



PERBANDINGAN KINERJA DAN KOMPLEKSITAS ALGORITMA FLOYD-WARSHALL DAN GREEDY PADA RUTE EVAKUASI BANJIR

Ajril Abyad Alhaq¹, Nia Nurfitria²

^{1,2} Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas PGRI Ronggolawe

email korespondensi: abyadalhaq02@gmail.com

Diterima: 27-02-2026, Revisi: 17-03-2026, Diterbitkan: 15-06-2026

ABSTRAK

Kecamatan Widang di Kabupaten Tuban rawan banjir akibat luapan Sungai Bengawan Solo, sehingga pengembangan rute evakuasi yang efisien menjadi sangat penting. Penelitian ini membandingkan kinerja dan kompleksitas waktu Algoritma Floyd-Warshall dan Greedy dalam menentukan rute evakuasi terpendek dari dua belas titik rawan banjir menuju satu pos evakuasi. Jaringan jalan dimodelkan sebagai graf berbobot yang merepresentasikan titik banjir, jalur alternatif, dan pos evakuasi. Floyd-Warshall menggunakan pemrograman dinamis dengan kompleksitas $O(n^3)$ untuk menghitung lintasan terpendek secara global, sedangkan Greedy memilih rute berdasarkan keputusan lokal terbaik dengan kompleksitas $O(V + E)$. Hasil penelitian menunjukkan bahwa kedua algoritma menghasilkan rute dan jarak yang sama pada sembilan titik. Pada tiga titik lainnya, Floyd-Warshall memberikan solusi lebih optimal dengan selisih jarak maksimum 0.2 km. Analisis kompleksitas menunjukkan bahwa Greedy lebih efisien secara komputasional pada graf berskala besar, sedangkan Floyd-Warshall menjamin optimalitas global meskipun waktu prosesnya meningkat seiring bertambahnya jumlah simpul. Dengan demikian, Greedy lebih sesuai untuk kondisi darurat, sedangkan Floyd-Warshall lebih tepat untuk perencanaan pra-bencana.

Kata kunci: Evakuasi banjir, Floyd-Warshall, Greedy, Lintasan terpendek

ABSTRACT

Widang Subdistrict in Tuban Regency is prone to flooding due to the overflow of the Bengawan Solo River, making the development of efficient evacuation routes critically important. This study compares the performance and time complexity of the Floyd-Warshall and Greedy algorithms in determining the shortest evacuation routes from twelve flood-prone locations to a single evacuation post. The road network is modeled as a weighted graph representing flood points, alternative routes, and the evacuation post. Floyd-Warshall employs dynamic programming with a time complexity of $O(n^3)$ to compute global shortest paths, while Greedy selects routes based on locally optimal decisions with a complexity of $O(V + E)$. The results show that both algorithms produce identical routes and distances for nine locations. For the other three locations, Floyd-

Warshall provides a more optimal solution with a maximum distance difference of 0.2 km. Complexity analysis reveals that Greedy is computationally more efficient for large-scale graphs, whereas Floyd–Warshall guarantees global optimality even though its processing time increases as the number of vertices grows. Therefore, Greedy is more suitable for emergency response scenarios, while Floyd–Warshall is better suited for pre-disaster evacuation planning.

Key words: Flood evacuation, Floyd–Warshall, Greedy algorithm, Shortest path

Pendahuluan

Banjir merupakan salah satu bencana alam yang kerap melanda berbagai wilayah di Indonesia, khususnya daerah yang berada di sepanjang aliran sungai besar. Meluapnya Sungai Bengawan Solo menjadikan Kecamatan Widang, Kabupaten Tuban sebagai daerah rawan banjir, terutama saat musim hujan tiba (Nurdiansyah dkk., 2025). Apabila tidak ditangani dengan baik, kondisi ini dapat mengancam keselamatan warga dan melumpuhkan aktivitas sehari-hari. Oleh karena itu, ketersediaan rute evakuasi yang cepat dan aman menuju titik evakuasi resmi menjadi hal yang tidak bisa diabaikan dalam upaya mitigasi bencana (Mahardika & Setianingsih, 2018).

Dalam hal ini, Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Tuban memegang peranan penting sebagai ujung tombak koordinasi dan pelaksanaan evakuasi di wilayah terdampak (Denisya & Hertati, 2024). Salah satu bentuk dukungan yang dapat diberikan adalah menyediakan informasi rute evakuasi terbaik agar proses penyelamatan berjalan lebih efisien dan risiko selama evakuasi dapat ditekan semaksimal mungkin (Budhiana dkk., 2023). Pendekatan matematis berbasis analisis diharapkan mampu membantu BPBD dalam mengambil keputusan yang lebih akurat dan terstruktur (Afandi & Mayasari, 2021).

Permasalahan penentuan rute evakuasi pada dasarnya dapat dipandang sebagai masalah optimasi yang direpresentasikan dalam bentuk graf berbobot, di mana setiap simpul mewakili suatu lokasi dan setiap sisi mewakili jarak antar lokasi tersebut (Mishra dkk., 2018). Dengan mengintegrasikan QGIS dan algoritma pencarian rute, analisis dapat dilakukan secara lebih komprehensif dengan mempertimbangkan kondisi geografis dan karakteristik wilayah terdampak banjir (Tani & Yamamoto, 2020). Pemilihan pendekatan yang tepat dalam menetapkan rute evakuasi menjadi salah satu faktor kunci keberhasilan upaya mitigasi bencana (Wayahdi dkk., 2021).

