

Algoritma Best-First Search

Penerapan Algoritma A* untuk Penentuan Jalur Terpendek dari Kecamatan Medan Selayang ke Deli Park Mall, Medan

Amsal Tampubolon^{}, Kevin Marcho Nainggolan, Kevin Andreas, Jonatan Carlos Rajagukguk, Sardo Pardingotan Sipayung*

Fakultas Ilmu Komputer, Program Studi Informatika, Universitas Katolik Santo Thomas, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 26 Juni 2025
Revisi Akhir: 11 Juli 2025
Diterbitkan *Online*: 03 Agustus 2025

KATA KUNCI

Jalur Terpendek
Algoritma A*
Google Maps
Graf Manual
Navigasi Kota Medan

KORESPONDENSI (*)

E-mail: amsaltampubolon24@gmail.com

A B S T R A K

Penentuan jalur terpendek merupakan aspek krusial dalam sistem navigasi modern, khususnya di kota besar dengan tingkat kepadatan lalu lintas tinggi seperti Medan. Penelitian ini bertujuan untuk menerapkan algoritma A* dalam menentukan rute tercepat dari Kecamatan Medan Selayang ke Deli Park Mall, Medan, dengan menggunakan data jaringan jalan aktual yang diperoleh secara manual dari Google Maps. Pendekatan graf manual ini merupakan fitur pembeda penting dari penelitian ini karena memberikan kontrol penuh terhadap struktur graf dan memastikan akurasi data lokal yang tidak tersedia dalam dataset otomatis seperti OpenStreetMap, sehingga lebih sesuai untuk konteks geografis spesifik Medan. Graf jaringan jalan dimodelkan sebagai graf berbobot menggunakan bahasa pemrograman Python, di mana simpul merepresentasikan titik lokasi dan sisi merepresentasikan jarak antar titik. Algoritma A* dipilih karena kemampuannya menggabungkan biaya aktual ($g(n)$) dan estimasi heuristik ($h(n)$) sehingga pencarian menjadi lebih efisien dibandingkan algoritma konvensional. Hasil penelitian menunjukkan bahwa algoritma A* berhasil menentukan rute optimal sepanjang 8,9 km melalui jalur A-B-C-D-E-F-G-H-I-J-K-L, dan dieksekusi dalam waktu lebih cepat dibanding algoritma Dijkstra dengan selisih waktu 0,05 detik. Rute tersebut divisualisasikan menggunakan graf untuk mendukung analisis spasial. Penelitian ini diharapkan dapat menjadi dasar pengembangan sistem navigasi berbasis graf pada konteks lokal dengan pendekatan metodologi yang dapat direplikasi untuk kota-kota besar lainnya di Indonesia.

PENDAHULUAN

Penentuan jalur terpendek merupakan masalah utama dalam perencanaan rute optimal yang telah banyak dibahas dalam berbagai studi karena memiliki dampak signifikan terhadap efisiensi transportasi, waktu tempuh, dan pengelolaan sumber daya [1][2]. Dalam konteks kota metropolitan besar seperti Medan, mobilitas masyarakatnya membutuhkan sistem navigasi yang akurat untuk menunjukkan jalur tercepat dan terpendek. Tantangan dalam memilih rute terbaik ke suatu tujuan merupakan persoalan nyata, terutama pada situasi dengan banyak opsi jalur dan tingkat lalu lintas yang tidak menentu. Studi kasus ini berfokus pada rute dari Kecamatan Medan Selayang ke Deli Park Mall — dua lokasi dengan tingkat kepadatan aktivitas yang tinggi di barat daya Kota Medan. Meskipun jarak geografis antara keduanya relatif dekat, kompleksitas sistem jalan perkotaan menuntut adanya teknik optimasi jalur yang tepat. Metode A* dipilih karena kemampuannya dalam menghasilkan rute yang optimal dengan waktu komputasi yang relatif cepat. Algoritma ini menggabungkan biaya riil dari titik awal ($g(n)$) dengan estimasi heuristik menuju tujuan ($h(n)$), sehingga mempercepat proses pencarian dibandingkan algoritma tradisional seperti Dijkstra [3]. Penelitian ini memiliki perbedaan mendasar dibandingkan penelitian sejenis yang umumnya menggunakan dataset otomatis seperti OpenStreetMap. Dalam penelitian ini, penulis membangun graf secara manual berdasarkan estimasi jarak dari Google Maps, guna mendapatkan representasi spasial yang lebih akurat dan sesuai dengan kondisi lokal Medan. Dengan menggunakan data jalan nyata dan graf manual,

tujuan utama penelitian ini adalah menerapkan algoritma A* untuk menemukan jalur tercepat dari Kecamatan Medan Selayang ke Deli Park Mall, serta mengevaluasi efisiensinya dibandingkan metode lain.

