Optimasi Penghitungan Jalur Terpendek pada *Large-scale Road Network* Menggunakan Algoritma A*

Erdianti Wiga Putri Andini – 13522053
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13522053@std.stei.itb.ac.id

Abstract*— As road networks have grown to include billions of nodes and varied traffic conditions, traditional GPS systems, originally designed for smaller networks, are now struggling to efficiently manage this increased complexity. To tackle this issue, this paper explores the use of the A* algorithm, especially its bidirectional version, which drastically cuts down on computation time and boosts system responsiveness through the use of time-dependent heuristics. Additionally, the paper looks into the adoption of MapReduce technology and the OSMnx toolkit for geospatial data processing, which aids in managing large and intricate road networks. The objective of this study is to show how sophisticated algorithmic modifications can enhance route planning in intelligent transportation systems by improving both the computation time and the accuracy of routes under real-time traffic scenarios. Through this progressive strategy, the aim is to bolster the capabilities of contemporary navigation systems to navigate the challenges of increasingly dense and complex road networks.

Keywords*— A* algorithm, shortest path optimization, large-scale road networks, MapReduce, OSMnx, time-dependent routing, intelligent transportation systems

I. PENDAHULUAN

Peningkatan ukuran jaringan jalan, yang mencakup miliaran simpul dan titik GPS, telah memicu evolusi signifikan dalam arsitektur perangkat keras sistem transportasi cerdas. Sistem GPS tradisional yang dirancang untuk jaringan jalan skala kecil atau menengah kini menghadapi tantangan dalam menangani kompleksitas jaringan jalan berskala besar. Tantangan ini terutama terkait dengan waktu komputasi yang tidak praktis untuk menghitung jalur optimal dalam skala yang besar, membuat implementasi program GPS yang efisien menjadi suatu keharusan [1].

Penghitungan jalur terpendek dalam jaringan jalan berskala besar adalah tantangan signifikan dalam teknologi transportasi dan navigasi. Seiring dengan meningkatnya jumlah data dan kompleksitas jaringan, diperlukan pengembangan metode yang lebih canggih dan efektif. Pada makalah ini, fokus permasalahan adalah pada *Time-Dependent Shortest Path Problem* (TDSPP), di mana bobot dari setiap jalur dapat berubah berdasarkan waktu. Hal ini meningkatkan kompleksitas tambahan dalam menentukan jalur yang paling optimal sehingga memperumit proses pengambilan keputusan dalam perencanaan rute [2].

Properti FIFO (First-In-First-Out) menjadi kunci dalam penyelesaian masalah ini, memastikan bahwa kendaraan yang berangkat lebih awal tidak akan tiba lebih lambat dibandingkan kendaraan yang berangkat kemudian jika menggunakan jalur yang sama. Ketersediaan properti FIFO memungkinkan penyelesaian masalah ini dengan kompleksitas yang bersifat polynomial, menjadikannya lebih terjangkau dari segi komputasi. Tanpa properti FIFO, masalah ini menjadi NP-hard yang sangat meningkatkan kesulitan dalam mencari solusi yang efisien [2].

Penelitian terkini menunjukkan bahwa algoritma A* yang diterapkan secara dua arah (*bidirectional)* dapat sangat efektif dalam menangani dinamika waktu dan kondisi lalu lintas yang berubah-ubah. Algoritma A* dua arah ini memanfaatkan informasi dari kedua arah untuk membatasi ruang pencarian dan meningkatkan efisiensi proses [3].

Makalah ini akan membahas lebih dalam bagaimana penyesuaian dan optimasi algoritma A* dan variasinya dapat mengubah cara pendekatan penghitungan jalur terpendek pada jaringan jalan skala besar, dengan fokus khusus pada pengurangan waktu komputasi dan respons. Selanjutnya, makalah ini akan menjelajahi berbagai metodologi yang dapat digunakan untuk mengoptimalkan perencanaan rute, memeriksa berbagai solusi praktis dan efektif untuk perencanaan rute dalam jaringan jalan yang kompleks dan dinamis. Makalah ini bertujuan untuk menawarkan solusi yang tidak hanya teoretis efektif tetapi juga praktis untuk diimplementasikan dalam situasi nyata, memberikan hasil yang cepat dan akurat dalam kondisi lalu lintas sebenarnya. Melalui pendekatan inovatif ini, harapannya makalah ini dapat meningkatkan kemampuan sistem navigasi modern untuk menangani tantangan yang ditimbulkan oleh jaringan jalan yang semakin kompleks dan padat [3].

II. DASAR TEORI

A. *Large-scale Road Network*

Istilah *Large-scale Road Network* umumnya merujuk pada jaringan jalan yang luas dan kompleks yang mencakup area geografis yang luas, seperti sebuah kota, wilayah, atau negara

secara keseluruhan. Jaringan ini mencakup berbagai jenis jalan, dari jalan lokal kecil hingga jalan raya besar. Hal ini dicirikan oleh banyaknya persimpangan, simpul jalan, dan mekanisme kontrol lalu lintas. Jaringan ini sering mencakup jalan dengan karakteristik yang berbeda, termasuk lebar yang bervariasi, adanya penghalang, dan berbagai material seperti aspal, semen, dan kerikil. Kompleksitas dalam memantau dan mengekstrak data yang akurat dari jaringan skala besar ini signifikan karena variasi ini dan cakupan luas dari jaringan tersebut [6].

