Journal of Computer and Information System (J-CIS)
Vol.8, No.2, October 2025, Page 119-131
ISSN (print): 2622-5859, ISSN (online): 2622-0881
DOI: https://doi.org/10.31605/jcis.v8i2

Metode Optimasi Cerdas Untuk Pencarian Jalur Evakuasi Bencana di Kabupaten Mamuju Menggunakan *Ant Colony Optimization*

Nur Halisah[1], Nurmadinah[2], Ariqah Maheswari Artalaysia Paturusi[3], Wawan Firgiawan*[4]

[1,2,3,4] Program Studi Teknik Informatika, Universitas Sulawesi Barat
E-mail: *[1]nurhalisah@unsulbar.ac.id, [2]nurmadinah@unsulbar.ac.id, [3]ariqah.maheswari@unsulbar.ac.id, [4] wawanfirgiawan@unsulbar.ac.id

Abstrak

Proses evakuasi bencana sering kali menghadapi kendala karena algoritma tradisional seperti Breadth-First Search (BFS) dan Depth-First Search (DFS) kurang adaptif terhadap kondisi jalan yang berubah akibat bencana. Hal ini dapat menyebabkan jalur evakuasi yang dipilih menjadi kurang efisien. Penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma pencarian jalur evakuasi, yaitu Breadth-First Search (BFS), Depth-First Search (DFS), dan Ant Colony Optimization (ACO), dalam menemukan jalur evakuasi yang paling efisien. Sebanyak 20 dataset digunakan dalam pengujian, dengan menguji setiap algoritma terhadap jalur optimal yang diperoleh secara manual sebagai data aktual. Dua skenario pengujian diterapkan dalam penelitian ini, yaitu Single Vertex Single Goals (SVSG) dan Single Vertex Multi Goals (SVMG). Hasil pengujian menunjukkan algoritma ACO konsisten menghasilkan jalur yang identik dengan jalur optimal, dengan nilai Root Mean Square Error (RMSE) sebesar 0.0, menandakan bahwa algoritma ini memberikan performa terbaik tanpa penyimpangan dari jalur aktual. Algoritma DFS mencatatkan RMSE sebesar 4.72, menunjukkan kinerja yang lebih baik daripada BFS, namun masih terdapat penyimpangan pada beberapa dataset. Sementara itu, BFS memiliki RMSE tertinggi sebesar 6.91, menunjukkan bahwa algoritma ini seringkali menghasilkan jalur yang lebih panjang dan kurang efisien. Hasil pengujian membuktikan algoritma ACO terbukti menjadi pilihan paling efisien untuk optimasi jalur evakuasi dalam kedua skenario pengujian.

Kata kunci — Evakuasi bencana, Pencarian jalur*, Breadth-First Search, Depth-First Search , Ant Colony Optimization.*

Abstract

Disaster evacuation processes often face obstacles because traditional algorithms such as Breadth-First Search (BFS) and Depth-First Search (DFS) are less adaptive to road conditions that change due to disasters. This can lead to the selection of less efficient evacuation routes. This study aims to compare the performance of three evacuation route-finding algorithms, namely Breadth-First Search (BFS), Depth-First Search (DFS), and Ant Colony Optimization (ACO), in finding the most efficient evacuation routes. A total of 20 datasets were used in this experiment, with each algorithm tested against the optimal routes obtained manually as the ground truth. Two testing scenarios were applied in this study, namely Single Vertex Single Goals (SVSG) and Single Vertex Multi Goals (SVMG). The results show that the ACO algorithm consistently produced routes identical to the optimal ones, with a Root Mean Square Error (RMSE) value of 0.0, indicating the best performance without any deviation from the actual routes. The DFS algorithm recorded an RMSE of 4.72, demonstrating better performance than BFS but still showing deviations in some datasets. Meanwhile, BFS had the highest RMSE of 6.91, indicating that this algorithm often produced longer and less efficient routes. These findings confirm that the ACO algorithm is the most efficient choice for evacuation route optimization in both testing scenarios.

Keywords — *Disaster evacuation, Pathfinding, Breadth-First Search, Depth-First Search , Ant Colony Optimization.*

1. PENDAHULUAN

Indonesia merupakan salah satu negara dengan tingkat kerentanan bencana alam tertinggi di dunia karena posisinya yang berada di pertemuan Lempeng Eurasia, Indo-Australia, dan Pasifik. Kondisi tektonik ini menjadikan wilayah Indonesia sering mengalami gempa bumi, tsunami, erupsi gunung api, banjir, hingga tanah longsor [1]. Berdasarkan data Badan Nasional Penanggulangan Bencana (BNPB), lebih dari 3.000 kejadian bencana tercatat setiap tahunnya dalam satu dekade terakhir. Sulawesi Barat, khususnya Kabupaten Mamuju, menjadi salah satu wilayah rawan gempa, terlihat dari kejadian gempa magnitudo 6,2 yang mengguncang Mamuju dan Majene pada 15 Januari 2021, mengakibatkan 105 korban jiwa, ribuan bangunan rusak, dan lebih dari 15.000 warga mengungsi [2], [3]. Kondisi ini menunjukkan bahwa kesiapsiagaan bencana dan penentuan jalur evakuasi yang efektif memiliki peran vital dalam mitigasi risiko korban jiwa.

Dalam proses evakuasi, ketepatan jalur evakuasi sangat menentukan waktu tiba ke titik aman. Jalur evakuasi yang tidak optimal dapat menyebabkan penumpukan massa, keterlambatan evakuasi, hingga meningkatnya risiko kematian. Oleh karena itu, penentuan rute evakuasi perlu mempertimbangkan faktor jarak, kondisi jaringan jalan, topografi wilayah, serta adanya hambatan seperti reruntuhan bangunan dan kemacetan. Penelitian-penelitian sebelumnya telah menggunakan algoritma pencarian jalur untuk pemodelan evakuasi. Budiarto et al. (2020) menerapkan *Breadth-First Search* (BFS) untuk evakuasi banjir, namun algoritma ini hanya efektif pada graf statis dan tidak mampu menyesuaikan diri terhadap perubahan lintasan akibat bencana [4]. Penelitian lain menggunakan *Depth-First Search* (DFS), tetapi metode ini tidak menjamin ditemukannya jalur terpendek dan cenderung menjelajahi jalur secara acak hingga mencapai titik tujuan [5].