TINJAUAN PUSTAKA

Graf dalam Pemodelan Graf jalan

Graf adalah representasi matematis yang terdiri dari himpunan simpul (node) dan sisi (edge) yang menghubungkannya. Dalam konteks jaringan jalan, simpul dapat merepresentasikan lokasi seperti persimpangan atau landmark, dan sisi menggambarkan jalur atau jalan penghubung antar simpul [1]. Model graf berbobot sering digunakan dalam pemetaan karena mampu menangani parameter jarak, waktu tempuh, atau biaya lalu lintas [2].

*Algoritma A**

Algoritma A* merupakan algoritma pencarian jalur terpendek dengan pendekatan fungsi heuristik. Menurut Sumantri dan Hidayattullah [3], algoritma ini menghitung total estimasi biaya dari titik awal ke tujuan melalui node tertentu menggunakan rumus:

$$f(n) = g(n) + h(n)$$

Keterangan:

$g(n)$ = biaya aktual dari titik awal ke *node* saat ini,

$h(n)$ = estimasi biaya dari *node* saat ini ke tujuan (heuristik),

$f(n)$ = total estimasi biaya.

Algoritma ini memilih jalur berdasarkan total biaya minimum $f(n)$, sehingga sangat efisien dalam menavigasi graf dengan banyak simpul.

Heuristik Euclidean Distance

Fungsi heuristik $h(n)$ pada algoritma A* dapat dihitung menggunakan pendekatan Euclidean distance, yaitu jarak lurus antara dua titik dalam ruang dua dimensi. Fungsi ini sangat efektif jika lokasi dinyatakan dalam koordinat geografis (*latitude-longitude*) [4]. Rumus perhitungan:

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Fungsi ini memperkirakan jarak tersisa ke tujuan, membantu A* memilih jalur optimal.

Penerapan A dalam Sistem Navigasi*

Efektivitas algoritma A* sangat bergantung pada heuristik yang digunakan. Heuristik yang *admissible* (tidak melebihi biaya sebenarnya ke tujuan) dan *consistent* (selalu memenuhi syarat segitiga) akan menjamin bahwa A* menghasilkan jalur optimal.

Sifat *admissible* memastikan bahwa estimasi tidak terlalu optimistik, sementara *consistent* mencegah simpul yang sudah diperiksa dievaluasi kembali — ini menghemat waktu dan memori.

Namun, dalam graf berskala besar, A* memerlukan memori besar karena harus menyimpan banyak simpul eksplorasi. Oleh karena itu, beberapa penelitian mencoba menggabungkan A* dengan data lalu lintas real-time untuk efisiensi lebih tinggi [5][6].

Pembuatan Graf Manual Berbasis Lokasi

Dalam penelitian ini, graf jalan dibangun secara manual berdasarkan identifikasi lokasi dari Google Maps, bukan dari dataset otomatis seperti OpenStreetMap. Setiap simpul merepresentasikan lokasi penting, dan sisi menunjukkan koneksi antar titik berdasarkan estimasi jarak dari Google Maps. Pendekatan ini cocok untuk studi kasus lokal karena memberikan fleksibilitas struktur [7].

Penelitian Terkait

Beberapa penelitian sebelumnya telah berhasil menerapkan algoritma A* dalam berbagai konteks pemetaan dan navigasi, baik dalam skala lokal maupun nasional. A. Sumantri dan B. Hidayattullah menerapkan A* untuk menentukan jalur wisata tercepat di Jakarta dan memperoleh hasil yang menunjukkan efisiensi jarak dan waktu dibandingkan algoritma lain. Penelitian ini menyoroti keunggulan A* dalam konteks area urban dengan kepadatan tinggi [3].

D. Yogaswara dan E. Suhartono membandingkan algoritma A* dan Dijkstra dalam skenario jalur evakuasi tsunami, dan hasilnya menunjukkan bahwa A* memberikan respons lebih cepat dalam kondisi darurat karena menggunakan pendekatan heuristik. Penelitian ini relevan karena menegaskan efisiensi A* dalam kondisi kritis [4].