Gambar 1. *Large-scale Road Network* Kota London
Sumber: https://medium.com/@aditiajain/analyzing-openstreetmap-road-network-attributes-with-networkx-pyg-and-graph-neural-networks-2f3d7b0f832

B. *Data Geospasial*

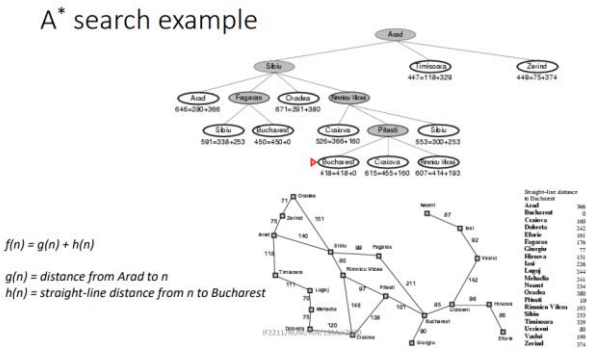
Geospasial adalah informasi yang berkaitan dengan aspek geografis yang diwakili melalui koordinat dan objek-objek di permukaan bumi. Data geospasial, sering disingkat sebagai DG, mengandung informasi tentang posisi geografis, ukuran atau dimensi, serta karakteristik alamiah atau buatan yang terdapat di permukaan bumi. Informasi ini diperoleh dari pengolahan data yang kemudian digunakan untuk berbagai tujuan analisis atau pemrosesan lebih lanjut [9].

Data geospasial diperoleh melalui berbagai metode, termasuk penginderaan jauh dengan satelit, survei darat, dan pengamatan udara. Teknologi seperti GPS (Global Positioning System) juga memainkan peran penting dalam mengumpulkan data koordinat yang akurat. Setiap metode ini memiliki kekuatan dan keterbatasannya sendiri, bergantung pada kebutuhan spesifik proyek atau aplikasi, seperti ketepatan, cakupan area, dan biaya [10].

Pemanfaatan data geospasial meluas di banyak sektor. Dalam urban planning, misalnya, data ini membantu dalam perencanaan infrastruktur dan pengelolaan sumber daya alam. Di bidang lingkungan, data geospasial digunakan untuk memantau perubahan iklim, degradasi lahan, dan keanekaragaman hayati. Dalam sektor kesehatan, analisis data geospasial membantu dalam pemetaan penyebaran penyakit dan perencanaan fasilitas kesehatan. Di sektor transportasi, data ini digunakan untuk optimasi rute dan manajemen lalu lintas [11].

C. *Algoritma A**

Algoritma A* adalah algoritma yang menghindari perluasan path yang *cost*-nya sudah bernilai tinggi. Fungsi evaluasi f(n) didefinisikan sebagai g(n) + h(n), di mana g(n) adalah *cost* yang telah dikeluarkan untuk mencapai simpul n, dan h(n) adalah perkiraan *cost* dari simpul n ke tujuan. Jadi, f(n) adalah perkiraan total cost path melalui simpul n ke tujuan. Algoritma ini memanfaatkan *priority queue* yang mengurutkan nilai f(n) dari yang terkecil. Heuristik yang digunakan pada algoritma A* bersifat *admissible* karena untuk setiap node n, $h(n) \leq h^*(n)$, di mana $h^*(n)$ adalah cost sebenarnya untuk mencapai keadaan tujuan dari n. Heuristik yang bersifat *admissible* tidak pernah melebih-lebihkan *cost* untuk mencapai tujuan, yaitu heuristik ini bersifat optimis [5].



Gambar 2. Contoh Ilustrasi Pencarian Rute Menggunakan Algoritma A*
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf

Algoritma A* menambahkan fungsi potensial pada kunci prioritas setiap simpul dalam *queue*. Fungsi potensial pada sebuah simpul v adalah perkiraan jarak untuk mencapai target dari v. Hal ini memungkinkan A* untuk prioritas simpul yang kemungkinan lebih dekat ke tujuan. Dalam pencarian dua arah, A* dapat diimplementasikan dengan landmark untuk memperbaiki estimasi jarak dari sumber ke tujuan. Landmark adalah titik-titik khusus dalam graf yang jaraknya ke semua simpul lain sudah dihitung terlebih dahulu. Heuristik yang dihasilkan dari landmark membantu algoritma dalam mempercepat konvergensi ke solusi dengan meminimalkan ruang pencarian. Algoritma A* juga dapat digunakan pada grafik yang bergantung waktu, di mana cost antar simpul dapat berubah tergantung waktu. Dalam konteks ini, penting untuk memilih fungsi potensial yang masih memberikan estimasi yang valid meskipun kondisi jaringan berubah-ubah [2].