Secara teoretis, BFS bekerja dengan prinsip eksplorasi level per level pada simpul graf dan menjamin ditemukannya jalur terpendek dalam graf tak berbobot [6]. Sementara DFS melakukan penelusuran hingga cabang terdalam sebelum mundur *(backtracking),* tetapi sering gagal menghasilkan solusi optimal jika terdapat banyak kombinasi jalur [7], [8]. Keterbatasan algoritma konvensional ini menjadi alasan perlunya pendekatan yang lebih adaptif, cerdas, dan mampu bekerja dalam lingkungan dinamis.

Ant Colony Optimization (ACO) merupakan salah satu metode optimasi cerdas berbasis metaheuristik yang terinspirasi dari perilaku semut dalam mencari makanan. Ketika semut menemukan makanan, semut akan meninggalkan jejak feromon pada jalur yang dilaluinya. Semakin banyak semut melewati jalur tersebut, semakin kuat intensitas feromon sehingga menarik semut lain untuk mengikuti jalur yang sama. Dalam pemodelan komputasi, ACO bekerja melalui tiga tahapan utama: (1) konstruksi solusi oleh populasi semut yang bergerak dari simpul awal ke simpul tujuan, (2) pembaruan feromon lokal dan global berdasarkan kualitas jalur yang ditemukan, dan (3) evaporasi feromon untuk menghindari konvergensi dini atau stuck pada solusi lokal [9], [10]. Algoritma ini dinilai unggul dalam memecahkan masalah optimasi jalur seperti *Travelling Salesman Problem* (TSP), sistem transportasi, logistik, hingga pencarian jalur evakuasi bencana [11], [12].

Studi Zhang et al. (2019) menunjukkan bahwa ACO mampu mempercepat proses evakuasi tsunami dengan mempertimbangkan kepadatan penduduk dan kondisi jalan [13]. Sementara itu, Sari et al. (2021) membuktikan bahwa ACO memberikan hasil yang lebih fleksibel dibanding BFS karena dapat mendeteksi dan menghindari jalur yang terputus akibat gempa [14]. Namun, masih terdapat kekurangan dalam penelitian terdahulu, yakni minimnya perbandingan kuantitatif ACO dengan algoritma pencarian jalur konvensional (BFS dan DFS) dalam konteks wilayah rawan gempa di Indonesia, khususnya Kabupaten Mamuju. Selain itu, sebagian besar penelitian hanya berfokus pada skenario *Single Vertex Single Goals* (SVSG), padahal dalam

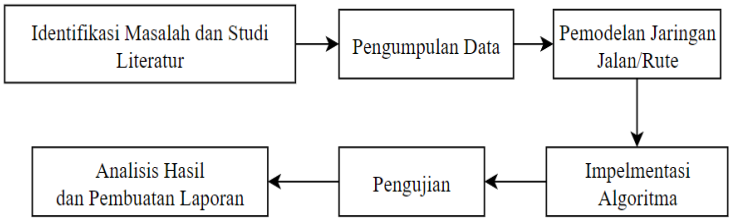
kondisi bencana nyata sering terdapat beberapa titik evakuasi *(Multi Goal)* atau beberapa titik awal evakuasi.

Berdasarkan hal tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma BFS, DFS, dan ACO dalam penentuan jalur evakuasi di Kabupaten Mamuju. Pengujian dilakukan dalam dua skenario, yaitu *Single Vertex Single Goals* (SVSG) dan *Single Vertex Multi Goals* (SVMG), dengan parameter jarak tempuh sebagai indikator utama. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem evakuasi berbasis teknologi komputasi, serta mendukung pemerintah daerah dalam penyusunan rencana mitigasi bencana yang lebih efektif dan adaptif.

2. METODE

2.1 Tahap Penelitian

Tahapan penelitian yang dilakukan untuk menyelesaikan penelitian ini diuraikan seperti pada gambar pada Gambar 1:



Gambar 1 Tahap Penelitian

2.1.1 Identifikasi Masalah dan Studi Literatur

Pada tahap awal, dilakukan identifikasi masalah utama terkait jalur evakuasi bencana, khususnya dalam konteks mitigasi bencana. Literatur yang relevan terkait algoritma pencarian jalur, seperti *Breadth-First Search* (BFS), *Depth-First Search* (DFS), dan *Ant Colony Optimization* (ACO) serta penelitian terdahulu mengenai perencanaan evakuasi bencana juga dikaji untuk mendasari penelitian ini.

2.1.2 Pengumpulan Data

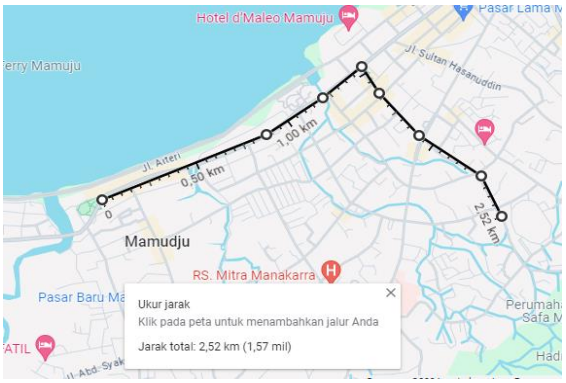
Tahap berikutnya adalah pengumpulan data. Data peta jalan di Kabupaten Mamuju diperoleh menggunakan Google Maps untuk mendapatkan detail jalur yang akurat. Selain itu, data tambahan seperti lokasi titik awal, persimpangan *(vertex),* dan titik tujuan evakuasi juga diidentifikasi untuk keperluan analisis lebih lanjut.