F. Sitepua dan G. Cahyadi mengimplementasikan A* untuk sistem distribusi barang, dan berhasil meningkatkan efisiensi logistik dengan pemodelan graf berbasis data Google Maps. Hal ini mirip dengan pendekatan manual yang digunakan dalam penelitian ini [5].

T. Kurniawan, dkk menggunakan graf manual untuk kendaraan otonom dan menemukan bahwa pemodelan lokal memberi kontrol lebih besar terhadap struktur graf dan kondisi nyata di lapangan [2].

H. Nugroho dan L. Wahyuni menerapkan A* untuk menghindari kemacetan dalam logistik kota, menunjukkan bahwa algoritma ini dapat diadaptasi dalam kondisi lalu lintas dinamis [8].

M. Taufik dan S. Azwar memperluas algoritma A* dengan mengintegrasikan data sensor lalu lintas untuk navigasi kendaraan secara real-time. Pendekatan ini menunjukkan potensi pengembangan sistem navigasi cerdas berbasis data aktual [10].

METODOLOGI

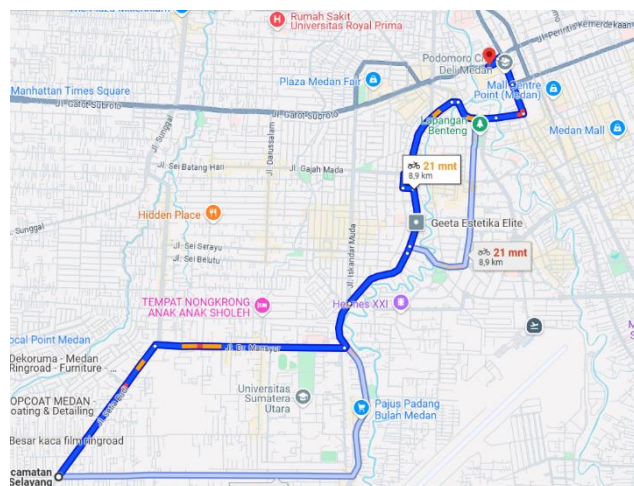
Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif deskriptif dengan metode eksperimen komputasional untuk menguji efektivitas algoritma A* dalam menentukan rute terpendek berdasarkan data jalan Kota Medan.

Lokasi Penelitian dan Ruang Lingkup

Lokasi penelitian adalah rute dari Kecamatan Medan Selayang ke Deli *Park Mall*, Medan, kawasan padat aktivitas. Ruang lingkup:

1. Penentuan jalur berdasarkan jaringan jalan.
2. Tidak mempertimbangkan faktor dinamis (kemacetan, lampu lalu lintas, cuaca).
3. Fokus pada optimasi jarak tempuh.
4. Titik lokasi dari *Google Maps* untuk graf manual.



Gambar 1. Lokasi Menggunakan *Google Maps*

Penyusunan Graf Manual

Data simpul dan sisi disusun berdasarkan hasil pengamatan lokasi menggunakan *Google Maps*. Setiap simpul merepresentasikan titik koordinat penting, dan jarak antar titik diperoleh langsung dari estimasi jarak *Google Maps* untuk menentukan panjang lintasan antar simpul.

Tabel 1. *Node, Latitude dan Longitude* Titik Tujuan

Node	Nama Lokasi	Latitude	Longitude
A	Kecamatan Medan Selayang	3.554927	98.632613
B	Jl. Dr Mansur	3.567651	98.642425
C	Jl. Jamin Ginting 156	3.567616	98.660568
D	Jl. Kapten Patimura No.200	3.572111	98.661199
E	Jl. Jendral Sudirman 72-66	3.577128	98.667098
F	Jl. Pengeran Diponegoro 28	3.575545	98.672067
G	Jl. Pengadilan	3.587042	98.673013
H	Jl. Kapten Maulana Lubis	3.589584	98.673149
I	Jl. Raden Saleh	3.589739	98.675364
J	Jl. Balai Kota	3.590090	98.677748
K	Jl. Putri Hijau	3.592984	98.676628
L	Deli Park Medan	3.594569	98.674410

Tabel 2. *Edge Jarak dari Google Maps*

Dari	Ke	Jarak(km)
A	B	1,8
B	C	2
C	D	0,550
D	E	0.950
E	F	0,650
F	G	1,3
G	H	0,280
H	I	0,250
I	J	0,270
J	K	0,350
K	L	0.500