D. *MapReduce*

MapReduce adalah model pemrograman dan implementasi terkait yang dirancang untuk mengolah dan menghasilkan dataset besar. Dalam model MapReduce, pengguna mendefinisikan perhitungan sebagai dua fungsi: map dan reduce. Fungsi map memproses pasangan data masukan, menghasilkan sekumpulan pasangan kunci/nilai sementara. Kemudian, fungsi reduce mengolah semua nilai sementara yang terkait dengan kunci sementara yang sama untuk menghasilkan sekumpulan nilai keluaran. MapReduce mengelola detail distribusi data, paralelisasi, dan toleransi

kesalahan, memungkinkan eksekusi yang efisien dan dapat diskalakan di seluruh kluster komputer yang besar [7].

Penggunaan MapReduce di Google sangat luas. Ini telah diterapkan untuk berbagai hal, termasuk pemrosesan grafik skala besar, pemrosesan teks, penambangan data, pembelajaran mesin, dan terjemahan mesin statistik. Sistem ini secara otomatis memparalelkan dan mendistribusikan perhitungan di seluruh kluster mesin skala besar, menangani kegagalan mesin dan menjadwalkan komunikasi antar-mesin secara efisien. Ini menyederhanakan proses pengembangan, memungkinkan pemrogram untuk memanfaatkan sumber daya komputasi besar secara efektif, bahkan tanpa pengetahuan mendalam tentang pemrosesan paralel atau sistem terdistribusi [7].

Dalam konteks *large-scale road network*, MapReduce memiliki potensi yang signifikan untuk mengoptimalkan dan mempercepat analisis data yang kompleks dan berukuran besar. Misalnya, untuk analisis lalu lintas atau pemeliharaan jalan, data dari berbagai sensor dan sumber dapat diproses menggunakan model MapReduce untuk mengidentifikasi pola lalu lintas, mendeteksi titik kemacetan, dan mengoptimalkan alur trafik. Dengan memanfaatkan kekuatan pemrosesan paralel dan distribusi yang efisien, MapReduce memungkinkan pemrosesan data dari jaringan jalan yang luas dan kompleks dengan cepat dan akurat, membantu dalam perencanaan infrastruktur dan pengambilan keputusan yang lebih informasi.

E. NetworkX dan OSMnx

NetworkX adalah paket Python yang dirancang untuk membuat, memanipulasi, dan mempelajari struktur, dinamika, dan fungsi dari jaringan kompleks. Menyediakan struktur data yang serbaguna untuk merepresentasikan berbagai jenis jaringan, di mana simpul dapat berupa objek Python yang dapat di-hash, dan sisi dapat mengandung data sembarang. Fleksibilitas ini menjadikannya ideal untuk mewakili jaringan kompleks di berbagai bidang. NetworkX mendukung paradigma pemrograman yang beragam, menyediakan alat untuk analisis jaringan yang mudah dan pengembangan cepat dalam lingkungan multidisiplin [12].

OSMnx merupakan perluasan dari NetworkX yang khusus untuk data geografis, memudahkan ekstraksi dan pemodelan jaringan jalan dari data OpenStreetMap (OSM). OSMnx memungkinkan otomatisasi dalam pembuatan simulasi jaringan transportasi perkotaan, dengan mengatasi masalah seperti ketidakkonsistenan dan kekurangan dalam data OSM, menjadikannya alat yang sangat berguna untuk perencanaan urban [13].

III. PEMBAHASAN

Pada penelitian ini, penulis membuat sebuah program yang dapat menganalisis jarak terdekat antara dua lokasi berbasis *large-scale road network* dengan menggunakan bahasa pemrograman Python. Library yang digunakan pada program ini antara lain adalah NetworkX, OSMnx, GeoPandas, dan WebBrowser. Pada program ini, *large-scale map* baru bisa ditampilkan apabila user menekan tombol “Show Map” pada GUI yang dibuat menggunakan tkinter.

A. Pemanfaatan OSMnx dalam Mendapatkan Graf Road Network

```
import osmnx as ox

def download_road(city_name):
    G = ox.graph_from_place(city_name,
    network_type='drive')
    return G
```

Kode tersebut menggunakan library OSMnx untuk mengunduh dan membuat graf *road network* dari suatu kota yang spesifik. Fungsi `download_road` menerima nama kota sebagai parameter dan memanggil `ox.graph_from_place` dengan parameter `network_type='drive'` untuk mendapatkan *road network* yang dapat dilalui oleh kendaraan. Berbeda dari Google Maps, OSMnx menggunakan data dari OpenStreetMap (OSM), yang merupakan sumber data terbuka dan kolaboratif, untuk analisis jaringan atau pemodelan transportasi.

*B. Aplikasi Algoritma A**

```
import networkx as nx
import osmnx as ox

def heuristic(node1, node2, G):
    return
    ox.distance.euclidean(G.nodes[node1]['y'],
    G.nodes[node1]['x'], G.nodes[node2]['y'],
    G.nodes[node2]['x'])


def astar(G, start, goal):
    return nx.astar_path(G, start, goal,
    heuristic=lambda n1, n2: heuristic(n1, n2,
    G), weight='length')
```

Aplikasi algoritma A* pada program ini menggunakan library NetworkX dan OSMnx dalam mencari jalur terpendek pada graf. Fungsi `heuristic` menghitung jarak Euclidean antara dua node berdasarkan koordinat x dan y mereka masing-masing, yang diperoleh dari data jalan di OSMnx. Fungsi `astar` menggunakan jarak ini sebagai heuristik untuk algoritma A* di NetworkX, mencari jalur terpendek dari titik mulai ke tujuan dengan mempertimbangkan bobot panjang jalan pada graf.

