titik akhir dari pengangkutan sampah.

2.1 Optimasi

Optimasi adalah prosedur di mana hasil optimal, juga dikenal sebagai nilai yang paling efisien, dicapai [11]. Dalam bidang matematika, optimasi terutama berkaitan dengan dilema yang berputar di sekitar pencapaian nilai tertinggi atau terendah dari fungsi tertentu. Untuk mengamankan nilai optimal, baik itu minimum atau maksimum, proses metodis pemilihan nilai variabel diimplementasikan. Variabel-variabel ini berpotensi menjadi bilangan bulat atau bilangan real.

2.2 Algoritma *Dijkstra*

Algoritma *dijkstra* merupakan algoritma yang dapat digunakan untuk mencari jalur terpendek dari suatu titik ke titik lain pada graf berarah [12] [13]. Berbagai algoritma telah diusulkan sejak algoritma *dijkstra* pertama kali dideskripsikan, termasuk A*, D*, D* Lite, dan ACO (*ant colony optimization*)[14]. algoritma *dijkstra* ini juga merupakan algoritma yang umum digunakan untuk menyelesaikan masalah optimasi, yaitu masalah yang bertujuan untuk

menemukan solusi terbaik dari berbagai alternatif yang tersedia. algoritma *Dijkstra* ditemukan oleh Edsger W. Dijkstra melalui tulisannya dengan judul “ *A Note on Two Problems in Connexion with Graphs* “ pada tahun 1959[15].

Saat melakukan pencarian rute optimal menggunakan algoritma Dijkstra, penting untuk memiliki bobot yang ditetapkan untuk setiap node. Prinsip inti dari algoritma Dijkstra difokuskan pada penentuan jalur terpendek ke node individu pada setiap langkah proses. Saat algoritma maju ke langkah ke-n, jumlah kumulatif dari n node diperkaya dengan wawasan ke jalur yang paling efisien. Selanjutnya, algoritma secara metodis menyempurnakan jalur paling langsung ke setiap simpul sambil bergerak dari satu simpul ke simpul lainnya untuk memastikan rute terpendek dari titik awal ke titik akhir yang ditentukan. Proses pencarian rute terdekat menggunakan Dijkstra dijelaskan dengan persamaan 1.

$$D(V_i) = \min(D(v_j), D(v_i) + W(v_i, v_j)) \quad (1)$$

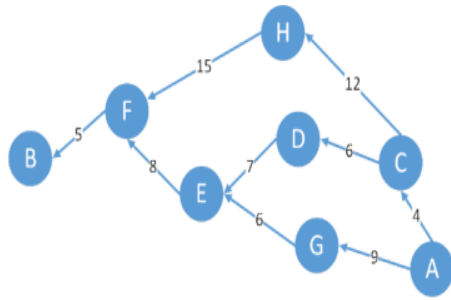
Rumus menggambarkan metode untuk menghitung bobot yang digunakan dalam menentukan node berikutnya yang akan dikunjungi. Di sini, $D(v_i)$ mewakili jumlah minimal bobot dari v_i ke v_j , dan $W(v_i, v_j)$ menunjukkan berat simpul dari v_i ke v_j , yang diturunkan dengan menentukan setidaknya berat minimum dari $D(v_i)$ ke $D(v_j)$. Rumus ini digunakan untuk setiap simpul untuk mengidentifikasi simpul dengan bobot terkecil. Selanjutnya, pencarian jalan terpendek dimulai :

1. Mendefinisikan $L = \{ \}$ dan $V = \{V_1, V_2, V_3, V_4, \dots, V_n\}$
2. Dalam iterasi berikutnya lakukan $D(v_i) = W(v_i, v_j)$
3. Apabila V_n belum menjadi titik tetap ($V_n \notin L$) maka :
 - a. Pilih titik $v_k \in V - L$ dengan $D(v_k)$ terkecil, lalu $L = L \cup \{v_k\}$ maka jadikan v_k sebagai titik permanen.
 - b. Untuk setiap $v_j \in V - L$ terapkan, jika $D(v_k) + W(v_k, v_j) < D(v_j)$ ubah $D(v_j)$ dengan $D(v_k) + W(v_k, v_j)$.