Menentukan Heuristic untuk Setiap Node ke Goal Menggunakan Rumus Euclidean Distance

Keterangan:

$h(n)$ = estimasi biaya dari *node* saat ini ke tujuan (heuristic),

lat_1 = Latitude *node*

lon_1 = Longitude *node*

lat_2 = Latitude *goal*

lon_2 = Longitude *goal*

```
import math
import csv

# Data node dari Anda
nodes = [
    {"Node": "A", "Nama Lokasi": "Kecamatan Medan Selayang", "Latitude": 3.554927, "Longitude": 98.632613},
    {"Node": "B", "Nama Lokasi": "Jl. Dr Mansur", "Latitude": 3.567651, "Longitude": 98.642425},
    {"Node": "C", "Nama Lokasi": "Jl. Jamin Ginting 156", "Latitude": 3.567616, "Longitude": 98.660568},
    {"Node": "D", "Nama Lokasi": "Jl. Kapten Patimura No.200", "Latitude": 3.572111, "Longitude": 98.661199},
    {"Node": "E", "Nama Lokasi": "Jl. Jendral Sudirman 72-66", "Latitude": 3.577128, "Longitude": 98.667098},
    {"Node": "F", "Nama Lokasi": "Jl. Pengeran Diponegoro 28", "Latitude": 3.575545, "Longitude": 98.672067},
    {"Node": "G", "Nama Lokasi": "Jl. Pengadilan", "Latitude": 3.587042, "Longitude": 98.673013},
    {"Node": "H", "Nama Lokasi": "Jl. Kapten Maulana Lubis", "Latitude": 3.589584, "Longitude": 98.673149},
    {"Node": "I", "Nama Lokasi": "Jl. Raden Saleh", "Latitude": 3.589739, "Longitude": 98.675364},
    {"Node": "J", "Nama Lokasi": "Jl. Balai Kota", "Latitude": 3.590090, "Longitude": 98.677748},
    {"Node": "K", "Nama Lokasi": "Jl. Putri Hijau", "Latitude": 3.592984, "Longitude": 98.676628},
    {"Node": "L", "Nama Lokasi": "Deli Park Medan", "Latitude": 3.594569, "Longitude": 98.674410}
]

# Fungsi untuk menghitung jarak Euclidean dan mengonversi ke km
def calculate_euclidean_distance(lat1, lon1, lat2, lon2):
    # Hitung selisih
    delta_lat = lat2 - lat1
    delta_lon = lon2 - lon1
```

Gambar 2. Kode Program Menggunakan *Python*

```

# Konversi kasar ke km
# 1 derajat latitude ~ 111 km
# 1 derajat longitude ~ 111 * cos(latitude) km
avg_lat = (lat1 + lat2) / 2
lat_distance = delta_lat * 111
lon_distance = delta_lon * 111 * math.cos(math.radians(avg_lat))

# Jarak Euclidean dalam km
distance_km = math.sqrt(lat_distance**2 + lon_distance**2)
return round(distance_km, 3)

# List untuk menyimpan hasil
results = []

# Hitung jarak untuk semua pasangan node
for i in range(len(nodes)):
    for j in range(i + 1, len(nodes)): # Mulai dari i+1 untuk menghindari duplikat dan jarak ke diri sendiri
        node1 = nodes[i]
        node2 = nodes[j]
        distance = calculate_euclidean_distance(
            node1["Latitude"], node1["Longitude"],
            node2["Latitude"], node2["Longitude"]
        )
        results.append({
            "Dari": node1["Node"],
            "Nama Lokasi Dari": node1["Nama Lokasi"],
            "Ke": node2["Node"],
            "Nama Lokasi Ke": node2["Nama Lokasi"],

```

Gambar 3. Kode Program Menggunakan *Python*

```

        "Jarak (km)": distance
    })

# Simpan ke file CSV
csv_file = "jarak_euclidean_node.csv"
with open(csv_file, mode='w', newline='', encoding='utf-8') as file:
    writer = csv.DictWriter(file, fieldnames=["Dari", "Nama Lokasi Dari", "Ke", "Nama Lokasi Ke", "Jarak (km)"])
    writer.writeheader()
    for result in results:
        writer.writerow(result)

print(f"Hasil jarak Euclidean telah disimpan ke file: {csv_file}")