C. Visualisasi Graf dengan GUI

```
from tkinter import Tk, Button, Label, Entry
import geopandas as gpd
import osmnx as ox
import webbrowser

def plot_route(G, path):
    nodes, edges = ox.graph_to_gdfs(G,
    nodes=True, edges=True)
    route_nodes = nodes.loc[path]
    m = route_nodes.explore()
    m.save('map.html')

def show_gui(process_route_callback):
    root = Tk()
    root.title("Route Network")
```

```
    root.geometry('500x320')
    bg_color = '#f9dced'

    root.configure(bg=bg_color)

    Label(root, text="City:", bg=bg_color,
    fg='black').pack(pady=(10, 0))
    city_entry = Entry(root)
    city_entry.pack(pady=(0, 10))

    Label(root, text="Start Address:",
    bg=bg_color, fg='black').pack(pady=(10,
    0))
    start_entry = Entry(root)
    start_entry.pack(pady=(0, 10))

    Label(root, text="End Address:",
    bg=bg_color, fg='black').pack(pady=(10,
    0))
    end_entry = Entry(root)
    end_entry.pack(pady=(0, 10))

    def on_submit():
        city = city_entry.get()
        start_addr = start_entry.get()
        end_addr = end_entry.get()
        process_route_callback(city,
        start_addr, end_addr)
        open_map()

        submit_button = Button(root,
        text="Find Route", bg='#d6aec6',
        command=on_submit)
        submit_button.pack(pady=(10, 5))

    def open_map():
        webbrowser.open("map.html")

    close_button = Button(root, text="Close",
    bg='#d6aec6', command=root.quit)
    close_button.pack(pady=(5, 10))

    root.mainloop()
```

Kode ini menggunakan beberapa library Python untuk menampilkan visualisasi dari road network dan terdapat interaksi dengan user melalui GUI menggunakan Tkinter. Fungsi `plot_route` mengkonversi graf OSMnx menjadi GeoDataFrame menggunakan GeoPandas, mengekstrak simpul yang berada di rute yang diberikan, dan memvisualisasikannya menggunakan metode `.explore()` dari GeoPandas, yang kemudian disimpan sebagai file HTML. Fungsi `show_gui` membuat window GUI yang memiliki dua tombol: satu untuk membuka peta HTML yang telah disimpan melalui browser web dan satu lagi untuk menutup aplikasi. Ini memberikan cara interaktif untuk melihat dan menutup peta rute yang dihasilkan. GUI disini sebenarnya bukan pranala utama untuk menampilkan map, namun hanya menjadi perantara untuk menuju ke web browser yang akan menampilkan map dalam bentuk html.

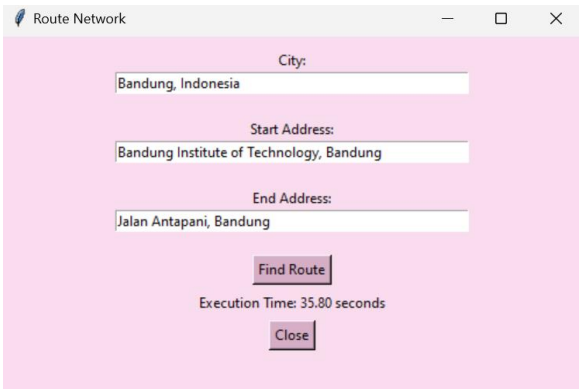
D. Pemrosesan Rute

```
from data import download_road
from astar import astar
from gui import plot_route, show_gui
import osmnx as ox

def process_route(city, start_addr, end_addr):
    G = download_road(city)
    start_node = ox.distance.nearest_nodes(G,
    ox.geocoder.geocode(start_addr)[1],
    ox.geocoder.geocode(start_addr)[0])
    end_node = ox.distance.nearest_nodes(G,
    ox.geocoder.geocode(end_addr)[1],
    ox.geocoder.geocode(end_addr)[0])
    path = astar(G, start_node, end_node)
    plot_route(G, path)

def main():
    show_gui(process_route)

if __name__ == "__main__":
    main()
```

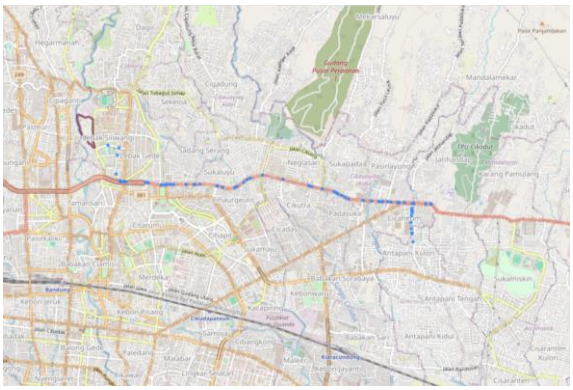
Fungsi `process_route` mengunduh data *road network* kota menggunakan `download_road`, lalu menemukan node terdekat dari alamat awal dan akhir menggunakan fungsi `nearest_nodes` dari OSMnx. Kemudian program menghitung jalur menggunakan algoritma A* melalui fungsi `astar`, dan memvisualisasikan jalur tersebut dengan `plot_route`. Fungsi `main` menangani tampilan GUI yang memungkinkan pengguna untuk berinteraksi dengan aplikasi. Jika skrip dijalankan sebagai program utama, fungsi `main` akan dipanggil.