Dari hasil pemetaan tersebut, disusun sebanyak 20 dataset yang mencakup variasi jalur evakuasi. Dataset ini digunakan untuk menguji dua skenario, yaitu:

- SVSG *(Single Vertex Single Goals),* dengan satu titik awal menuju satu titik tujuan evakuasi.
- SVMG *(Single Vertex Multi Goals)*, dengan satu titik awal menuju beberapa titik tujuan evakuasi.

Gambar 2 Peta Wilayah Kota Mamuju (Source: Google Maps)

2.1.3 Pemodelan Jaringan Jalan



Gambar 3 Pengambilan Data Jarak (Source: Google Maps)

Setelah data dikumpulkan, langkah selanjutnya adalah membangun model jaringan jalan. Dalam tahap ini, graf berbobot dibuat untuk merepresentasikan jaringan jalan di Kabupaten Mamuju. Node dalam graf ini mewakili persimpangan jalan, sedangkan bobot merepresentasikan jarak atau waktu tempuh antar Node satu ke Node lainnya yang kemudian direpresentasikan dengan menggunakan *Matrix Adjacency*. Model ini akan menjadi dasar bagi implementasi algoritma yang digunakan pada penelitian ini.

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	-1	2	7	10	5	4	9	4	6	6	6
B	5	-1	4	9	6	7	8	-1	-1	-1	-1	8
C	2	-1	0	7	6	2	9	8	2	-1	10	6
D	3	9	1	0	8	2	9	3	7	-1	6	8
E	7	3	-1	-1	-1	-1	-1	-1	3	-1	9	9
F	10	10	-1	8	2	0	4	3	5	-1	10	6
G	8	8	-1	6	3	5	0	5	9	10	2	5
H	6	1	-1	2	5	3	3	0	1	9	10	5
I	-1	-1	-1	2	1	10	1	1	0	1	10	3
J	-1	2	-1	6	7	8	8	4	4	0	6	9
K	-1	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	2
L	-1	6	3	9	7	8	8	5	1	1	5	0

Gambar 4 Contoh Dataset pada skenario SVMG

Kotak berwarna merah pada Gambar 4 menandakan titik awal evakuasi, sedangkan kotak berwarna hijau menunjukkan titik tujuan atau lokasi aman (terdapat 2 kotak warna hijau). Nilai pada matriks menunjukkan bobot jarak antar node, di mana semakin kecil nilainya maka semakin dekat jarak antar titik. Tetapi, nilai 0 menunjukkan bahwa tidak terdapat koneksi langsung antara dua node, sementara nilai –1 merepresentasikan jalur yang tidak dapat dilalui atau terputus.

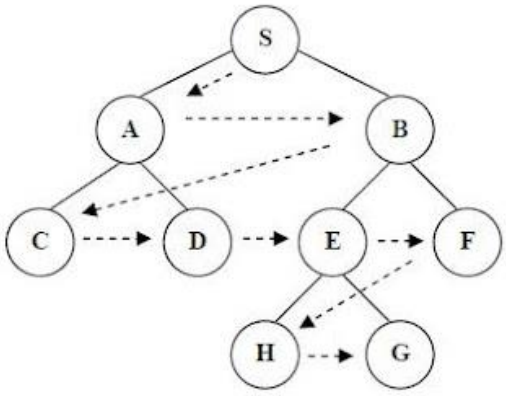
Pada skenario SVSG, hanya terdapat satu node merah sebagai titik awal dan satu node (kotak) hijau sebagai tujuan tunggal. Sedangkan pada SVMG, dapat terdapat lebih dari satu node (kotak) hijau pada dataset *Matrix Adjacency* karena skenario ini melibatkan beberapa titik tujuan potensial. Dalam konteks evakuasi, algoritma pencarian akan dimulai dari node merah dan mencari jalur optimal menuju salah satu node hijau dengan total jarak paling efisien.

2.1.4 Implementasi Algoritma

Penelitian ini akan menerapkan 3 algoritma yang biasa digunakan dalam pencarian jalur. Adapun algoritma tersebut dijelaskan sebagai berikut.

a) *Breadth-First Search* (BFS)

Untuk melakukan pencarian jalur, Algoritma *Breadth-First Search*, juga dikenal sebagai BFS, melakukan pencarian jalur dengan menelusuri simpul-simpul pohon dari simpul utama hingga menemukan jalur ke simpul yang dimaksud [15]. Berikut adalah skema pencarian BFS.

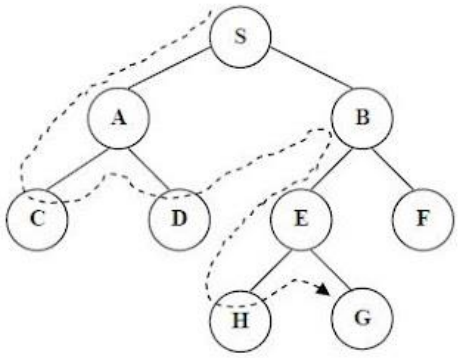


Gambar 5 Skema Pencarian BFS

Salah satu keunggulan BFS adalah kemampuannya menemukan jalur terpendek dalam graf tidak berbobot, karena BFS menjelajahi semua simpul di tingkat yang sama sebelum melanjutkan ke tingkat berikutnya. BFS sering digunakan dalam berbagai aplikasi seperti penentuan jalur terpendek, pencarian jalur dalam labirin, dan pemetaan jaringan.

b) *Depth-First Search* (DFS)

Algoritma *Depth-First Search* (DFS) adalah salah satu algoritma pencarian yang digunakan untuk menjelajahi atau menelusuri graf atau pohon. Algoritma ini bekerja dengan mengeksplorasi jalur dari simpul awal ke simpul yang dalam terlebih dahulu sebelum kembali untuk mengeksplorasi cabang lainnya. DFS bergerak secara rekursif atau menggunakan tumpukan (stack) untuk mengunjungi simpul-simpul yang belum dikunjungi, menelusuri satu jalur hingga mencapai simpul tujuan atau simpul akhir dari graf [15]. Berikut adalah skema pencarian DFS.