2.3 Algoritma Greedy

Algoritma *greedy* adalah algoritma yang digunakan untuk memecahkan persoalan optimasi. Persoalan optimasi adalah persoalan yang mencari solusi terbaik dari sekumpulan solusi yang mungkin. algoritma *greedy* bekerja dengan memilih jalur yang paling menguntungkan pada setiap langkah. Dengan memilih jalur yang paling menguntungkan pada setiap langkah, diharapkan algoritma *greedy* akan menghasilkan solusi terbaik secara keseluruhan.

Algoritma *greedy* bekerja secara bertahap. Pada setiap tahap, algoritma ini perlu memilih salah satu dari beberapa pilihan yang tersedia. Pemilihan pilihan yang terbaik pada setiap tahap akan menentukan solusi akhir yang dihasilkan. Keputusan yang telah diambil pada suatu tahap tidak dapat diubah lagi pada tahap selanjutnya[16]. Proses pencarian rute terdekat menggunakan *greedy* dijelaskan sebagai berikut.



Gambar 2. Grafik Arah dari titik A ke B

Untuk menentukan rute yang paling efisien dari titik A ke titik B, algoritma *greedy* akan menjalankan serangkaian langkah yang mirip dengan yang diuraikan di atas :

1. Kunjungi satu titik pada grafik, dan ambil seluruh titik yang dapat dikunjungi dari titik sekarang.
2. Temukan maksimum lokal ke titik berikutnya.
3. Tandai grafik sekarang sebagai grafik yang telah Anda kunjungi, dan pindahkan ke maksimum lokal yang telah ditentukan sebelumnya.
4. Kembali ke langkah 1 sampai titik tujuan diperoleh.

3. HASIL DAN PEMBAHASAN

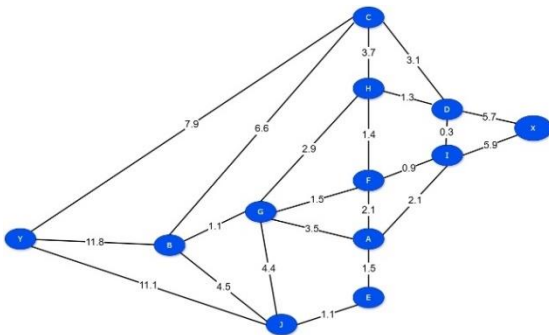
Pada tahap ini analisa data titik Lokasi TPS yang sudah didapatkan dari Dinas Lingkungan Hidup Kota Tangerang. Dalam proses pencarian rute terdekat, hanya mencari 1 rute untuk digunakan sebagai penentuan rute terpendek dari titik awal pengangkutan lalu menuju TPS yang ada di Kecamatan Periuk hingga mencapai ke titik akhir yaitu TPA dengan menerapkan dua algoritma yaitu Algoritma Dijkstra dan Algoritma Greedy. Data – data yang akan digunakan dalam penelitian ini yaitu pada tabel 1.

Tabel 1. Simpul TPS

Simpul	TPS	Kelurahan	Titik Lokasi TPS	
			x	y
A	Jalan Prabu Kian Santang	Gebang Raya	-6.18	106.594
B	JL.Protokol Wisma Harapan R/L s/d Kabupaten	Gembor	-6.176	106.577
C	Jalan Raya Mauk -TPS Elang	Periuk	-6.148	106.592
D	JL.M Toha, Sisir Jalur Protokol transit gerobak dan bentor didepan bulog	Periuk Jaya	-6.166	106.6
E	Polsek Jatiuwung-Jln. Gatsu - Pasar Laris	Sangiang Jaya	-6.191	106.59
F	Tps Gor Priuk dan Perumahan Gebang Raya Rw 04	Gebang Raya	-6.171	106.594
G	Rw 004 Purati, Protokol Wisma Harapan,Bentor Rw 11, Bentor dan Gerobak Rw 005, Cluster Alamanda	Gembor	-6.173	106.583
H	TPS Situ Bulakan	Periuk	-6.162	106.593
I	JL.M Toha, Sisir Jalur Protokol transit gerobak dan bentor didepan bulog	Periuk Jaya	-6.168	106,6
J	Pasar Laris - Tps Rw 04	Sangiang Jaya	-6,188	106.596