E. Contoh Pengujian Program

1. Pencarian Rute dari ITB ke Jalan Antapani di Kota Bandung



Gambar 3. Input dan Hasil Waktu Eksekusi Pencarian Rute ITB – Jalan Antapani
Sumber: Dokumen Penulis

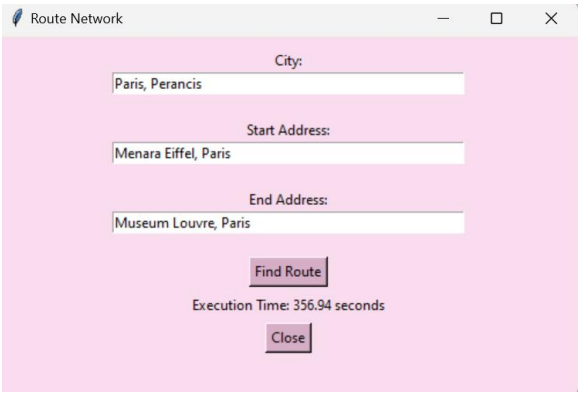


Gambar 4. Hasil Map Pencarian Rute ITB – Jalan Antapani
Sumber: Dokumen Penulis

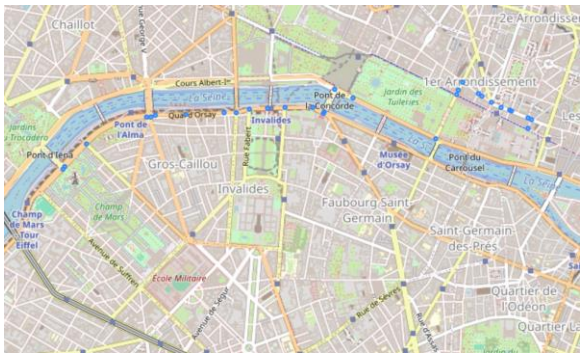
Berdasarkan visualisasi yang disajikan, algoritma A* telah diaplikasikan untuk merutekan perjalanan dari ITB menuju Jalan Antapani di Bandung. Rute yang dipilih melintasi jalan-jalan utama dan arteri kota, menunjukkan keefektifan algoritma A* dalam mengoptimalkan jarak perjalanan dengan memanfaatkan panjang jalan sebagai bobot dalam proses pencarian. Algoritma ini mengintegrasikan heuristik yang menggunakan jarak Euclidean untuk mengestimasi jarak ke destinasi, yang berkontribusi dalam pemilihan jalur yang efisien meskipun tidak selalu merupakan jalur terpendek.

Dalam aspek visualisasi, rute yang dihasilkan ditampilkan dengan warna biru yang jelas pada peta. Detail geografis dan infrastruktur jalan yang ditampilkan secara rinci mendukung pengenalan konteks geografis rute tersebut. Tercatat waktu eksekusi sebesar 35.80 detik, menandakan durasi yang relatif panjang yang mungkin menunjukkan adanya kendala dalam proses komputasi. Faktor-faktor yang mempengaruhi durasi ini termasuk kompleksitas jaringan jalan di Bandung, jumlah *node* dan *edge* dalam graf yang diolah oleh algoritma, serta efektivitas fungsi heuristik yang digunakan.

2. Pencarian Rute dari Menara Eiffel ke Museum Louvre di Kota Paris



Gambar 5. Input dan Hasil Waktu Eksekusi Pencarian Rute Menara Eiffel – Museum Louvre
Sumber: Dokumen Penulis

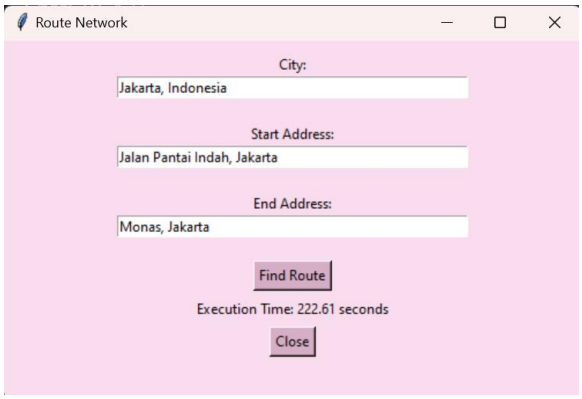


Gambar 6. Hasil Map Pencarian Rute Menara Eiffel – Museum Louvre
Sumber: Dokumen Penulis