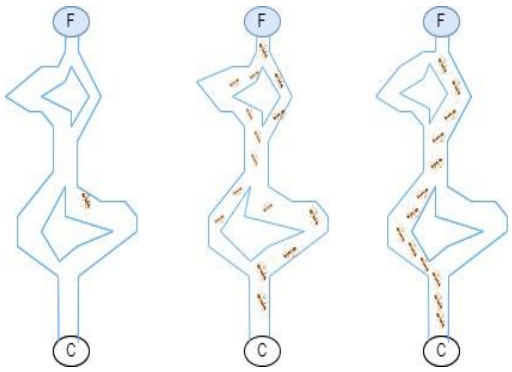


Gambar 6 Skema Pencarian DFS

Karena DFS menjelajahi jalur hingga kedalaman maksimum sebelum kembali, algoritma ini cocok digunakan pada masalah yang memerlukan pencarian menyeluruh dalam graf yang besar atau untuk menemukan solusi yang berada jauh dari simpul awal. Namun, DFS tidak selalu menemukan solusi terpendek karena fokusnya pada kedalaman pencarian, bukan pada jarak atau biaya antar simpul.

c) Ant Colony Optimization

Ant Colony Optimization (ACO) merupakan suatu teknik untuk menemukan rute evakuasi yang optimal dalam berbagai skema pencarian rute. ACO terinspirasi dari perilaku mencari makan semut [11, 14], dimana semut meninggalkan jejak feromon yang memandu semut lain untuk menemukan rute terbaik.



Gambar 7 Skema pencarian jalur *Ant Colony Optimization*
(Source: Google)

Dalam skema tersebut, titik-titik awal mewakili lokasi awal evakuasi, sementara titik-titik akhir mewakili tujuan evakuasi. Semut-semut virtual akan menjelajahi jalur-jalur yang tersedia dan meninggalkan jejak feromon pada jalur yang lebih baik. Intensitas feromon pada jalur terpendek akan meningkat seiring waktu, sehingga semakin banyak semut akan memilih jalur tersebut, hingga ditemukan jalur evakuasi optimal.

2.1.5 Pengujian

Dalam penelitian ini, pengujian dilakukan menggunakan dua metode utama untuk mengevaluasi kinerja algoritma dalam optimasi jalur evakuasi, yaitu Pengujian Jalur Optimal dan Pengujian *Root Mean Square Error* (RMSE). Pengujian Jalur Optimal bertujuan untuk menilai seberapa efektif algoritma dalam menemukan jalur evakuasi terpendek atau tercepat berdasarkan parameter tertentu, seperti jarak, waktu tempuh, atau tingkat risiko. Evaluasi dilakukan dengan membandingkan jalur hasil algoritma terhadap jalur referensi atau kondisi ideal yang telah ditentukan sebelumnya. Sementara itu, pengujian RMSE digunakan untuk mengukur tingkat akurasi hasil prediksi algoritma terhadap data aktual. RMSE memberikan gambaran sejauh mana hasil optimasi menyimpang dari nilai yang seharusnya. Nilai RMSE yang lebih kecil menunjukkan bahwa algoritma mampu menghasilkan solusi yang mendekati nilai optimal, sehingga dapat diandalkan dalam skenario evakuasi darurat.

a) Pengujian Jalur Optimal

Pengujian ini bertujuan untuk menentukan algoritma mana yang menghasilkan jalur tercepat atau terpendek dari titik asal ke titik tujuan. Algoritma yang diuji meliputi *Breadth-First Search* (BFS), *Depth-First Search* (DFS), *Ant Colony Optimization* (ACO). Metode pengujian dilakukan dengan mengukur total jarak tempuh dari titik awal ke titik tujuan untuk setiap jalur evakuasi yang dihasilkan oleh masing-masing algoritma. Rumus yang digunakan untuk menghitung jalur terpendek adalah:

$$Total\ Jarak\ = \sum_{i=1}^{n} d(i, i+1) \quad (1)$$

Dimana:

- d(i,i+1) adalah jarak antara dua titik i dan $i + 1$,
- n adalah jumlah titik yang dilalui pada jalur evakuasi.

Setiap algoritma akan dibandingkan berdasarkan total jarak tempuh, dengan jalur terpendek ditentukan sebagai jalur dengan nilai Total Jarak Total Jarak terkecil.

b) Pengujian RMSE

Selain menghitung jalur terpendek, efisiensi setiap algoritma juga diuji untuk mengetahui seberapa efektif algoritma dalam menemukan jalur evakuasi yang optimal. Efisiensi dihitung berdasarkan perbandingan antara waktu eksekusi dari setiap algoritma dan hasil jalur yang dihasilkan. Rumus yang digunakan untuk menghitung efisiensi adalah sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2} \quad (2)$$

Dimana:

- yi : Nilai aktual (observasi yang benar)
- y῀i : Nilai yang diprediksi
- n : Jumlah total data

RMSE adalah akar dari nilai *Mean Squared Error* (MSE). RMSE memberikan nilai dalam satuan yang sama dengan data asli, sehingga lebih mudah untuk diinterpretasikan dibandingkan MSE. RMSE lebih sensitif terhadap error besar karena penggunaan kuadrat [16], sehingga lebih cocok digunakan untuk analisis eror.