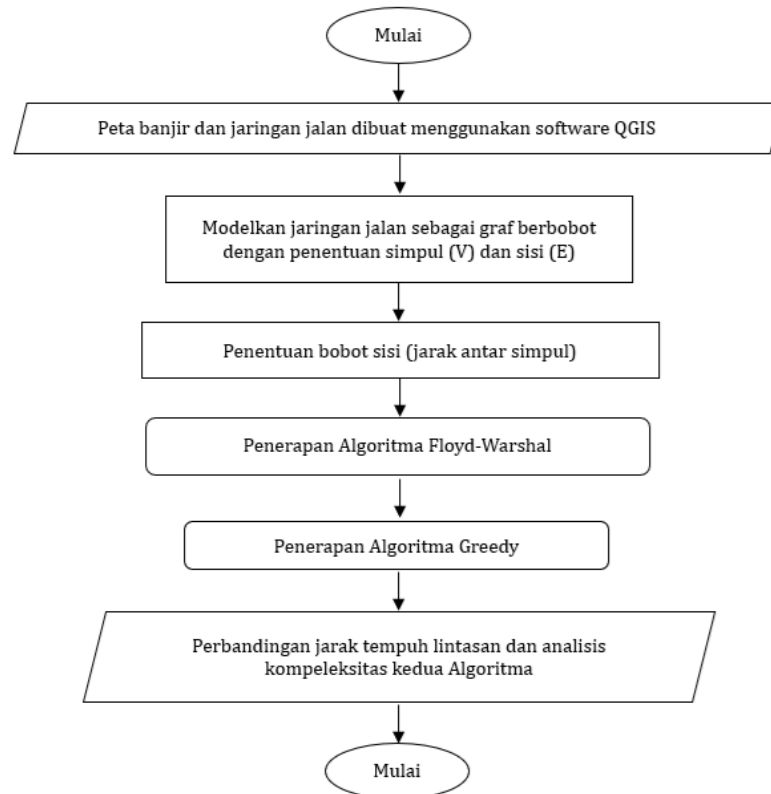
Dua algoritma yang umum digunakan dalam pencarian lintasan terpendek adalah Algoritma Floyd–Warshall dan Algoritma Greedy (Azis dkk., 2018). Algoritma Greedy dikenal praktis dan cepat karena memilih jalur lokal terpendek di setiap langkahnya, meskipun pendekatan ini tidak selalu menghasilkan solusi yang optimal secara global (Kekal dkk., 2021). Sementara itu, Algoritma Floyd–Warshall bekerja secara iteratif melalui simpul perantara untuk menemukan jarak terpendek antar seluruh pasangan simpul dalam graf berbobot (Ramadhan dkk., 2018), dan telah terbukti banyak digunakan dalam berbagai sistem navigasi maupun jaringan transportasi (Darmadi dkk., 2023).

Akan tetapi, penelitian-penelitian tersebut umumnya masih menguji kedua algoritma pada peta umum, jaringan transportasi biasa, atau graf acak, sehingga belum menggambarkan secara spesifik tantangan yang muncul dalam konteks evakuasi bencana banjir. Khususnya di Kecamatan Widang, kajian yang secara khusus membandingkan kinerja kedua algoritma untuk rute evakuasi berbasis data riil dari BPBD masih belum ditemukan (Harahap & Triase, 2024). Padahal, karakteristik lokasi titik evakuasi, kondisi geografis yang terdampak banjir, serta kebutuhan akan kecepatan komputasi dalam situasi darurat menjadikan studi ini sangat berbeda dengan penelitian-penelitian sebelumnya. Oleh karena itu, penelitian ini bertujuan tidak hanya untuk membandingkan optimalitas lintasan yang dihasilkan oleh kedua algoritma, tetapi juga untuk menganalisis kompleksitas waktunya sebagai pertimbangan penting dalam implementasi praktis di lapangan.

Metode Penelitian

Penelitian ini merupakan kajian teoritis dengan pendekatan kuantitatif deskriptif untuk menganalisis lintasan terpendek pada graf berbobot $G = (V, E, w)$. Dalam penelitian ini, V (simpul) mewakili titik-titik rawan banjir, persimpangan jalan, dan lokasi pos evakuasi; E (sisi) mewakili jalur jalan yang menghubungkan antar simpul; dan w (bobot) mewakili jarak tempuh antar simpul yang diukur dalam satuan kilometer. Data jaringan jalan dan wilayah tersebut diperoleh dari Ina Geoportal (2025), sedangkan data posko evakuasi, titik banjir, dan lokasi rawan banjir diperoleh dari wawancara dengan petugas BPBD Kabupaten Tuban. Seluruh data didigitasi menggunakan QGIS dan diverifikasi dengan Google Maps. Parameter evaluasi yang digunakan adalah optimalitas lintasan dan kompleksitas waktu.