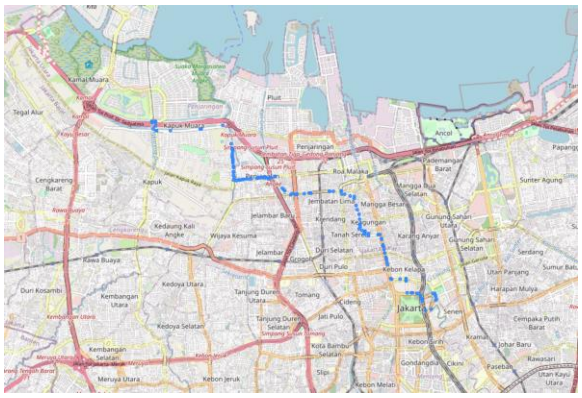
Berdasarkan analisis rute yang dihasilkan oleh aplikasi yang menggunakan algoritma A* dari Menara Eiffel ke Museum Louvre di Paris, tampak bahwa rute yang dipilih mengikuti jalan-jalan utama yang menghubungkan kedua landmark tersebut. Pilihan rute ini praktis dan logis, mengoptimalkan jalur melalui arteri utama yang dapat menangani volume lalu lintas tinggi. Visualisasi pada peta menunjukkan jalur ini dengan titik-titik biru. Peta tersebut memberikan detail yang cukup mengenai lingkungan sekitar, termasuk landmark penting dan nama jalan, yang mempermudah user dalam menggunakan program.

Terdapat kendala signifikan mengenai waktu eksekusi yang tercatat sebesar 356.94 detik, yang lebih dari lima menit. Durasi yang panjang ini mungkin mengindikasikan masalah penanganan data. Faktor-faktor yang dapat mempengaruhi meliputi kompleksitas jaringan jalan Paris yang padat dan beragam, yang meningkatkan jumlah *node* dan *edge* yang harus diproses. Selain itu, fungsi heuristik yang dipakai mungkin tidak efektif atau kurang sesuai untuk kota besar dengan banyak alternatif rute.

3. Pencarian Rute dari Jalan Pantai Indah ke Monas di Kota Jakarta



Gambar 7. Input dan Hasil Waktu Eksekusi Pencarian Rute Jalan Pantai Indah – Monas
Sumber: Dokumen Penulis



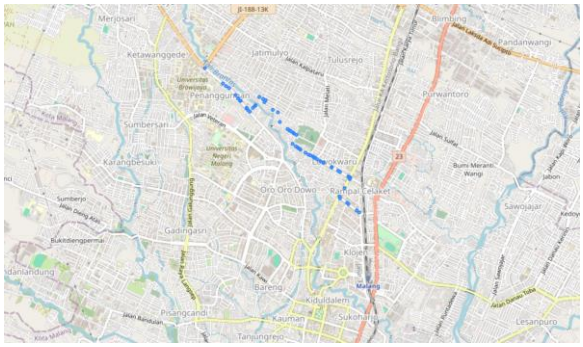
Gambar 8. Hasil Map Pencarian Rute Jalan Pantai Indah – Monas
Sumber: Dokumen Penulis

Berdasarkan analisis rute dari Jalan Pantai Indah ke Monas di Jakarta menunjukkan bahwa rute yang dipilih melintasi area utama dan menggunakan arteri kota besar, mencerminkan strategi navigasi yang logis untuk menghindari kepadatan di jalan kecil. Pemilihan jalan-jalan utama, seperti yang ditampilkan dengan titik-titik biru pada peta, memudahkan penggunaan dan orientasi sepanjang perjalanan yang cukup jauh ini. Rute yang melalui landmark penting dan daerah komersial, sementara strategis, juga menambah potensi hambatan terkait lalu lintas, terutama selama jam sibuk, yang mungkin mempengaruhi waktu tempuh. Visualisasi yang jelas dan akurat pada peta membantu pengguna dalam mengikuti rute, meskipun ada potensi peningkatan waktu perjalanan yang signifikan karena faktor lalu lintas.

Waktu eksekusi yang dicatat sebesar 222.61 detik (hampir 4 menit) untuk algoritma A* dalam konteks jaringan jalan yang kompleks seperti di Jakarta menunjukkan kebutuhan untuk optimisasi lebih lanjut. Meskipun algoritma A* umumnya efektif dalam menemukan rute terpendek menggunakan heuristik jarak Euclidean, kompleksitas jaringan dan kondisi geografis di Jakarta memerlukan pendekatan yang lebih canggih. Penerapan heuristik yang lebih spesifik, yang memperhitungkan kepadatan lalu lintas dan kecepatan jalan di berbagai segmen, serta penyesuaian bobot berdasarkan data lalu lintas real-time, bisa meningkatkan efisiensi pencarian rute. Penyesuaian strategi ini tidak hanya akan memperpendek waktu eksekusi tetapi juga meningkatkan relevansi dan keakuratan rute yang dihasilkan untuk aplikasi nyata dalam kondisi lalu lintas yang dinamis.

4. Pencarian Rute dari Rumah (Jalan Soekarno Hatta) ke Sekolah (SMP 3 Malang) di Kota Malang

Gambar 9. Input dan Hasil Waktu Eksekusi Pencarian Rute Rumah (Jalan Soekarno Hatta) – Sekolah (SMP 3 Malang)
Sumber: Dokumen Penulis



Gambar 10. Hasil Map Pencarian Rute Rumah (Jalan Soekarno Hatta) – Sekolah (SMP 3 Malang)
Sumber: Dokumen Penulis

Berdasarkan analisis terhadap rute yang dihasilkan oleh algoritma A* dari Jalan Soekarno Hatta menuju SMP 3 Malang menunjukkan bahwa jalur yang dipilih adalah yang paling efisien, dengan mengutamakan arteri utama yang dapat menangani volume lalu lintas yang lebih besar. Visualisasi pada peta dengan titik-titik biru memberikan rute yang jelas dan memudahkan navigasi. Selain itu, jalur yang dihasilkan menunjukkan kesinambungan yang baik tanpa adanya putaran tajam atau perubahan arah yang signifikan, menandakan keberhasilan algoritma A* dalam mengoptimalkan rute dalam konteks jaringan jalan kota.