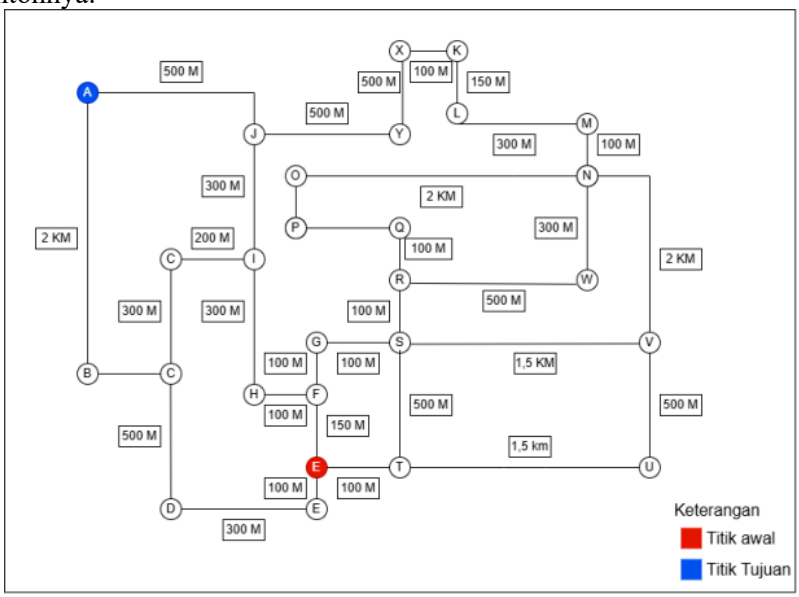
Dalam penelitian ini, nilai RMSE dihitung dengan membandingkan hasil jalur yang diperoleh dari algoritma (ACO, BFS, dan DFS) dengan jalur aktual yang diperoleh melalui Google Maps, baik dari segi jarak maupun waktu tempuh. Dengan demikian, RMSE memberikan gambaran mengenai tingkat kesalahan hasil algoritma terhadap kondisi jalur sebenarnya.

2. 2 Skenario Pengujian

Data yang digunakan dalam penelitian ini diperoleh dari peta jalan di Kabupaten Mamuju, yang diambil menggunakan Google Maps untuk mendapatkan detail jalur. Data ini kemudian dijadikan sebagai dataset untuk analisis. Penelitian ini menguji beberapa skenario yang direncanakan sebagai berikut:

1) Single Vertex Single Goals (SVSG)

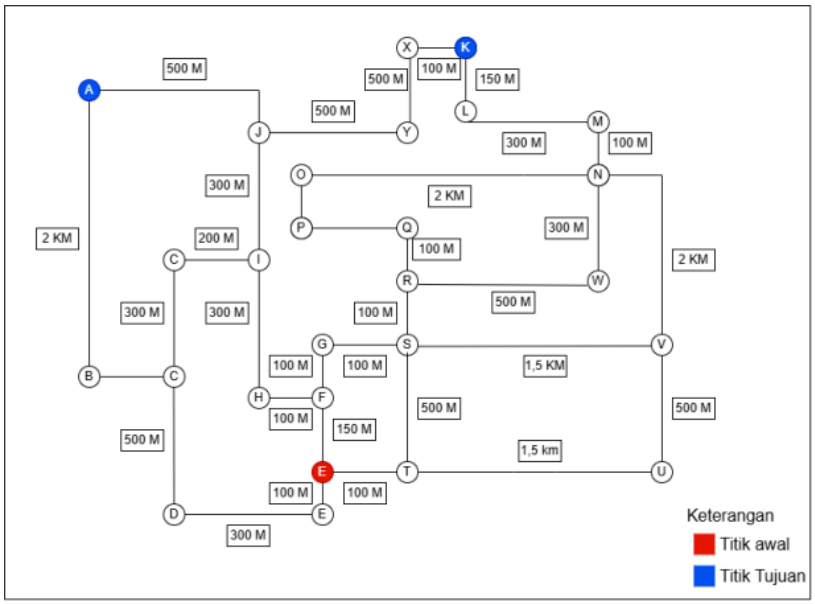
Skenario ini melibatkan satu titik awal dan satu titik tujuan. Fokusnya adalah menemukan jalur evakuasi paling efisien dari satu lokasi spesifik ke satu lokasi aman tertentu. Berikut adalah contohnya:



Gambar 8 Skenario SVSG.

2) Single Vertex Multi Goals (SVMG)

Skenario ini melibatkan satu titik awal dan beberapa titik tujuan. Tujuannya adalah untuk mengoptimalkan jalur evakuasi terdekat dari berbagai pilihan lokasi evakuasi yang ada.



Gambar 9 Skenario SVMG.

Kedua skenario ini menggambarkan kondisi nyata pada berbagai kemungkinan situasi darurat yang bisa terjadi selama proses evakuasi.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian

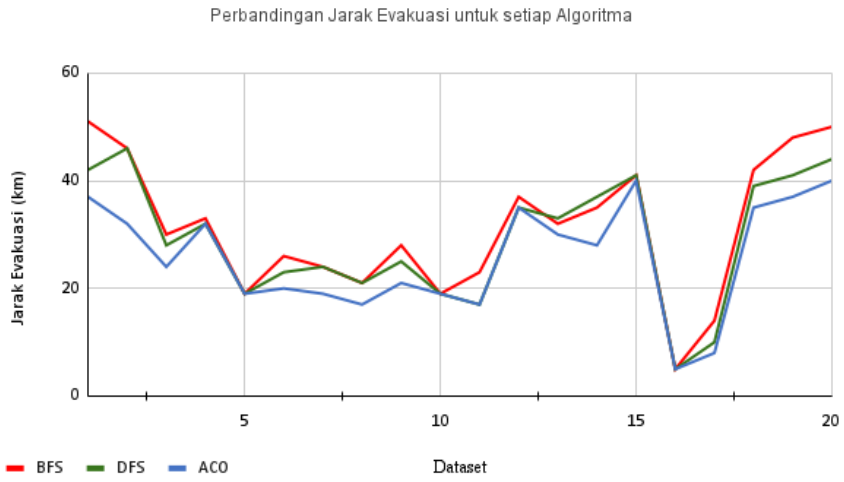
Penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma pencarian jalur evakuasi, yaitu *Breadth-First Search* (BFS), *Depth-First Search* (DFS), dan *Ant Colony Optimization* (ACO). Sebanyak 20 dataset digunakan dalam pengujian ini. Setiap algoritma diuji untuk menemukan jalur evakuasi paling efisien, dan hasilnya dibandingkan dengan jalur optimal (jalur aktual) yang diperoleh secara manual terhadap dataset yang digunakan. Hasil pengujian dari ketiga algoritma dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1 pengujian algoritma