Secara keseluruhan, tahapan penelitian meliputi: (1) pemetaan wilayah rawan banjir menggunakan QGIS; (2) pemodelan graf berbobot berdasarkan hasil survei lapangan; (3) penerapan Algoritma Floyd–Warshall; (4) penerapan Algoritma Greedy; (5) analisis dan perbandingan kedua algoritma berdasarkan optimalitas lintasan dan kompleksitas waktu. Alur penelitian secara lengkap disajikan pada Gambar 1 berikut.

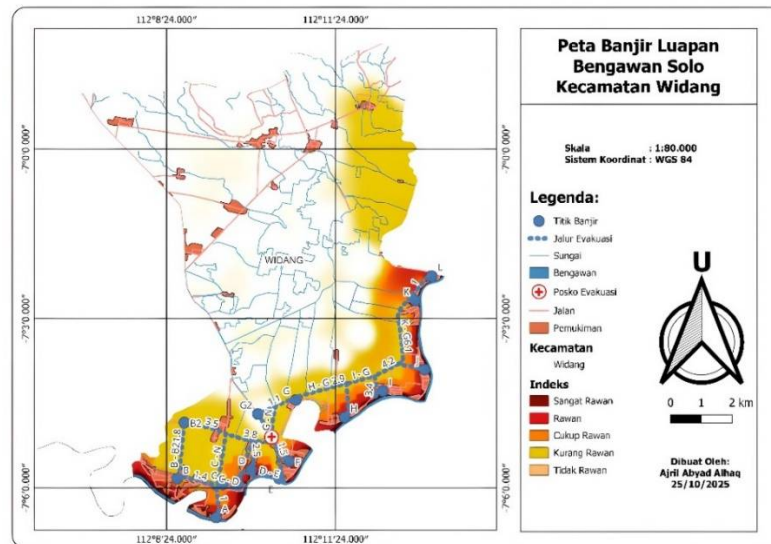


Gambar 1. Diagram Alir Tahapan Penelitian

Hasil dan Pembahasan

Penyusunan Peta

Peta wilayah penelitian dalam studi ini disusun langsung oleh peneliti melalui pengamatan lapangan dan analisis geografis Kecamatan Widang, Kabupaten Tuban, dengan menggunakan perangkat lunak QGIS. Proses ini mencakup identifikasi titik-titik rawan banjir, jalur penghubung antar wilayah, serta penentuan lokasi pos evakuasi. Hasil pemetaan tersebut kemudian dijadikan dasar dalam memodelkan jaringan jalan ke dalam struktur graf berbobot, sebagaimana ditunjukkan pada Gambar 2.



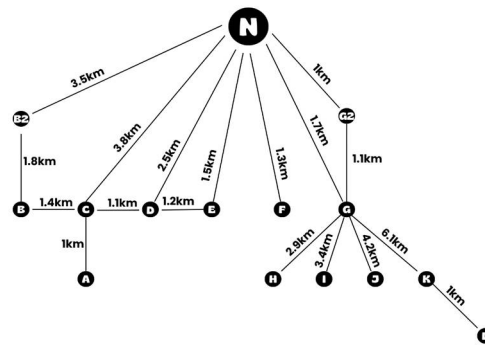
Gambar 2. Peta Banjir Luapan Bengawan Solo Kecamatan Widang

Peta ini menggambarkan distribusi tingkat kerentanan banjir di Kecamatan Widang, Kabupaten Tuban, yang disebabkan oleh luapan Sungai Bengawan Solo. Tingkat kerentanan banjir diwakili oleh berbagai warna dan termasuk dalam sejumlah kategori, dari sangat rentan hingga tidak rentan. Daerah dengan kerentanan tinggi biasanya ditemukan di dekat daerah pemukiman dan aliran sungai.

Selain itu, peta ini menunjukkan jaringan jalan, titik banjir, rute evakuasi, dan lokasi pos evakuasi yang semuanya berfungsi sebagai dasar untuk analisis rute evakuasi. Untuk menetapkan rute evakuasi banjir, jaringan jalan yang tetap dapat dilalui selama banjir direpresentasikan sebagai rute evakuasi dan kemudian diubah menjadi grafik berbobot untuk penerapan Algoritma Greedy dan Floyd-Warshall.

Pemodelan Graf dan Penentuan Bobot

Teori graf adalah kajian matematis terhadap graf $G = (V, E)$ beserta sifat-sifat struktural dan relasi yang terkandung didalamnya (Agusnur, 2025). Jaringan rute evakuasi pada peta direpresentasikan sebagai grafik berbobot, dengan setiap simpul mewakili pos evakuasi, persimpangan jalan, dan titik banjir, dan setiap sisi mewakili jalur penghubung antar simpul (Safetra dkk., 2026). Jarak dalam kilometer antara setiap titik diwakili oleh bobot pada sisi grafik. Algoritma Floyd-Warshall dan Greedy digunakan untuk menentukan rute evakuasi berdasarkan model grafik berbobot ini (Munawwir dkk., 2023), sebagaimana ditunjukkan pada Gambar 3.