Simpul	TPS	Kelurahan-an	Titik Lokasi TPS	
			x	y
X	dan 05	Neglasari	-	106.629
	Kantor DLH Kota		6.1565	670
	Tangerang		95	
Y	TPA Rawa Kucing	Neglasari	-	106.617
			6.1392	148
			90	

Selanjutnya melibatkan penggambaran garis untuk berfungsi sebagai lintasan, yang selanjutnya dapat dimasukkan ke dalam Grafik, diikuti dengan menilai panjang lintasan yang sudah ada sebelumnya. Dalam proses pengangkutan pada Kecamatan Periuk, DLH sebagai simpul awal (simpul X) menuju ke lokasi pembuangan akhir atau sebagai simpul tujuan (simpul Y) dengan total 22 simpul yang disajikan pada gambar 2.



Gambar 3. Graf Berbobot

Selanjutnya dalam proses pengangkutan pada Kecamatan Periuk, DLH sebagai simpul awal (simpul X) menuju ke lokasi pembuangan akhir atau sebagai simpul tujuan (simpul Y) dengan total 22 simpul yang ditampilkan pada Tabel 2 dibawah.

Tabel 2. Simpul dan jarak antar simpul

No	Simpul	Bobot		No	Simpul	Bobot	
		Jarak (km)				Jarak (km)	
1	X - D	5.7		12	H - G	2.9	
2	X - I	5.9		13	F - G	1.5	
3	D - I	0.3		14	F - A	2.1	
4	D - H	1.3		15	A - E	1.5	
5	D - C	3.1		16	A - G	3.5	

No	Simpul	Bobot		No	Simpul	Bobot	
		Jarak (km)				Jarak (km)	
6	I - F	0.9		17	E - J	1.1	
7	I - A	2.1		18	G - B	1.1	
8	C - Y	7.9		19	G - J	4.4	
9	C - H	3.7		20	J - Y	11.1	
10	C - B	6.6		21	B - J	4.5	
11	H - F	1.4		22	B - Y	11.8	

3.1 Penerapan Algoritma Dijkstra

Kemudian, algoritma Dijkstra digunakan untuk menentukan rute terpendek dalam sebuah graf. Untuk menghitung rute terpendek dari titik X ke titik Y menggunakan algoritma Dijkstra, kita akan menerapkan persamaan 2.

$$d[v] = \min\{d[u] + w(u, v)\} \quad (2)$$

Keterangan :

- d[v] : Jarak terpendek dari simpul awal (start) ke simpul
- u : Simpul yang sudah diketahui jarak terpendeknya dan merupakan tetangga dari v.
- w(u, v) : Bobot (jarak) dari simpul u ke simpul v.

Untuk dapat menemukan rute terpendek menuju titik Y yaitu TPA Rawa Kucing sebagai lokasi tujuan, maka perhitungan menggunakan Algoritma Dijkstra, langkah-langkah nya dapat dilihat sebagai berikut :

- a. Langkah 1 Inisialisasi

$$d[X] = 0 \text{ (jarak ke dirinya sendiri adalah 0)}$$

$$d[A] = d[B] = d[C] = d[D] = d[E] = d[F] = d[G] = d[H] = d[I] = d[J] = d[Y] = \infty$$

- b. Langkah 2 Iterasi

1. Iterasi 1 :

Mulai dari simpul X:

Tetangga dari X = D (5.7 km), I (5.9 km)

Perbarui jarak:

$$d[D] = \min (\infty, 0 + 5.7) = 5.7 \text{ km}$$

$$d[I] = \min (\infty, 0 + 5.9) = 5.9 \text{ km}$$

2. Iterasi 2 :

Pilih simpul dengan jarak minimum yang belum dikunjungi : D (5.7 km)

Tetangga dari D = X(5.7 km), I(0.3 km), H(1.3 km), C(3.1 km)

Perbarui jarak :

$$d[I] = \min (5.9, 5.7 + 0.3) = 6.0 \text{ km (tidak diperbarui)}$$

$$d[H] = \min (\infty, 5.7 + 1.3) = 7.0 \text{ km}$$

$$d[C] = \min (\infty, 5.7 + 3.1) = 8.8 \text{ km}$$

3. Iterasi 3 :

Pilih simpul dengan jarak minimum yang belum dikunjungi: I (5.9 km)