Dataset	Jarak Evakuasi			Jalur Optimal (Km)
	BFS (Km)	DFS (Km)	ACO (Km)	
1	51	42	37	37
2	46	46	32	32
3	30	28	24	24
4	33	32	32	32
5	19	19	19	19
6	26	23	20	20
7	24	24	19	19

8	21	21	17	17
9	28	25	21	21
10	19	19	19	19
11	23	17	17	17
12	37	35	35	35
13	32	33	30	30
14	35	37	28	28
15	41	41	40	40
16	5	5	5	5
17	14	10	8	8
18	42	39	35	35
19	48	41	37	37
20	50	44	40	40

Pada Tabel 1, terlihat bahwa algoritma *Ant Colony Optimization* (ACO) secara konsisten menghasilkan jalur evakuasi yang identik dengan jalur aktual pada seluruh dataset. *Depth-First Search* (DFS) menghasilkan hasil yang cukup kompetitif, meskipun ada beberapa kasus di mana DFS memberikan jalur yang lebih panjang dibandingkan ACO. Sebaliknya, *Breadth-First Search* (BFS) sering kali menghasilkan jalur yang lebih panjang dibandingkan ACO dan DFS. Berdasarkan tabel tersebut maka diperoleh grafik perbandingan jarak evakuasi seperti pada Gambar 10.



Gambar 10. Perbandingan Jarak Evakuasi untuk Setiap Algoritma

Gambar 10 memperlihatkan perbandingan Jarak evakuasi yang dihasilkan oleh masing-masing algoritma pada setiap dataset. Dari grafik tersebut, terlihat bahwa *Ant Colony Optimization* (ACO) memiliki performa paling efisien dalam mengoptimalkan jarak evakuasi, menghasilkan jarak tempuh terendah di hampir semua dataset. Sebaliknya, *Breadth-First Search*

(BFS) secara konsisten menghasilkan jarak evakuasi tertinggi, menunjukkan bahwa algoritma ini kurang efisien dalam kondisi dinamis. *Depth-First Search* (DFS) berada di antara kedua algoritma ini, dengan performa lebih baik dibandingkan BFS, namun masih berada di bawah ACO dalam hal kecepatan waktu evakuasi.

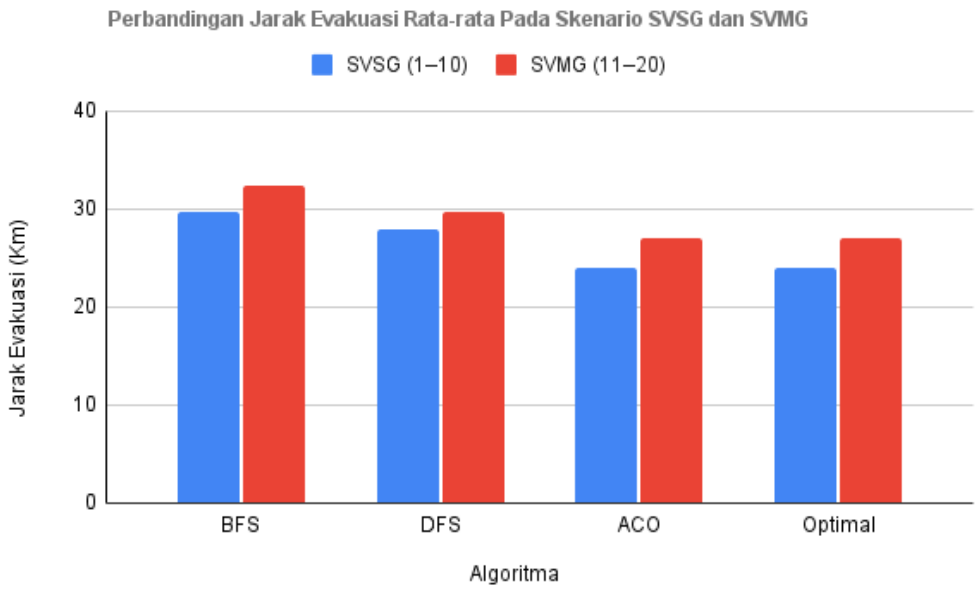
Selain analisis per dataset pada Tabel I dan Gambar 10, hasil pengujian juga dirangkum berdasarkan dua skenario yang digunakan dalam penelitian ini, yaitu *Single Vertex Single Goals* (SVSG) dan *Single Vertex Multi Goals* (SVMG).

Tabel 2 Rata-rata Jarak Evakuasi pada Skenario SVSG dan SVMG

Skenario	BFS (km)	DFS (km)	ACO (km)	Jalur Optimal (km)
SVSG (1–10)	29.7	27.9	24.0	24.0
SVMG (11–20)	32.3	29.7	27.0	27.0

Dari Tabel 2 terlihat bahwa pada kedua skenario, algoritma *Ant Colony Optimization* (ACO) secara konsisten menghasilkan jarak evakuasi paling dekat dan mendekati jalur optimal. Pada skenario SVSG (dataset 1–10), rata-rata jarak evakuasi ACO adalah 24 km, identik dengan jalur optimal. Sementara itu, algoritma DFS menghasilkan rata-rata jarak 27,9 km dan BFS sebesar 29,7 km, yang menunjukkan adanya selisih cukup jauh dari jalur optimal.

Pada skenario SVMG (dataset 11–20), ACO tetap memberikan hasil yang sangat dekat dengan jalur optimal (27 km), sedangkan DFS menghasilkan rata-rata jarak 29,7 km dan BFS 32,3 km. Hal ini menunjukkan bahwa meskipun jumlah titik awal lebih dari satu, ACO tetap mampu menyesuaikan pencarian jalur dengan baik Secara keseluruhan, hasil ini memperkuat temuan bahwa ACO lebih unggul dibandingkan BFS dan DFS, baik pada skenario sederhana *Single Vertex Single Goals*(SVSG) maupun skenario kompleks *Single Vertex Multi Goals* (SVMG). Berdasarkan tabel tersebut maka diperoleh grafik perbandingan jarak evakuasi seperti pada gambar 11.