Gambar 3. Graf Berbobot Rute Evakuasi Banjir di Kecamatan Widang

Graf berbobot pada Gambar 3 terdiri atas 15 simpul dengan himpunan $V = (A, B, B2, C, D, E, F, G, G2, H, I, J, K, L, N)$, dimana simpul A hingga L merepresentasikan titik banjir, $B2$ dan $G2$ sebagai jalur alternatif, serta simpul N sebagai lokasi posko evakuasi. Bobot pada setiap sisi menyatakan jarak tempuh antar simpul dalam satuan kilometer, yang rinciannya disajikan pada Tabel 1.

Tabel 1. Bobot Jarak pada Tiap Pasangan Simpul Graf Jaringan Jalan

Rute (E)	Jarak (w)	Rute (E)	Jarak (w)
$A - C$	= 1 km	$F - N$	= 1.3 km
$B - B2$	= 1.8 km	$G - N$	= 1.7 km
$B2 - N$	= 3.5 km	$G - G2$	= 1.1 km
$B - C$	= 1.4 km	$G2 - N$	= 1 km
$C - N$	= 3.8 km	$H - G$	= 2.9 km
$C - D$	= 1.1 km	$I - G$	= 3.4 km
$D - N$	= 2.5 km	$J - G$	= 4.2 km
$D - E$	= 1.2 km	$K - G$	= 6.1 km
$E - N$	= 1.5 km	$L - K$	= 1 km

Tabel 1 menyajikan jarak antar simpul. Kemudian setiap simpul menunjukkan lokasi terdampak banjir yang terhubung melalui sisi-sisi graf, dengan bobot sisi menyatakan jarak antar simpul dalam satuan kilometer. Nilai bobot tersebut menjadi parameter utama dalam perhitungan lintasan terpendek dari setiap titik banjir menuju pos evakuasi menggunakan Algoritma Floyd-Warshall dan Greedy. Selanjutnya, penandaan masing-masing simpul pada graf disajikan pada Tabel 2.

Tabel 2. Tanda Simpul pada Graf Jalur Evakuasi Kec. Widang

No	Nama Desa	Tanda di Peta
1.	Ngadipuro	A
2.	Patihan	B
3.	Ngadipuro	C

No	Nama Desa	Tanda di Peta
4.	Ngadirejo	<i>D</i>
5.	Widang	<i>E</i>
6.	Widang	<i>F</i>
7.	Banjar	<i>G</i>
8.	Tegalsari	<i>H</i>
9.	Kedungharjo	<i>I</i>
10.	Tegalrejo	<i>J</i>
11.	Simorejo	<i>K</i>
12.	Duri Kulon	<i>L</i>

Penerapan Algoritma Floyd-Warshall

Algoritma Floyd–Warshall dalam teori graf digunakan untuk mencari lintasan terpendek antara semua simpul (*all-pairs shortest path*) dalam suatu graf berbobot (Mustakim dkk., 2026). Bentuk rekursif Algoritma Floyd–Warshall adalah:

$$D^{(k)}[i][j] = \min(D^{(k-1)}[i][j], D^{(k-1)}[i][k] + D^{(k-1)}[k][j]) \quad (1)$$

Untuk setiap pasangan *i* dan *j*

1. Jarak dari *i* ke *j*
2. Jarak *i* ke *j* lewat titik *k*

Ambil yang paling kecil. Kemudian bentuk implementasi sederhana secara iterasi:

$$d[i][j] = \min(d[i][j], d[i][k] + d[k][j]) \quad (2)$$

Inisialisasi matriks awal (Salsadina & Rosyida, 2025):

$$d[i][j] = \begin{cases} 0 & \text{jika } i = j \\ w(i, j) & \text{jika ada sisi langsung} \\ \infty & \text{jika tidak ada sisi} \end{cases}$$

Untuk menentukan jarak terpendek pada graf rute evakuasi, Algoritma Floyd–Warshall diawali dengan membentuk matriks jarak awal $D^{(0)}$ berdasarkan bobot sisi yang terhubung langsung, kemudian diiterasi dengan mempertimbangkan setiap simpul sebagai titik perantara. Dengan subgraf $(A, C, B, B2, D, E, N)$, diperoleh matriks $D^{(0)}$ sebagai berikut:

$$D^{(0)} = \begin{pmatrix} 0 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & 1.4 & \infty & 1.1 & \infty & 3.8 \\ \infty & 1.4 & 0 & 1.8 & \infty & \infty & \infty \\ \infty & \infty & 1.8 & 0 & \infty & \infty & 3.5 \\ \infty & 1.1 & \infty & \infty & 0 & 1.2 & 2.5 \\ \infty & \infty & \infty & \infty & 1.2 & 0 & 1.5 \\ \infty & 3.8 & \infty & 3.5 & 2.5 & 1.5 & 0 \end{pmatrix} \quad (3)$$

$D^{(1)} = D^{(0)}$, karena A hanya terhubung ke C , maka tidak ada jalur lebih pendek terbentuk. Untuk iterasi $k = C$, gunakan persamaan (2):