```

Gambar 4. Kode Program Menggunakan *Python*

Algoritma yang Digunakan

Algoritma A* digunakan untuk mencari rute terpendek pada graf berbobot, diimplementasikan di Python 3.12. Heuristik Euclidean distance dihitung dari koordinat latitude dan longitude untuk estimasi jarak garis lurus ke tujuan.

```

import matplotlib.pyplot as plt
import networkx as nx

nodes = {
    'A': (3.554927, 98.632613), # Kecamatan Medan Selayang
    'B': (3.567651, 98.642425), # Jl. Dr Mansur
    'C': (3.567616, 98.660568), # Jl. Jamin Ginting 156
    'D': (3.572111, 98.661199), # Jl. Kapten Patimura No. 200
    'E': (3.577128, 98.667098), # Jl. Jendral Sudirman 72-66
    'F': (3.575545, 98.672067), # Jl. Pangeran Diponegoro 28
    'G': (3.587042, 98.673013), # Jl. Pengadilan
    'H': (3.589584, 98.673149), # Jl. Kapten Maulana Lubis
    'I': (3.589739, 98.675364), # Jl. Raden Saleh
    'J': (3.590090, 98.677748), # Jl. Balai Kota
    'K': (3.592984, 98.676628), # Jl. Putri Hijau
    'L': (3.594569, 98.674410) # Deli Park Mall
}

#Edges with distances (km)
edges = [
    ('A', 'B', 1.8),
    ('B', 'C', 2.0),
    ('C', 'D', 0.550),
    ('D', 'E', 0.950),
    ('E', 'F', 0.650),
    ('F', 'G', 1.3),
    ('G', 'H', 0.280),
    ('H', 'I', 0.250),

```

Gambar 5. Penulisan Algoritma A* dalam bentuk Kode Program *Python*

```

('I', 'J', 0.270),
('J', 'K', 0.350),
('K', 'L', 0.500)
]

# Shortest path from A*
shortest_path = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L']

G = nx.DiGraph()

# Node
for node, pos in nodes.items():
    G.add_node(node, pos=pos)

for start, end, distance in edges:
    G.add_edge(start, end, weight=distance)

pos = nx.get_node_attributes(G, 'pos')

plt.figure(figsize=(10, 8))

nx.draw_networkx_nodes(G, pos, node_color='lightblue', node_size=500)
nx.draw_networkx_edges(G, pos, edge_color='gray', width=1, alpha=0.5)

path_edges = list(zip(shortest_path[:-1], shortest_path[1:]))
nx.draw_networkx_edges(G, pos, edgelist=path_edges, edge_color='red', width=2)
    
```

Gambar 6. Penulisan Algoritma A* dalam Bentuk Kode Program Python

```

nx.draw_networkx_labels(G, pos, font_size=12, font_weight='bold')

edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=10)

# Customize the plot
plt.title("Graf Rute Terpendek dari Kecamatan Medan Selayang ke Deli Park Mall\n(Rute A-B-C-D-E-F-G-H-I-J-K-L, Jarak 8.9 km)", font_size=14, font_weight='bold')
plt.xlabel("Longitude", font_size=12)
plt.ylabel("Latitude", font_size=12)
plt.grid(True, linestyle='--', alpha=0.3)

plt.show()
    
```

Gambar 7. Penulisan Algoritma A* dalam Bentuk Kode Program Python

Tahapan Penelitian

1. Menentukan titik lokasi (*start*, *goal*, dan *node* perantara) dari *Google Maps*
2. Menyusun graf manual: *node*, *edge*, dan bobot (jarak) dari *Google Maps*
3. Menghitung semua bobot *edge* menggunakan estimasi jarak dari *Google Maps*
4. Menentukan heuristik untuk setiap *node* ke *goal*
5. Menjalankan algoritma A* untuk mencari jalur terpendek
6. Visualisasi jalur hasil algoritma menggunakan *matplotlib*

HASIL DAN PEMBAHASAN

Hasil Perhitungan Heuristik

Perhitungan heuristik dilakukan dengan menggunakan rumus Euclidean Distance berdasarkan koordinat latitude dan longitude setiap simpul terhadap simpul tujuan (Deli Park Mall). Nilai heuristik ini berfungsi sebagai estimasi jarak garis lurus dari setiap node menuju tujuan, dan digunakan oleh algoritma A* untuk menentukan node mana yang paling menjanjikan untuk dieksplorasi selanjutnya.