Gambar 11. Perbandingan Jarak Evakuasi Rata-rata pada Skenario SVSG dan SVMG

Berdasarkan Gambar 11, algoritma ACO konsisten menghasilkan jarak evakuasi paling efisien pada kedua skenario. Pada SVSG, ACO identik dengan jalur optimal, sedangkan DFS dan BFS menunjukkan selisih jarak. Pada SVMG, ACO tetap mendekati jalur optimal, sementara DFS dan BFS menghasilkan jalur lebih panjang.

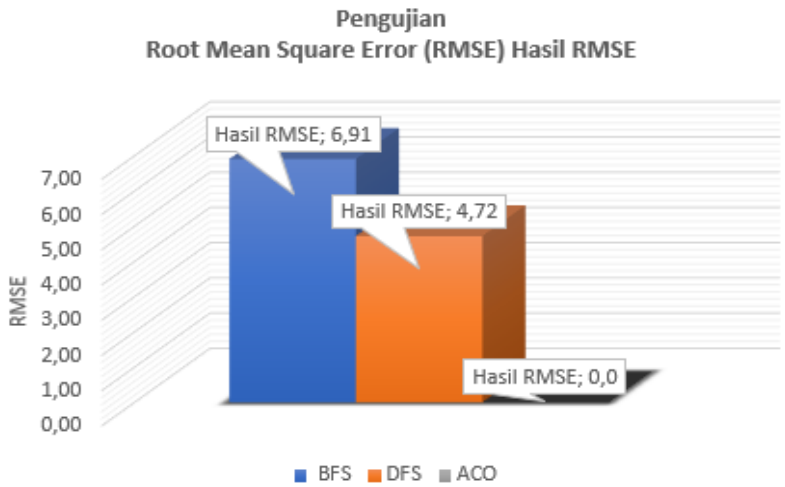
Kinerja BFS dan DFS sangat bergantung pada struktur graf dan distribusi jalur. BFS optimal saat graf dangkal dan tujuan berada di tingkat yang lebih tinggi, karena algoritma ini menjelajahi node-node terdekat terlebih dahulu. Namun, BFS menjadi kurang efisien pada graf besar atau dalam, karena banyaknya node yang tidak relevan dieksplorasi. Sementara itu, DFS cocok untuk graf dengan jalur tujuan yang jauh di kedalaman atau cabang yang sedikit, tetapi kurang efisien jika banyak cabang tidak relevan dieksplorasi lebih dulu. Sebagai solusi, ACO lebih unggul dalam kondisi dinamis karena kemampuannya untuk beradaptasi dengan cepat terhadap perubahan lingkungan dan menemukan jalur optimal berdasarkan prinsip feromon, menjadikannya sangat efektif untuk skenario evakuasi bencana

Selain pengujian jarak evakuasi, penelitian ini juga mengevaluasi efisiensi ketiga algoritma berdasarkan *Root Mean Square Error* (RMSE), yang merupakan indikator seberapa besar penyimpangan hasil algoritma dari jalur aktual. Tabel 3 menyajikan nilai RMSE untuk masing-masing algoritma pada setiap dataset.

Tabel 3 Pengujian RMSE

Algoritma	Hasil RMSE
BFS	6,91
DFS	4,72
ACO	0,0

Dari hasil pengujian RMSE tersebut, maka diperoleh hasil perbandingan antara 3 algoritma yang di ujikan seperti pada gambar 12.



Gambar 12. Grafik Perbandingan Nilai RMSE

Berdasarkan hasil pengujian, algoritma *Ant Colony Optimization* (ACO) terbukti paling efisien dalam menemukan jalur evakuasi optimal dengan RMSE sebesar 0.0, menunjukkan bahwa ACO secara konsisten memberikan jalur yang identik dengan jalur aktual tanpa penyimpangan. Algoritma *Depth-First Search* (DFS) mencatatkan RMSE sebesar 4.72, yang menunjukkan bahwa meskipun DFS lebih mendekati jalur optimal dibandingkan BFS, masih terdapat

penyimpangan yang signifikan pada beberapa dataset. Sementara itu, *Breadth-First Search* (BFS) memiliki RMSE tertinggi sebesar 6.91, menandakan bahwa algoritma ini sering kali menghasilkan jalur yang lebih panjang dan kurang efisien dibandingkan dua algoritma lainnya. Oleh karena itu, ACO adalah algoritma yang paling cocok untuk digunakan dalam skenario optimasi jalur evakuasi, sementara DFS bisa menjadi alternatif dalam kasus yang kurang kompleks, dan BFS lebih sesuai untuk situasi di mana pencarian menyeluruh lebih penting daripada efisiensi.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa algoritma *Ant Colony Optimization* (ACO) memiliki performa terbaik dalam menentukan jalur evakuasi bencana dibandingkan dengan *Breadth-First Search* (BFS) dan *Depth-First Search* (DFS). Pada kedua skenario pengujian, yaitu *Single Vertex Single Goals* (SVSG) dan *Single Vertex Multi Goals* (SVMG), ACO secara konsisten menghasilkan jalur yang identik dengan jalur optimal. Hal ini dibuktikan dengan nilai *Root Mean Square Error* (RMSE) sebesar 0,00, yang mengindikasikan tidak adanya penyimpangan terhadap jarak optimal. Sebaliknya, algoritma DFS memperoleh nilai RMSE sebesar 4,72 dan BFS sebesar 6,91, menunjukkan bahwa keduanya masih memiliki selisih jarak terhadap rute optimal, dengan BFS menjadi algoritma yang paling jauh dari solusi ideal.

Secara keseluruhan, ACO terbukti lebih adaptif dalam menghadapi struktur graf berbobot dan kompleksitas jaringan jalan, terutama karena mekanisme feromon dan pembaruan jalur yang memungkinkan eksplorasi dan eksploitasi solusi secara seimbang. DFS masih menunjukkan performa yang cukup baik pada jaringan yang tidak terlalu kompleks, sedangkan BFS, meskipun mampu mengeksplorasi seluruh simpul secara merata, kurang efisien dalam konteks graf berbobot karena tidak mempertimbangkan bobot jarak secara optimal. Dengan demikian, ACO direkomendasikan sebagai metode utama dalam penentuan jalur evakuasi bencana di Kabupaten Mamuju, khususnya pada kondisi lingkungan yang dinamis dan melibatkan banyak alternatif tujuan evakuasi.