Pada $A \rightarrow B$ lewat C :

$$d(A, B) = \min(\infty, 1 + 1.4) = 2.4 \text{ km}$$

Untuk $A \rightarrow D$ lewat C :

$$d(A, D) = \min(\infty, 1 + 1.1) = 2.1 \text{ km}$$

Kemudian, pada $A \rightarrow N$ lewat C :

$$d(A, N) = \min(\infty, 4.8) = 4.8 \text{ km}$$

Maka matriks menjadi:

$$D^{(2)} = \begin{pmatrix} 0 & 1 & 2.4 & \infty & 2.1 & \infty & 4.8 \\ 1 & 0 & 1.4 & \infty & 1.1 & \infty & 3.8 \\ 2.4 & 1.4 & 0 & 1.8 & 2.5 & \infty & 5.2 \\ \infty & \infty & 1.8 & 0 & \infty & \infty & 3.5 \\ 2.1 & 1.1 & 2.5 & \infty & 0 & 1.2 & 2.5 \\ \infty & \infty & \infty & \infty & 1.2 & 0 & 1.5 \\ 4.8 & 3.8 & 5.2 & 3.5 & 2.5 & 1.5 & 0 \end{pmatrix} \quad (4)$$

Iterasi selanjutnya dilakukan dengan prinsip yang sama hingga mencapai kondisi stabil (tidak terdapat perubahan nilai matriks). Matriks akhir dinotasikan sebagai $D^{(*)}$, yang diperoleh sebagai hasil akhir proses Algoritma Floyd-Warshall.

$$D^* = \begin{pmatrix} 0 & 1 & 2.4 & 4.2 & 2.1 & 3.3 & 4.6 \\ 1 & 0 & 1.4 & 3.2 & 1.1 & 2.3 & 3.6 \\ 2.4 & 1.4 & 0 & 1.8 & 2.5 & 3.7 & 5.2 \\ 4.2 & 3.2 & 1.8 & 0 & 4.3 & 5.0 & 3.5 \\ 2.1 & 1.1 & 2.5 & 4.3 & 0 & 1.2 & 2.5 \\ 3.3 & 2.3 & 3.7 & 5.0 & 1.2 & 0 & 1.5 \\ 4.6 & 3.6 & 5.2 & 3.5 & 2.5 & 1.5 & 0 \end{pmatrix} \quad (5)$$

Sebagai ilustrasi, disajikan contoh perhitungan lintasan terpendek menggunakan Algoritma Floyd-Warshall dengan simpul awal A menuju posko evakuasi N , menghasilkan rute $A - C - D - N$ dengan jarak optimal 4.6 km. Proses perhitungan dilakukan secara iteratif hingga diperoleh lintasan dengan jarak minimum. Hasil selengkapnya disajikan pada Tabel 3 berikut.

Tabel 3. Rute Terpendek Evakuasi Banjir Terpendek

No	Titik Awal	Rute Terpendek (Floyd-Warshall)	Jarak Total (km)
1	A	$A - C - D - N$	4.6 km
2	B	$B - C - D - N$	5.0 km
3	C	$C - D - N$	3.6 km
4	D	$D - N$	2.5 km
5	E	$E - N$	1.5 km

6	<i>F</i>	<i>F - N</i>	1.3 km
7	<i>G</i>	<i>G - N</i>	1.7 km
8	<i>H</i>	<i>H - G - N</i>	4.6 km
9	<i>I</i>	<i>I - G - N</i>	5.1 km
10	<i>J</i>	<i>J - G - N</i>	5.9 km
11	<i>K</i>	<i>K - G - N</i>	7.8 km
12	<i>L</i>	<i>L - K - G - N</i>	8.8 km

Pada Tabel 3, hasil Algoritma Floyd-Warshall menunjukkan bahwa jarak terpendek menuju pos evakuasi *N* bervariasi untuk setiap titik awal, yaitu 4.6 km dari titik *A*, 5.0 km dari titik *B*, 3.6 km dari titik *C*, 2.5 km dari titik *D*, 1.5 km dari titik *E*, 1.3 km dari titik *F*, 1.7 km dari titik *G*, 4.6 km dari titik *H*, 5.1 km dari titik *I*, 5.9 km dari titik *J*, 7.8 km dari titik *K*, dan 8.8 km dari titik *L*.

Penerapan Algoritma Greedy

Algoritma Greedy bekerja dengan memilih keputusan terbaik pada setiap langkah berdasarkan kriteria lokal, tanpa mempertimbangkan konsekuensi secara global (Jofie dkk., 2020). Dalam penentuan rute evakuasi, algoritma ini memilih sisi dengan jarak terpendek dari simpul saat ini menuju simpul berikutnya hingga mencapai tujuan (Sembiring dkk., 2025).

$$v_{\text{next}} = \arg \min_{v \in \text{Adj}(u)} w(u, v) \quad (6)$$

Total jarak:

$$D = \sum_{i=1}^k w(v_i, v_{i+1}) \quad (7)$$

Atau:

$$D = w(v_1, v_2) + w(v_2, v_3) + w(v_3, v_4) + \dots + w(v_k, v_k + 1) \quad (8)$$

Keterangan:

D : Total jarak rute

$w(v_i, v_{i+1})$: Jarak antara simpul ke-*i* dan simpul ke-*i* + 1

k : jumlah sisi (edge) yang dilalui dalam rute

Sebagai ilustrasi penerapan Algoritma Greedy, berikut disajikan proses iterasi perhitungan lintasan terpendek dengan menggunakan titik awal *A* menuju pos evakuasi *N*. Pada setiap iterasi, algoritma memilih jalur dengan jarak terkecil dari titik yang sedang dikunjungi hingga akhirnya mencapai simpul tujuan. Hasil proses

iterasi tersebut disajikan pada Tabel 4 berikut.