Tetangga dari I = F (0.9 km), A (2.1 km), X (5.9 km), D (0.3 km)

Perbarui jarak :

$$d[F] = \min (\infty, 5.9 + 0.9) = 6.8 \text{ km}$$

$$d[A] = \min (\infty, 5.9 + 2.1) = 8.0 \text{ km}$$

4. Iterasi 4 :

Pilih simpul dengan jarak minimum yang belum dikunjungi: F (6.8 km)

Tetangga dari F = G (1.5 km), A (2.1 km), I (0.9 km), H (1.4 km)

Perbarui jarak:

$$d[G] = \min (\infty, 6.8 + 1.5) = 8.3 \text{ km}$$

$$d[H] = \min (7.0, 6.8 + 1.4) = 8.2 \text{ km}$$

5. Iterasi 5 :

Pilih simpul dengan jarak minimum yang belum dikunjungi: D (7.0 km)

Tetangga dari G = B(1.1 km), J (4.4 km), A(3.5 km), F(1.5 km),

H(2.9 km)

Perbarui jarak:

$$d[B] = \min (\infty, 8.3 + 1.1) = 9.4 \text{ km}$$

6. Iterasi 6 :

Pilih simpul dengan jarak minimum yang belum dikunjungi: C (8.8 km)

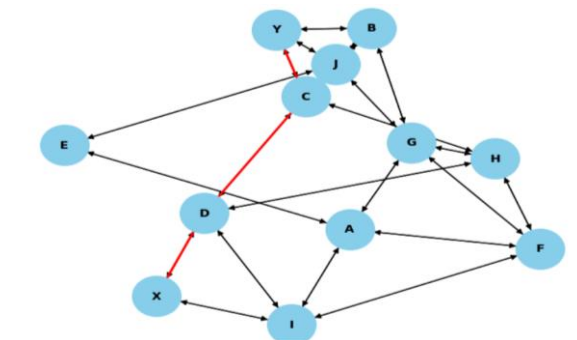
Tetangga dari C = B (6.6 km), H (3.7 km), Y (7.9 km)

Perbarui jarak

$$d[Y] = \min (\infty, 8.8 + 7.9) = 16.7 \text{ km}$$

c. Hasil

Dari perhitungan tersebut, rute terpendek dari X ke Y ditemukan dengan jarak 16.7 km atau bisa dilihat pada gambar dibawah.



Gambar 4. Graf berarah rute terpendek dengan Dijkstra

Maka dari itu hasil akhir dari Algoritma Dijkstra telah menemukan bahwa jalur terpendek dari simpul awal X ke simpul akhir Y adalah X -> D -> C -> Y dengan total jarak 16.7 km. Jalur ini dipilih karena merupakan kombinasi jalur-jalur yang memiliki bobot (jarak) terkecil di antara semua kemungkinan jalur dari simpul X ke simpul Y.

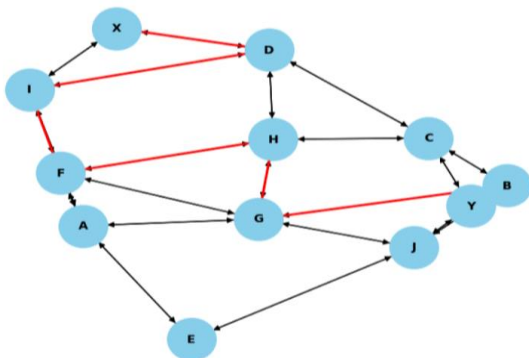
3.2 Penerapan Algoritma Greedy

Algoritma greedy yang digunakan tidak menggunakan rumus matematis yang eksplisit seperti Dijkstra. Prinsipnya adalah memilih

tetangga terdekat yang belum dikunjungi pada setiap langkah. Meskipun tidak ada rumus matematis yang spesifik, ada logika "pemilihan terbaik lokal" yang menjadi inti dari pendekatan *greedy*:

$$\text{nearest_neighbor} = \min(\text{unvisited_neighbors}, \text{key}=\text{unvisited_neighbors.get})$$

Di sini, simpul tetangga terdekat (*nearest_neighbor*) dipilih dari *unvisited_neighbors* berdasarkan bobot sisi terkecil. Ini adalah contoh dari "greedy choice" karena *algoritma* hanya mempertimbangkan keuntungan langsung (jarak terdekat) pada setiap langkah, tanpa memperhitungkan konsekuensi jangka panjang.