Untuk penelitian selanjutnya, disarankan agar model diperluas dengan mempertimbangkan faktor selain jarak, seperti estimasi waktu tempuh, kepadatan lalu lintas, kapasitas jalan, hingga integrasi dengan data spasial dan sistem informasi geografis (GIS) secara real-time. Selain itu, kombinasi ACO dengan algoritma lain seperti Dijkstra, Genetic Algorithm, atau A* juga berpotensi meningkatkan efisiensi dan ketepatan hasil, sehingga sistem jalur evakuasi menjadi lebih realistis dan aplikatif di lapangan.

REFERENSI

[1] H. Rakuasa, "Pemetaan Kondisi Fisik Wilayah Sebagai Upaya dalam Mitigasi Bencana Tsunami di Kecamatan Moa Lakor, Kabupaten Maluku Barat Daya, Provinsi Maluku," GEOFORUM. Jurnal Geografi dan Pendidikan Geografi , vol. 2, no. 1, pp. 13-20, 2023.

[2] BPS Sulbar, "BAPPENAS Menyelenggarakan FGD Statistik Kebencanaan Sulawesi Barat," Badan Pusat Statistik Provinsi Sulawesi Barat, 27 Juni 2022. [Online]. Available: https://sulbar.bps.go.id/id/news/2022/06/27/118/bappenas-menyelenggarakan-fgd-statistik-kebencanaan-sulawesi-barat.html. [Accessed 13 September 2024].

[3] M. Basri, K. Kistan and N. Najman, "Relawan Aksi Peduli Gempa Mamuju - Majene Kecamatan Tappalang Sulawesi Barat Tahun 2021," Ahmar Metakarya: Jurnal Pengabdian Masyarakat, vol. 2, no. 2, p. 54–59, 2023.

[4] R. Budiarto, A. Permana, dan S. Nugroho, "Optimasi Jalur Evakuasi Banjir Menggunakan Algoritma *Breadth-First Search* (BFS)," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 235–242, 2020.

[5] D. Prasetyo dan A. Hidayat, "Implementasi Algoritma *Depth-First Search* dalam Penelusuran Jalur Evakuasi Bencana Gunung Meletus," Jurnal Ilmiah Teknologi Informasi Asia, vol. 14, no. 2, pp. 101–108, 2018.

[6] J. Sihotang, "Analysis Of Shortest Path Determination By Utilizing *Breadth-First Search* Algorithm," Jurnal Info Sains : Informatika dan Sains, vol. 10, no. 2, pp. 1-5, 2020.

[7] Y.-H. Chen and C.-M. Wu, "An Improved Algorithm for Searching Maze Based on *Depth-First Search* ," in 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), Taiwan, 2020.

[8] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, p. 269–271, 1959.

[9] W. Firgiawan, A. A. Cirua, M. Akbar and S. Cokrowibowo, "Perbandingan Waktu Komputasi Algoritma Greedy-Backtracking, BFS, DFS, dan Genetika pada Masalah Penukaran Koin," Proceeding KONIK (Konferensi Nasional Ilmu Komputer), pp. 5-8, 2021.

[10] L. Wu, X. Huang, J. Cui, C. Liu and W. Xiao, "Modified adaptive *Ant Colony Optimization* algorithm and its application for solving path planning of mobile robot," Expert Systems with Applications, vol. 215, p. 119410, 2023.

[11] E. Alhenawi, R. A. Khurma, A. A. Sharieh, O. Al-Adwan, A. A. Shorman and F. Shannaq, "Parallel *Ant Colony Optimization* Algorithm for Finding the Shortest Path for Mountain Climbing," IEEE Access , vol. 11, pp. 6185 - 6196, 2023.

[12] H. Sun, K. Zhu, W. Zhang, Z. Ke, H. Hu, K. Wu, T. Zhang, "Emergency path planning based on improved *Ant Colony Optimization* algorithm," Journal of Building Engineering, vol. 100, 2025.

[13] X. Diao, "Route optimization for hazardous materials transportation vehicles based on BFA-ACO under carbon tax policy. *Journal of Computational Methods in Sciences and Engineering*," Journal of Combinatorial Mathematics & Combinatorial Computing, vol. 24, no. 3, p. 1943-1954, 2024.

[14] K. Dong, D. Yang, J. Sheng, W. Zhang and P. Jing, "Dynamic planning method of evacuation route in dam-break flood scenario based on the ACO-GA hybrid algorithm," International Journal of Disaster Risk Reduction, vol. 104219, 2024.

[15] R. Scheffler, "On the recognition of search trees generated by BFS and DFS," Theoretical Computer Science, vol. 936, pp. 116-128, 2022.

[16] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the *Root Mean Square Error* (RMSE) in Assessing Rounding Model," IOP Conference Series: Materials Science and Engineering, vol. 324 , 2018.

[17] D. Rachmawati, P. Sihombing and B. Halim, "Implementation of Best First Search Algorithm in Determining Best Route Based on Traffic Jam Level in Medan City," in 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), Medan, 2018.

[18] L. G. Iriani, Determining tsunami evacuation building location and evacuation routes based on population dynamic and human behaviour in disaster evacuation in Pacitan sub-district area, Universitas Gadjah Mada: Doctoral dissertation, 2017.

[19] L. E. Zen and D. U. Iswavigra, "Penggunaan Algoritma *Depth-First Search* Dalam Sistem Pakar: Studi Literatur Article Sidebar," Jurnal Informasi dan Teknologi, vol. 5, no. 2, pp. 95-90, 2023.

[20] R. Jain, "A Comparative Study of *Breadth-First Search* and *Depth-First Search* Algorithms in Solving the Water Jug Problem on Google Colab," Advances in AI, vol. 1, no. 1, 2023.