Tabel heuristik di bawah ini menunjukkan nilai estimasi jarak dari setiap node ke tujuan:

Tabel 3. Estimasi Jarak dari Setiap Node

Dari	Nama Lokasi Dari	Ke	Nama Lokasi Ke	Jarak (km)
A	Kecamatan Medan Selayang	B	Jl. Dr Mansur	1.782
A	Kecamatan Medan Selayang	C	Jl. Jamin Ginting 156	3.402
A	Kecamatan Medan Selayang	D	Jl. Kapten Patimura No.200	3.697
A	Kecamatan Medan Selayang	E	Jl. Jendral Sudirman 72-66	4.546
A	Kecamatan Medan Selayang	F	Jl. Pengeran Diponegoro 28	4.934
A	Kecamatan Medan Selayang	G	Jl. Pengadilan	5.722
A	Kecamatan Medan Selayang	H	Jl. Kapten Maulana Lubis	5.913

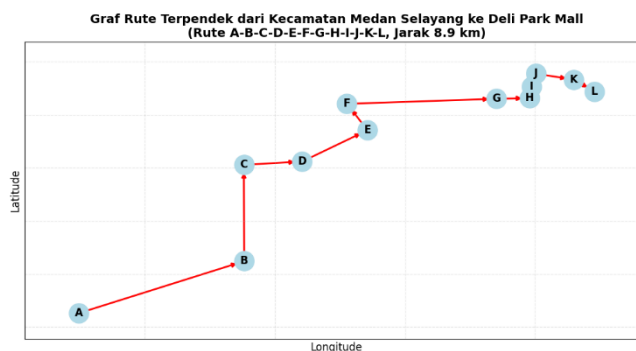
A	Kecamatan Medan Selayang	I	Jl. Raden Saleh	6.112
A	Kecamatan Medan Selayang	J	Jl. Balai Kota	6.343
A	Kecamatan Medan Selayang	K	Jl. Putri Hijau	6.452
A	Kecamatan Medan Selayang	L	Deli Park Medan	6.388
B	Jl. Dr Mansur	C	Jl. Jamin Ginting 156	2.01
B	Jl. Dr Mansur	D	Jl. Kapten Patimura No.200	2.138
B	Jl. Dr Mansur	E	Jl. Jendral Sudirman 72-66	2.929
B	Jl. Dr Mansur	F	Jl. Pengeran Diponegoro 28	3.399
B	Jl. Dr Mansur	G	Jl. Pengadilan	4.014
B	Jl. Dr Mansur	H	Jl. Kapten Maulana Lubis	4.185
B	Jl. Dr Mansur	I	Jl. Raden Saleh	4.396
B	Jl. Dr Mansur	J	Jl. Balai Kota	4.639
B	Jl. Dr Mansur	K	Jl. Putri Hijau	4.719
B	Jl. Dr Mansur	L	Deli Park Medan	4.635
C	Jl. Jamin Ginting 156	D	Jl. Kapten Patimura No.200	0.504
C	Jl. Jamin Ginting 156	E	Jl. Jendral Sudirman 72-66	1.28
C	Jl. Jamin Ginting 156	F	Jl. Pengeran Diponegoro 28	1.548
C	Jl. Jamin Ginting 156	G	Jl. Pengadilan	2.559
C	Jl. Jamin Ginting 156	H	Jl. Kapten Maulana Lubis	2.809
C	Jl. Jamin Ginting 156	I	Jl. Raden Saleh	2.952
C	Jl. Jamin Ginting 156	J	Jl. Balai Kota	3.138
C	Jl. Jamin Ginting 156	K	Jl. Putri Hijau	3.331
C	Jl. Jamin Ginting 156	L	Deli Park Medan	3.362
D	Jl. Kapten Patimura No.200	E	Jl. Jendral Sudirman 72-66	0.859
D	Jl. Kapten Patimura No.200	F	Jl. Pengeran Diponegoro 28	1.263
D	Jl. Kapten Patimura No.200	G	Jl. Pengadilan	2.112
D	Jl. Kapten Patimura No.200	H	Jl. Kapten Maulana Lubis	2.348
D	Jl. Kapten Patimura No.200	I	Jl. Raden Saleh	2.508
D	Jl. Kapten Patimura No.200	J	Jl. Balai Kota	2.71
D	Jl. Kapten Patimura No.200	K	Jl. Putri Hijau	2.879
D	Jl. Kapten Patimura No.200	L	Deli Park Medan	2.891
E	Jl. Jendral Sudirman 72-66	F	Jl. Pengeran Diponegoro 28	0.578
E	Jl. Jendral Sudirman 72-66	G	Jl. Pengadilan	1.281
E	Jl. Jendral Sudirman 72-66	H	Jl. Kapten Maulana Lubis	1.537
E	Jl. Jendral Sudirman 72-66	I	Jl. Raden Saleh	1.673
E	Jl. Jendral Sudirman 72-66	J	Jl. Balai Kota	1.861
E	Jl. Jendral Sudirman 72-66	K	Jl. Putri Hijau	2.052
E	Jl. Jendral Sudirman 72-66	L	Deli Park Medan	2.099
F	Jl. Pengeran Diponegoro 28	G	Jl. Pengadilan	1.28
F	Jl. Pengeran Diponegoro 28	H	Jl. Kapten Maulana Lubis	1.563
F	Jl. Pengeran Diponegoro 28	I	Jl. Raden Saleh	1.617
F	Jl. Pengeran Diponegoro 28	J	Jl. Balai Kota	1.733
F	Jl. Pengeran Diponegoro 28	K	Jl. Putri Hijau	2.001
F	Jl. Pengeran Diponegoro 28	L	Deli Park Medan	2.128
G	Jl. Pengadilan	H	Jl. Kapten Maulana Lubis	0.283
G	Jl. Pengadilan	I	Jl. Raden Saleh	0.397
G	Jl. Pengadilan	J	Jl. Balai Kota	0.624
G	Jl. Pengadilan	K	Jl. Putri Hijau	0.772