Gambar 5. Graf berarah rute terpendek dengan Greedy

Hasil tersebut menunjukkan bahwa algoritma *greedy* berhasil sampai menuju titik akhir pada rute dalam graf yang diberikan. Pada graf yang digunakan, algoritma memulai dari simpul X dan secara iteratif memilih tetangga terdekat yang belum dikunjungi hingga mencapai simpul Y. Total jarak tempuh yang dihitung adalah 24.1 km, yang sesuai dengan panjang rute terpendek dalam graf tersebut.

Dalam menganalisis langkah demi langkah algoritma *greedy* dimulai dari titik awal yaitu :

1. simpul X, algoritma *greedy* akan memilih tetangga terdekat yang belum dikunjungi, yaitu simpul D (5.7 km).
2. Dari simpul D, tetangga terdekat yang belum dikunjungi adalah simpul I (0.3 km).
3. Dari simpul I, tetangga terdekat yang belum dikunjungi adalah simpul F (0.9 km).
4. Dari simpul F, tetangga terdekat yang belum dikunjungi adalah simpul H (1.4 km).
5. Dari simpul H, tetangga terdekat yang belum dikunjungi adalah simpul G (2.9 km).
6. Dari simpul G, tetangga terdekat yang belum dikunjungi adalah simpul B (1.1 km).
7. Terakhir dari simpul B, algoritma *greedy* akan langsung menuju titik tujuan yaitu simpul Y (11.8 km) karena merupakan tetangga yang belum dikunjungi.

Maka dari itu hasil akhir dari Algoritma Greedy telah menemukan bahwa jalur terpendek dari simpul awal X ke simpul akhir Y adalah X -> D -> I -> F -> H -> G -> B -> Y dengan total jarak 24.1 km.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan pada bab sebelumnya, dapat disimpulkan bahwa penerapan algoritma Dijkstra dan Greedy untuk optimasi rute angkut sampah di Kecamatan Periuk menghasilkan rute terpendek yang berbeda. Algoritma Dijkstra menghasilkan rute terpendek dengan jarak 16,7 km melalui simpul X -> D -> C -> Y, sedangkan algoritma Greedy

menghasilkan rute dengan jarak 24,1 km melalui simpul X -> D -> I -> F -> H -> G -> B -> Y. Perbedaan ini terjadi karena algoritma Dijkstra mempertimbangkan jarak kumulatif terpendek dari simpul awal ke setiap simpul yang dikunjungi, sedangkan algoritma Greedy hanya memilih simpul terdekat pada setiap langkah tanpa mempertimbangkan optimasi global.