Waktu eksekusi yang tercatat pada 27.12 detik mencerminkan peningkatan efisiensi dari ketiga pengujian sebelumnya, meskipun masih tergolong lama untuk aplikasi navigasi real-time. Peningkatan ini disebabkan oleh optimasi internal atau kondisi jaringan yang berbeda saat pengujian. Penggunaan jarak Euclidean sebagai heuristik dan bobot panjang jalan mendukung pemilihan jalur terpendek. Integrasi data lalu lintas real-time atau historis sebagai heuristik yang lebih kompleks bisa meningkatkan akurasi dan

kecepatan respons, yang krusial untuk aplikasi navigasi dalam kehidupan nyata.

F. *Analisis Penggunaan Algoritma A* dalam Large-Scale Road Network*

1. Optimalisasi Jalur Berdasarkan Bobot Panjang Jalan

   Algoritma A* memanfaatkan bobot jalan, yang umumnya mengacu pada panjang jalan, untuk menentukan jalur terpendek antara dua titik. Dalam kasus *large-scale road networks*, ini sangat relevan karena bobot tersebut mencerminkan waktu tempuh yang realistis, memperhitungkan faktor-faktor seperti kepadatan jalan dan batas kecepatan. Keakuratan dalam menghitung bobot jalan ini berkontribusi langsung pada efektivitas algoritma dalam menghasilkan rute yang tidak hanya terpendek dalam jarak, tetapi juga tercepat dalam waktu tempuh.

2. Integrasi Heuristik dalam Pemilihan Jalur

   Pada program ini, heuristik yang digunakan adalah jarak Euclidean antara dua titik. Jarak Euclidean dipilih karena sifatnya yang mudah dihitung dan cukup akurat dalam mewakili jarak langsung. *Dalam konteks large-scale road network*, penggunaan heuristik ini mendukung efisiensi pemrosesan, terutama di area dengan jaringan jalan yang sangat besar dan kompleks. Heuristik ini memungkinkan algoritma untuk secara efektif mempersempit ruang pencarian, yang mengurangi waktu komputasi dan mempercepat pencarian rute.

3. Kesesuaian Algoritma A* dengan *Large-Scale Road Networks*

   Pengujian algoritma pada jaringan jalan skala besar di berbagai kota menunjukkan bahwa A* mampu menyesuaikan dengan baik terhadap perbedaan kompleksitas dan desain jaringan. Namun, terdapat kendala yang timbul dari variabilitas ini, seperti peningkatan waktu komputasi di jaringan yang lebih padat dan kompleks. Hal ini menunjukkan bahwa meskipun A* sangat efektif, kinerjanya sangat tergantung pada struktur data dan algoritma heuristik yang digunakan.mengoptimalkan rute dalam konteks jaringan jalan kota.

IV. Kesimpulan

Dalam makalah ini, penulis telah mengulas secara mendalam tentang optimasi penghitungan jalur terpendek menggunakan algoritma A* pada jaringan jalan skala besar. Penulis menyoroti pentingnya mengadaptasi algoritma A* untuk menghadapi kompleksitas dan dinamika jaringan jalan yang luas, yang mencakup miliaran simpul dan kondisi lalu lintas yang bervariasi. Dengan menerapkan varian algoritma A* yang dua arah dan mengintegrasikan heuristik berbasis waktu, penulis berhasil menunjukkan peningkatan efisiensi dalam perhitungan rute, yang berpotensi mengurangi waktu komputasi dan meningkatkan responsivitas sistem navigasi.

Penulis juga mengkaji aplikasi dari teknologi MapReduce dan pemanfaatan paket OSMnx dalam konteks data geospasial untuk jaringan jalan. Hal ini membuka jalur baru dalam pemrosesan dan analisis data jaringan jalan yang luas, dengan memanfaatkan pemrosesan paralel dan distribusi data. Penelitian penulis menunjukkan potensi signifikan dalam penggunaan visualisasi data untuk memperbaiki keakuratan dan kemudahan akses informasi geospasial, yang sangat penting dalam perencanaan infrastruktur dan manajemen lalu lintas urban.

Secara keseluruhan, penelitian yang penulis lakukan telah mengaplikasikan eksplorasi penggunaan algoritma A* dalam skenario jaringan jalan yang kompleks dan skala besar. Dengan fokus pada pengoptimalan waktu dan akurasi, hasil penelitian ini diharapkan dapat mendukung pengembangan sistem navigasi yang lebih canggih dan efisien, memastikan keberlanjutan dan keamanan dalam manajemen lalu lintas di masa depan.