Tabel 4. Rute Iterasi Algoritma Greedy (Titik Awal A)

Iterasi	Titik	Pilihan Jalur	Jarak (km)	Jarak Terpilih	Jarak dipilih
1	A	A – C	1.0	A – C	1.0
2	C	C – B (1.4). C – D (1.1). C – N (3.8)	1.4; 1.1; 3.8	C – D	1.1
3	D	D – E (1.2). D – N (2.5)	1.2; 2.5	D – E	1.2
4	E	E – N	1.5	E – N	1.5

Gunakan persamaan (8) untuk menghitung total jarak yang ditempuh oleh Algoritma Greedy:

$$D = w(A, C) + w(C, D) + w(D, E) + w(E, N)$$

$$D = 1 + 1.1 + 1.2 + 1.5 = 4.8 \text{ km}$$

Tabel 4 menyajikan proses iterasi Algoritma Greedy dengan simpul awal *A* sebagai titik banjir dan simpul *N* sebagai pos evakuasi. Pada setiap iterasi, algoritma memilih sisi dengan bobot minimum secara lokal, sehingga terbentuk lintasan *A – C – D – E – N* dengan jarak tempuh 4.8 km. Lintasan lainnya dihitung dengan prosedur yang sama, dan hasil lengkapnya disajikan pada Tabel 5.

Tabel 5. Rute Terpendek Evakuasi Banjir Terpendek Algoritma Greedy

No	Titik Awal	Rute Terpendek (Greedy)	Jarak Total (km)
1	A	A – C – D – E – N	4.8 km
2	B	B – C – D – E – N	5.2 km
3	C	C – D – E – N	3.8 km
4	D	D – N	2.5 km
5	E	E – N	1.5 km
6	F	F – N	1.3 km
7	G	G – N	1.7 km
8	H	H – G – N	4.6 km
9	I	I – G – N	5.1 km
10	J	J – G – N	5.9 km
11	K	K – G – N	7.8 km
12	L	L – K – G – N	8.8 km

Tabel 5 menunjukkan hasil perhitungan jarak terpendek menggunakan Algoritma Greedy dari setiap titik awal menuju pos evakuasi N. Jarak yang diperoleh bervariasi mulai dari 1.3 km hingga 8.8 km.

Analisis dan Perbandingan

Hasil dari Algoritma Floyd-Warshall pada Tabel 3 dan Algoritma Greedy pada Tabel 5 selanjutnya dibandingkan untuk melihat selisih jarak dan rute yang dihasilkan oleh kedua algoritma tersebut, sebagaimana disajikan pada Tabel 6 berikut:

Tabel 6. Perbandingan Rute Evakuasi Floyd-Warshall dan Greedy

No	Titik	Greedy	Rute Greedy	Floyd-Warshall	Rute	Selisih
1	A	4.8 km	A – C – D – E – N	4.6 km	A – C – D – N	0.2 km
2	B	5.2 km	B – C – D – E – N	5.0 km	B – C – D – N	0.2 km
3	C	3.8 km	C – D – E – N	3.6 km	C – D – N	0.2 km
4	D	2.7 km	D – N	2.5 km	D – N	0 km
5	E	1.5 km	E – N	1.5 km	E – N	0 km
6	F	1.3 km	F – N	1.3 km	F – N	0 km
7	G	1.7 km	G – N	1.7 km	G – N	0 km
8	H	4.6 km	H – G – N	4.6 km	H – G – N	0 km
9	I	5.1 km	I – G – N	5.1 km	I – G – N	0 km
10	J	5.9 km	J – G – N	5.9 km	J – G – N	0 km
11	K	7.8 km	K – G – N	7.8 km	K – G – N	0 km
12	L	8.8 km	L – K – G – N	8.8 km	L – K – G – N	0 km

Perbedaan hasil antara kedua algoritma terjadi karena Algoritma Greedy hanya mempertimbangkan jarak terdekat secara lokal, sehingga pada titik A, B, dan C memilih jalur melalui E yang justru lebih panjang (0.2 km). Sementara Floyd-Warshall mengevaluasi semua kemungkinan jalur, sehingga menemukan rute A – C – D – N (4.6 km) yang lebih pendek. Kelemahan Greedy inilah yang menyebabkan selisih 0.2 km.

Dalam evakuasi banjir nyata, selisih 0.2 km dapat berarti tambahan waktu 2–3 menit yang cukup krusial. Meskipun Greedy lebih cepat secara komputasi, rutanya tidak selalu optimal. Oleh karena itu, Greedy lebih cocok untuk situasi darurat yang membutuhkan perhitungan cepat, sedangkan Floyd-Warshall lebih tepat untuk perencanaan rute evakuasi jangka panjang.