DAFTAR PUSTAKA

- [1] I. nurani Dewi, I. Royani, S. Sumarjan, and H. Jannah, "Pemberdayaan Masyarakat Melalui Pengelolaan Sampah Skala Rumah Tangga Menggunakan Metode Komposting," *Sasambo: Jurnal Abdimas (Journal of Community Service)*, vol. 2, no. 1, pp. 12–18, Mar. 2020, doi: <https://doi.org/10.36312/sasambo.v2i1.172>
- [2] S. Manyullei, L. M. Saleh, N. I. Arsyi, A. P. Azzima, and N. Fadhilah, "Penyuluhan Pengelolaan Sampah dan PHBS di Sekolah Dasar 82 Barangmamas Kecamatan Galesong Selatan Kab. Takalar," *Jurnal Altifani Penelitian dan Pengabdian kepada Masyarakat*, vol. 2, no. 2, pp. 169–175, Mar. 2022, doi: <https://doi.org/10.25008/altifani.v2i2.210>
- [3] W. Czekala, J. Drozdowski, and P. Łabiak, "Modern Technologies for Waste Management: A Review," Aug. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: <https://doi.org/10.3390/app13158847>
- [4] B. Turan, V. Hemmelmayr, A. Larsen, and J. Puchinger, "Transition towards sustainable mobility: the role of transport optimization," *Cent Eur J Oper Res*, vol. 32, no. 2, pp. 435–456, Jun. 2024, doi: <https://doi.org/10.1007/s10100-023-00888-8>
- [5] R. R. Al Hakim *et al.*, "Aplikasi Algoritma Dijkstra dalam Penyelesaian Berbagai Masalah," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 11, no. 1, p. 42, Jun. 2021, doi: <http://dx.doi.org/10.36448/expert.v11i1.1939>
- [6] A. Cantona, F. Fauziah, and W. Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *Jurnal Teknologi dan Manajemen Informatika*, vol. 6, no. 1, Apr. 2020, doi: <https://doi.org/10.26905/jtmi.v6i1.3837>
- [7] S. Yosua, C. Sigalingging, J. Jipesya, and Y. Jumaryadi, "Implementasi Algoritma Dijkstra dalam Pencarian Klinik Hewan Terdekat," *Jurnal Ilmiah FIFO*, vol. 13, no. 1, p. 85, Dec. 2021, doi: <https://dx.doi.org/10.22441/fifo.2021.v13i1.009>
- [8] A. C. Wibowo and A. D. Wowor, "Perancangan Distribusi Hasil Produk Textil dengan Rute Terdekat dengan Algoritma Greedy," *J-SAKTI (Jurnal Sains Komputer & Informatika)*, vol. 7, no. 1, 2023. Doi : <http://dx.doi.org/10.30645/j-sakti.v7i1.592>
- [9] L. Suryani and E. Murniyasih, "Pencarian Rute Terpendek pada Aplikasi Ojek Sampah dengan Menggunakan Algoritma Dijkstra," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 2, p. 385, Dec. 2022, doi: <https://doi.org/10.37600/tekinkom.v5i2.586>
- [10] M. R. Wayahdi, S. H. N. Ginting, and D. Syahputra, "Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path," *International Journal of Advances in Data and Information Systems*, vol. 2, no. 1, pp. 45–52, Feb. 2021, doi: <https://doi.org/10.25008/ijadis.v2i1.1206>
- [11] R. Rizky, T. Hidayat, A. H. Nugroho, and Z. Hakim, "Implementasi Metode A*Star Pada Pencarian Rute Terdekat Menuju Tempat Kuliner di Menes Pandeglang Banten," *Geodika: Jurnal Kajian Ilmu dan Pendidikan Geografi*, vol. 4, no. 1, pp. 85–94, Jun. 2020, doi: <https://doi.org/10.29408/geodika.v4i1.2068>
- [12] H. Pratiwi, "Application Of The Dijkstra Algorithm To Determine The Shortest Route From City Center Surabaya To Historical Places," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 4, no. 1, pp. 213–223, Jan. 2022, doi: [10.47233/jteksis.v4i1.407](https://doi.org/10.47233/jteksis.v4i1.407)

- [13] J. Setiawan, R. S. Prakoso, and K. Suryaningrum., “Penentuan Rute Terpendek menuju pusat Perbelanjaan di Jakarta menggunakan Algoritma Dijkstra,” *jurnalmatrik*, vol. 21, no. 3, pp. 156–165, Dec. 2019., doi : <https://doi.org/10.33557/jurnalmatrik.v21i3.715>
- [14] H. Aburas, I. Shahrour, and C. Giglio, “Route Planning under Mobility Restrictions in the Palestinian Territories,” *Sustainability*, vol. 16, no. 2, p. 660, Jan. 2024, doi: <https://doi.org/10.3390/su16020660>
- [15] Y. Nora, M., D. Jollyta., and F. Saputra, “Analisis Sistem Jalur Terpendek Menggunakan Algoritma Dijkstra dan Evaluasi Usability,” *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 6, no. 1, 2020. doi: <https://dx.doi.org/10.26418/jp.v6i1.37627>
- [16] A. Januanto, A. Faqih Septiyanto, A. Kartika, and W. Hapantenda, “Penentuan Rute Optimal pada Distribusi Barang Menggunakan Algoritma Greedy (Studi Kasus: UD XYZ),” *KONVERGENSI*, Vol 17 No 1 (2021). doi: <https://doi.org/10.30996/konv.v17i1.5295>