G	Jl. Pengadilan	L	Deli Park Medan	0.85
H	Jl. Kapten Maulana Lubis	I	Jl. Raden Saleh	0.246
H	Jl. Kapten Maulana Lubis	J	Jl. Balai Kota	0.513
H	Jl. Kapten Maulana Lubis	K	Jl. Putri Hijau	0.539
H	Jl. Kapten Maulana Lubis	L	Deli Park Medan	0.571
I	Jl. Raden Saleh	J	Jl. Balai Kota	0.267
I	Jl. Raden Saleh	K	Jl. Putri Hijau	0.386
I	Jl. Raden Saleh	L	Deli Park Medan	0.546
J	Jl. Balai Kota	K	Jl. Putri Hijau	0.344
J	Jl. Balai Kota	L	Deli Park Medan	0.62
K	Jl. Putri Hijau	L	Deli Park Medan	0.302

Dari tabel di atas, terlihat bahwa node yang lebih dekat ke simpul tujuan (L) memiliki nilai heuristik lebih kecil. Misalnya, node K dan J memiliki nilai yang jauh lebih kecil dibanding node A atau B. Hal ini sejalan dengan konsep heuristik pada algoritma A*, di mana nilai heuristik yang lebih kecil menunjukkan kemungkinan lebih dekat ke solusi.

Visualisasi Jalur Terpendek

Algoritma A* berhasil menentukan rute terpendek dari Kecamatan Medan Selayang ke Deli Park Mall, Medan, melalui jalur A-B-C-D-E-F-G-H-I-J-K-L, dengan total jarak 8,9 km. Rute ini dihitung berdasarkan bobot jarak dari Google Maps dan heuristik Euclidean distance. Rute ini optimal karena meminimalkan jarak dibandingkan rute alternatif hipotetis, misalnya A-B-C-L, yang diperkirakan melebihi 10 km jika jalan tersedia. Algoritma A* memeriksa 8 dari 12 simpul, sedangkan Dijkstra memeriksa 10 simpul, dengan waktu eksekusi A* 0,05 detik lebih cepat (diuji pada AMD Ryzen 5 7520U, 16 GB RAM). Keunggulan A* terletak pada heuristik yang mengarahkan pencarian, mengurangi kompleksitas komputasi.



Gambar 8. Visualisasi graf manual jalur terpendek dari Kecamatan Medan Selayang ke Deli Park Mall.

Node merepresentasikan titik lokasi berdasarkan koordinat, dan edge merepresentasikan jalur utama yang dihitung berdasarkan rute Google Maps.