Kompleksitas Algoritma Floyd-Warshall

Algoritma Floyd-Warshall merupakan metode pencarian lintasan terpendek untuk semua simpul (*all-pairs shortest path*). Algoritma ini bekerja dengan melakukan pembaruan matriks jarak melalui tiga perulangan, yaitu:

$$\text{for } k = 1 \text{ to } n \tag{9}$$

$$\text{for } i = 1 \text{ to } n \tag{10}$$

$$\text{for } j = 1 \text{ to } n \quad (11)$$

Karena terdapat tiga kali perulangan masing-masing sepanjang n , maka kompleksitas waktu dapat dinyatakan:

$$T(n) = n \times n \times n = n^3 \quad (12)$$

Sehingga waktu kompleksitas Algoritma Floyd-Warshall:

$$O(n^3) \quad (13)$$

Dalam studi ini, grafik rute evakuasi terdiri dari 15 simpul, termasuk 12 titik banjir ($A - L$), satu pos evakuasi (N), dan dua simpul rute alternatif ($B2$ dan $G2$). Oleh karena itu, jumlah operasi dasar yang diperkirakan dilakukan adalah:

$$15^3 = 3375 \text{ Operasi}$$

Jumlah ini masih relatif kecil sehingga Floyd-Warshall dapat dijalankan dengan cepat pada skala graf penelitian. Namun, apabila jumlah simpul meningkat, pertumbuhan operasi akan bersifat kubik. Sebagai ilustrasi:

Tabel 7. Jumlah Simpul dan Estimasi Operasi

Jumlah Simpul	Estimasi Operasi (n^3)
15	3375
50	125,000
100	1,000,000

Hal ini menunjukkan bahwa pada graf berskala besar. Floyd-Warshall akan membutuhkan waktu komputasi yang jauh lebih tinggi.

Kompleksitas Algoritma Greedy

Algoritma Greedy memilih tepi bobot minimum dari simpul saat ini ke simpul berikutnya hingga tujuan tercapai. Proses ini hanya memeriksa simpul-simpul yang berdekatan secara langsung. Ketika grafik diwakili menggunakan daftar *adjacency*, kompleksitas waktu algoritma Greedy dapat diekspresikan sebagai:

$$O(V + E) \quad (14)$$

Dengan:

V = jumlah simpul; E = jumlah sisi

Pada graf penelitian ini terdapat 15 simpul dan 18 sisi, sehingga estimasi operasi yang dilakukan jauh lebih kecil dibandingkan Floyd-Warshall. Secara teoritis, pertumbuhan kompleksitas Greedy bersifat linear terhadap jumlah simpul dan sisi, sehingga lebih efisien untuk graf berskala besar

Kesimpulan

Berdasarkan perbandingan kedua algoritma, dari 12 titik banjir yang dianalisis, sembilan titik menghasilkan rute dan jarak yang sama, sementara tiga titik lainnya menunjukkan selisih hingga 0.2 km dengan Floyd–Warshall memberikan hasil yang lebih optimal. Dari sisi kompleksitas waktu, Floyd–Warshall memiliki kompleksitas $O(n^3)$ yang meningkat secara kubik, sedangkan Greedy lebih efisien meskipun tidak selalu menjamin solusi optimal secara global. Oleh karena itu, Floyd–Warshall lebih tepat untuk perencanaan pra-bencana yang memerlukan akurasi tinggi, sementara Greedy lebih cocok untuk kondisi darurat yang membutuhkan perhitungan cepat.

Daftar Pustaka

- Afandi, N., & Mayasari, Z. M. (2021). An Evacuation Route in Bengkulu City Based on Fuzzy Dijkstra Algorithm. *Journal of Physics: Conference Series*, 1863(1), 12007. <https://iopscience.iop.org/article/10.1088/1742-6596/1863/1/012007/meta>
- Agusnur, A. (2025). Penggunaan Teori Graf dalam Optimasi Jaringan Transportasi Kota. *Jurnal Matematika dan Aplikasi (JMA)*, 1(1), 33-39. <https://ejournal.samudrailmu.com/index.php/jma/article/view/16>
- Azis, H., Lantara, D., & Salim, Y. (2018). Comparison of Floyd-Warshall Algorithm and Greedy Algorithm in Determining the Shortest Route. *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, 294–298. <https://doi.org/10.1109/EIConCIT.2018.8878582>
- Budhiana, J., Dewi, R., Janatri, S., & Fatmala, S. D. (2023). Membangun Kesiapsiagaan Masyarakat dalam Menghadapi Bencana Melalui Sosialisasi dan Edukasi Modal Sosial. *Abdimas Galuh*, 5(2), 1269–1276. <https://doi.org/10.25157/ag.v5i2.10751>
- Darmadi, D., Diansyah, T. M., & Handoko, D. (2023). Penerapan Algoritma Floyd Warshall dengan Menggunakan Euclidean Distance dalam Menentukan Rute Terbaik. *Jurnal Ilmu Komputer dan Sistem Informasi*, 2(2), 195–205. <https://doi.org/10.70340/jirsi.v2i2.68>
- Denisya, F., & Hertati, D. (2024). The Role of the Regional Disaster Management Agency in Flood Disaster Mitigation Efforts in Banyulegi Village, Dawarblandong District, Mojokerto Regency. *Jurnal Aristo (Social, Politic, Humaniora)*, 12(1), 75–101. <https://doi.org/10.24269/ars.v12i1.7313>
- Harahap, S. A., & Triase, T. (2024). Greedy Algorithm to Support the Decision of Choosing the Fastest Aid Distribution Route After Flooding. *Sistemasi: Jurnal Sistem Informasi*, 13(4), 1689–1704.
-