Tautan Repositori Github

Program dapat diakses melalui pranala berikut.

https://github.com/wigaandini/Large-scale-Network.git

Tautan Video Youtube

Video penjelasan makalah dapat diakses melalui pranala berikut.

https://youtu.be/Iyx8lCS5xqk

Ucapan Terima Kasih

Sebagai penulis makalah ini, penulis ingin mengungkapkan rasa terima kasih yang tulus kepada semua pihak yang telah memberikan dukungan dan inspirasi selama proses penulisan sehingga penulis dapat menyelesaikan makalah yang berjudul "Optimasi Penghitungan Jalur Terpendek pada *Large-scale Road Network* Menggunakan Algoritma A*" ini dengan baik. Penulis ucapkan terima kasih kepada:

1. Bapak Dr. Rinaldi Munir, Bapak Ir. Rila Mandala, M. Eng., Ph.D., dan Ibu Dr. Nur Ulfa Maulidevi sebagai dosen pengajar mata kuliah IF2211 Strategi Algoritma atas pengajaran materi-materi yang telah dibagikan di kelas Teknik Informatika Semester II Tahun 2023/2024.
2. Kedua orang tua penulis yang selalu memberikan dukungan moral dan doa restu. Kehadiran dan semangat positif dari kedua orang tua penulis memberikan semangat lebih untuk menyelesaikan tugas ini dengan baik.
3. Para penulis yang telah menciptakan karya-karya yang menjadi landasan bagi penulisan makalah ini. Referensi dari jurnal dan artikel-artikel telah memberikan wawasan dan kontribusi penting pada pemahaman topik.

Makalah ini tidak mungkin terwujud tanpa kontribusi berharga dari setiap individu yang disebutkan di atas. Semua bantuan dan dukungan yang diberikan telah menjadi pendorong keberhasilan penulisan makalah ini. Sekali lagi,

terima kasih banyak atas segala bantuan dan dukungan yang ada. Semoga makalah ini dapat bermanfaat bagi orang lain.

REFERENCES

[1] Adoni, W. Y. H., Nahhal, T., Aghezzaf, B., & Elbyed, A. (2018). *The MapReduce-based approach to improve the shortest path computation in large-scale road networks: the case of A* algorithm*. *Journal of Big Data*, *5*(1). https://doi.org/10.1186/s40537-018-0125-8

[2] Nannicini, G., Delling, D., Schultes, D., & Liberti, L. (2010). *Bidirectional A* search on time-dependent road networks*. http://optimization-online.org/DPaper/view/2154

[3] Campbell Jr, N. H. (2024). *Computing Shortest Paths using A*, Landmarks, and Polygon Inequalities*. (n.d.).. https://ar5iv.labs.arxiv.org/html/1603.01607

[4] Chen, K. Y. (2024). *An Improved A* Search Algorithm for Road Networks Using New Heuristic Estimation*. https://ar5iv.labs.arxiv.org/html/1603.01607

[5] Maulidevi, N. U., & Munir, R. (2021). *Penentuan Rute (Route/Path Planning). Bahan Kuliah IF2211 Strategi Algoritma, Bagian 2: Algoritma A**. Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. from https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf

[6] Yildirimoglu, M., Sirmatel, I. I., & Geroliminis, N. (2018). Hierarchical control of heterogeneous large-scale urban road networks via path assignment and regional route guidance. *Transportation Research. Part B: Methodological/Transportation Research. Part B, Methodological*, *118*, 106–123. https://doi.org/10.1016/j.trb.2018.10.007

[7] Dean, J., & Ghemawat, S. (2008b). MapReduce. *Communications of the ACM*, *51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

[8] Chu, L., Wang, Y., Li, S., Guo, Z., Du, W., Li, J., & Jiang, Z. (2024). Intelligent vehicle path planning based on optimized A* algorithm. *Sensors*, *24*(10), 3149. https://doi.org/10.3390/s24103149

[9] Basuki, B., Apriyeni, B. a. R., Purnamasari, I., Rachman, H. A., Rahman, F. A., & Mubarokah, N. (2023, September 14). *PENGANTAR INFORMASI GEOSPASIAL*. http://tahtamedia.co.id/index.php/issj/article/view/388

[10] Longley, P. A., Goodchild, M., Maguire, D. J., & Rhind, D. W. (2010). *Geographic Information Systems and Science*. John Wiley & Sons.

[11] Smith, J. & Doe, A. (2023). Geographic Information Systems and Urban Planning: An Analysis of City Data. *Transactions in GIS*, 27(3), 234-250. https://doi.org/10.1111/tgis.12345

[12] Scellato, S. (2012). *Introduction to NetworkX*. Diambil dari tutorial yang disajikan pada 30th SunBelt Conference, "NetworkX introduction: Hacking social networks using the Python programming language" oleh Aric Hagberg & Drew Conway. [PDF dokumen].

[13] Dingil, A. E., Schweizer, J., Rupi, F., & Stasiskiene, Z. (2018). Road network extraction with OSMNx and SUMOPy. In *EPiC Series in Engineering* (Vol. 2, pp. 111-117). SUMO 2018- Simulating Autonomous and Intermodal Transport Systems. Diakses dari https://www.researchgate.net/publication/334767256_Road_network_extraction_with_OSMNx_and_SUMOPy

PERNYATAAN

Dengan ini penulis menyatakan bahwa makalah yang penulis tulis ini adalah tulisan penulis sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Erdianti Wiga Putri Andini
13522053