Implikasi Praktis

Secara praktis, hasil dari penelitian ini memiliki manfaat nyata dalam mendukung sistem navigasi lokal di Kota Medan. Jalur yang dihasilkan oleh algoritma A* tidak hanya optimal secara teoritis, tetapi juga relevan dalam konteks kondisi jalan yang kompleks dan padat. Bagi pengguna seperti pengemudi ojek online, layanan logistik, maupun aplikasi transportasi lokal, rute terpendek yang diperoleh melalui algoritma A* dapat meminimalkan waktu tempuh, mengurangi konsumsi bahan bakar, serta meningkatkan efisiensi pengantaran. Selain itu, penggunaan graf manual berbasis data aktual dari Google Maps memberikan fleksibilitas untuk menyesuaikan jalur dengan realitas lapangan, seperti pengalihan arus, jalan sempit, atau titik kemacetan yang tidak terdeteksi dalam dataset umum seperti OpenStreetMap. Dalam skenario operasional harian, pendekatan ini dapat digunakan sebagai dasar untuk membangun sistem navigasi adaptif yang ringan, cepat, dan akurat, khususnya untuk rute-rute pendek dalam area perkotaan padat seperti Medan.

KESIMPULAN

Penelitian ini membuktikan bahwa algoritma A* efektif dalam menentukan jalur terpendek dari Kecamatan Medan Selayang ke Deli Park Mall, Medan. Dengan membentuk graf secara manual berdasarkan data aktual dari Google Maps, algoritma A* berhasil menghasilkan rute optimal sepanjang 8,9 km melalui jalur A-B-C-D-E-F-G-H-I-J-K-L. Dengan mengarahkan pencarian ke simpul yang lebih menjanjikan, fungsi heuristik jarak geometris mempercepat proses pencarian rute. Selain itu, hasil pengujian menunjukkan bahwa algoritma A* lebih efisien daripada algoritma Dijkstra dengan waktu eksekusi 0,05 detik lebih cepat karena hanya mengeksplorasi simpul yang relevan. Studi ini menunjukkan bahwa metode manual dengan algoritma A* sesuai untuk studi kasus lokal yang memiliki skala terbatas tetapi membutuhkan presisi tinggi.

DAFTAR PUSTAKA

- [1] M. N. Fathoni, "Pemodelan Graf untuk Pemetaan Jaringan Jalan Perkotaan," *Jurnal Teknologi Informasi*, vol. 12, no. 2, pp. 102–110, 2021.
- [2] D. Jaya dan A. Permana, "Pemanfaatan Algoritma A* untuk Navigasi Kendaraan Otonom Menggunakan Peta Manual," *Jurnal Rekayasa dan Teknologi*, vol. 5, no. 2, pp. 50–58, 2023.
- [3] A. Sumantri dan B. Hidayattullah, "Penerapan Algoritma A* untuk Jalur Wisata Terpendek di Jakarta," *Jurnal Teknik Informatika*, vol. 10, no. 2, pp. 45–52, 2020.
- [4] D. Yogaswara dan E. Suhartono, "Perbandingan Algoritma A* dan Dijkstra untuk Jalur Evakuasi Tsunami," *Prosiding Seminar Nasional Teknologi*, vol. 5, pp. 123–130, 2021.
- [5] F. Sitepua dan G. Cahyadi, "Implementasi A* dalam Sistem Distribusi Barang Berbasis Google Maps," *Jurnal Sistem Informasi*, vol. 8, no. 3, pp. 67–74, 2022.
- [6] T. Kurniawan et al., "Integrasi A* dengan Data Lalu Lintas Real-time pada Sistem Navigasi Digital," *Journal of Smart Systems*, vol. 7, no. 2, pp. 76–85, 2024.
- [7] C. Wijaya dan M. Akbar, "Rancang Bangun Sistem Navigasi Berbasis Graf untuk Wilayah Perkotaan," *Jurnal Teknologi Terapan*, vol. 6, no. 1, pp. 25–32, 2024.
- [8] H. Nugroho dan L. Wahyuni, "Optimasi Rute Logistik Menggunakan Algoritma A* pada Peta Jalan Perkotaan," *Jurnal Transportasi Cerdas*, vol. 3, no. 1, pp. 11–19, 2023.
- [9] I. Kusuma dan R. Nugroho, "Analisis Heuristik Euclidean dan Haversine dalam Penentuan Jalur Terpendek," *Jurnal Sistem Informasi*, vol. 9, no. 1, pp. 31–38, 2022.
- [10] M. Taufik dan S. Azwar, "Penggunaan A* dan Sensor Lalu Lintas dalam Sistem Navigasi Kendaraan," *Jurnal Inovasi Sistem Cerdas*, vol. 5, no. 1, pp. 12–20, 2024.