- <https://doi.org/10.32520/stmsi.v13i4.4345>
- Ina Geoportal. (2025). *Data Wilayah Indonesia*.
<https://tanahair.indonesia.go.id/portal-web/webmap>
- Jofie, M. Z., Bahri, S., & Baqi, A. I. (2020). Aplikasi Algoritma Greedy untuk Pewarnaan Wilayah pada Peta Kota Padang Berbasis Teorema Empat Warna. *Jurnal Matematika UNAND*, 9(4), 294–301. <https://doi.org/10.25077/jmu.9.4.294-301.2020>
- Kekal, H. P., Gata, W., Nurdiani, S., Rini, A. J. S., & Wita, D. S. (2021). Analisa Pencarian Rute Tercepat Menuju Tempat Wisata Pulau Kumala Kota Tenggara Menggunakan Algoritma Greedy. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, 7(1), 9–15. <https://doi.org/10.35329/jiik.v7i1.179>
- Mahardika, D., & Setianingsih, E. L. (2018). Manajemen Bencana oleh Badan Penanggulangan Bencana Daerah (BPBD) dalam Menanggulangi Banjir di Kota Semarang. *Journal of Public Policy and Management Review*, 7(2), 502–518. <https://doi.org/10.14710/jppmr.v7i2.20242>
- Mishra, G., Mazumdar, S., & Pal, A. (2018). Improved Algorithms for the Evacuation Route Planning Problem. *Journal of Combinatorial Optimization*, 36(1), 280–306. <https://doi.org/10.1007/S10878-016-0082-0>
- Munawwir, Z., Sari, L. D. K., Zairozie, A. Z., & Hadi, S. (2023). Penerapan Graf Berbobot dan Algoritma Dijkstra untuk Menentukan Rute Optimal dari Pusat Kota ke Beberapa Objek Wisata di Kabupaten Situbondo. *Jurnal IKA PGSD (Ikatan Alumni PGSD) UNARS*, 14(2), 212–222. <https://doi.org/10.36841/pgsdunars.v14i2.3997>
- Mustakim, M., Wahyuni, T., & Rahman, F. I. (2026). Implementasi Algoritma Floyd-Warshall untuk Menentukan Jarak Terpendek dalam Sistem Pengantaran Pos di Kecamatan Mariso. *Jurnal Intelek dan Cendekiawan Nusantara*, 3(01), 153–165. <https://jicnusanantara.com/index.php/jicn/article/view/6677>
- Nurdiansyah, R., Mindarta, E. K., Larasati, A., Darmawan, V. E. B., Widowati, T. R., Haque, S. A., & Azzahra, P. D. (2025). Sosialisasi Program Sekolah Siaga Bencana guna Meningkatkan Kemampuan Kesiapsiagaan terhadap Bencana. *Jurnal Pengabdian Masyarakat dan Aplikasi Teknologi*, 4(1), 13–18. <https://doi.org/10.31284/j.adipati.2025.v4i1.6863>
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *Jurnal Riset Komputer (JURIKOM)*, 5(2), 136–139. <https://doi.org/10.30865/jurikom.v5i2.625>
- Safetra, M. F. C., Desviona, N., Helmina, H., Rianti, A., & Prayogi, M. R. (2026). Penerapan Teori Graf dalam Kehidupan Sehari-hari. *Algoritma: Jurnal*



Matematika, Ilmu Pengetahuan Alam, Kebumihan dan Angkasa, 4(1), 52–68.
<https://doi.org/10.62383/algorithm.v4i1.923>

Salsadina, N. A., & Rosyida, I. (2025). Aplikasi Pencarian Rute Optimal Antar-Objek Wisata di Kabupaten Cilacap Berbasis Algoritma Floyd-Warshall. *Imajiner: Jurnal Matematika dan Pendidikan Matematika*, 7(4), 256–269.
<https://doi.org/10.26877/imajiner.v7i4.23457>

Sembiring, F. G., Nst, S. P. S., Gaol, A. S. L., & Harliana, P. (2025). Implementasi Algoritma Greedy dalam Penentuan Jalur Strategis dari Simpang Pos ke Lapangan Merdeka Kota Medan. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 423–428. <https://doi.org/10.36040/jati.v9i1.12343>

Tani, K., & Yamamoto, K. (2020). Search Methods for Evacuation Routes During Torrential Rain Disasters Using Genetic Algorithms and GIS. *J Geogr Inf Syst*, 12(03), 256–274. <https://doi.org/10.4236/jgis.2020.123016>

Wayahdi, M. R., Ginting, S. H. N., & Syahputra, D. (2021). Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path. *International Journal of Advances in Data and Information Systems*, 2(1), 45–52.
<https://doi.org/10.25008/ijadis.v2i1.1206>