

**SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN ANGKUTAN UMUM DI  
KOTA MEDAN BERBASIS MOBILE MENGGUNAKAN  
OPENTRIPPLANNER**

**SKRIPSI**

**REKSA ANJANGSARA**

**171401097**



**PROGRAM STUDI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN ANGKUTAN UMUM DI  
KOTA MEDAN BERBASIS MOBILE MENGGUNAKAN  
OPENTRIPPLANNER**

**SKRIPSI**

Diajukan untuk melengkapi dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer

**REKSA ANJANGSARA**

**171401097**



**PROGRAM STUDI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**PERSETUJUAN**

Judul : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN  
ANGKUTAN UMUM DI KOTA MEDAN BERBASIS  
MOBILE MENGGUNAKAN OPENTRIPPLANNER

Kategori : SKRIPSI

Nama : REKSA ANJANGSARA

Nomor Induk Mahasiswa : 171401097

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA

Tanggal Sidang : 29 April 2024

Komisi Pembimbing :

Dosen Pembimbing I

Dosen Pembimbing II

Sri Melvani Hardi S.Kom., M.Kom.  
NIP. 198805012015042006

Fuzy Yustika Manik S.Kom., M.Kom.  
NIP. 198710152019032010

Diketahui/Disetujui oleh  
Program Studi S1 Ilmu Komputer  
Ketua,

Dr. Amalia S.T., M.T.  
NIP. 197812212014042001

**PERNYATAAN****SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN ANGKUTAN UMUM DI  
KOTA MEDAN BERBASIS MOBILE MENGGUNAKAN  
OPENTRIPLANNER****SKRIPSI**

Saya mengakui bahwa Skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Maret 2024

Reksa Anjangsara

171401097

## PENGHARGAAN

Penulis ingin mengucapkan rasa syukur dan terima kasih kepada Allah Subhanahu Wa Ta'ala atas berkah-Nya yang telah memungkinkan penulis menyelesaikan Skripsi dengan judul “Sistem Pendukung Keputusan Pemilihan Angkutan Umum Di Kota Medan Berbasis Mobile Menggunakan Opentripplanner” sebagai persyaratan untuk menyelesaikan pendidikan dalam program studi S1 Ilmu Komputer di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara serta meraih gelar sarjana.

Dengan rendah hati, penulis ingin mengungkapkan penghargaan yang besar kepada semua pihak yang telah memberikan bantuan dalam proses penyelesaian skripsi ini, antara lain :

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia ST., M.T. selaku Ketua Program Studi Ilmu Komputer Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi S.Kom., M.Kom selaku Sekretaris Program Studi Ilmu Komputer Universitas Sumatera Utara serta sebagai Dosen Pembimbing I yang telah memberikan dukungan berupa bimbingan, evaluasi konstruktif, serta motivasi kepada penulis, yang sangat berperan dalam menyelesaikan skripsi ini.
5. Ibu Fuzy Yustika Manik S.Kom., M.Kom selaku Dosen Pembimbing II yang telah memberikan kontribusi berupa masukan, evaluasi kritis, serta saran kepada penulis, yang sangat membantu dalam menyelesaikan skripsi ini.
6. Bapak Jos Timanta Tarigan, S.Kom, M.Sc. selaku Dosen Penguji I yang telah memberikan masukan serta saran kepada penulis dalam penulisan Skripsi ini.
7. Ibu Anandhini Medianty Nababan S. Kom., M. T. selaku Dosen Penguji II yang telah memberikan kontribusi dalam bentuk masukan dan saran kepada penulis dalam proses penulisan skripsi ini.
8. Ibu Dr. Amalia ST., M.T. selaku Dosen Pembimbing Akademik yang telah memberikan dorongan kepada penulis sehingga skripsi ini dapat terselesaikan.

9. Orangtua penulis, Risdhawaty Tanjung yang selalu mendoakan penulis dan memberikan dorongan serta motivasi untuk dapat menyelesaikan skripsi ini.
10. Dinas Perhubungan Kota Medan yang telah menyediakan data-data dan memberikan masukan selama penelitian dan penulisan skripsi ini.
11. Asosiasi Trufi yang telah berbaik hati memberikan sponsor berupa langganan *Virtual Private Server* sebagai *Backend Server* dari sistem aplikasi penelitian ini.
12. Saudari Tengku Arifah Inayyah yang telah dengan rela menjadi tempat untuk berdiskusi, mendengarkan keluhan, serta memberikan dukungan kepada penulis, baik secara praktis maupun dalam memperbaiki semangat yang terkadang meredup saat proses pembuatan skripsi ini.

Medan, Maret 2024

Penulis,

Reksa Anjangsara

## **SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN ANGKUTAN UMUM DI KOTA MEDAN BERBASIS MOBILE MENGGUNAKAN OPENTRIPPLANNER**

### **ABSTRAK**

Aplikasi navigasi adalah jenis aplikasi yang memegang peranan penting dalam pemetaan daerah perkotaan dan pemilihan jenis moda transportasi. Meskipun telah ada beberapa aplikasi navigasi yang populer seperti *Google Map* & *Moovit*, pemetaan moda transportasi umum non-formal masih sulit dilakukan karena minimnya data rute yang tersedia, dan format data yang tak didukung oleh sistem aplikasi-aplikasi navigasi tersebut. Oleh karena itu, pada penelitian ini dikembangkan sebuah aplikasi yang dapat mengubah format data rute menjadi format yang dapat digunakan oleh aplikasi navigasi dan menghasilkan rute yang optimal dengan menggunakan *OpenTripPlanner* sebagai *Framework* perutean. Aplikasi ini bekerja dengan menggunakan algoritma A\* dengan penyesuaian *cost* (*Generalized Cost A\**), dan menghasilkan beberapa *output* yang diurutkan berdasarkan *cost*. Peneliti melakukan pengujian terhadap hasil dari penggunaan *framework* ini untuk mencari rute dari Gedung A Ilmu Komputer USU ke Lapangan Merdeka Medan. Berdasarkan hasil dari pengujian tersebut, peneliti mendapatkan satu hasil *output*, yaitu rute yang menggunakan angkot dengan trayek KPUM 10 & KPUM 78 dengan jarak tempuh total sejauh 7970 meter. Hal ini disebabkan karena *output* lainnya tidak memenuhi syarat kriteria dari parameter yang ditetapkan. Temuan ini menunjukkan bahwa sistem ini beroperasi sesuai dengan perancangan yang telah direncanakan.

Kata-kata kunci : Transportasi Umum, Sistem Pendukung Keputusan, Pencarian Rute, OpenTripPlanner.

## **MOBILE-BASED PUBLIC TRANSPORT ELECTION DECISION SUPPORT SYSTEM IN MEDAN CITY USING OPENTRIPPLANNER**

### **ABSTRACT**

Navigation applications are a type of application that plays an important role in mapping urban areas and selecting types of transportation modes. Even though there are several popular navigation applications such as Google Map & Moovit, mapping non-formal public transportation modes is still difficult due to the lack of route data available, and data formats that are not supported by the navigation application systems. Therefore, in this research an application was developed that can change the route data format into a format that can be used by navigation applications and produce optimal routes using OpenTripPlanner as a routing framework. This application works using the A\* algorithm with cost adjustments (Generalized Cost A\*), and produces several outputs sorted by cost. Researchers tested the results of using this framework to find a route from USU Computer Science Building A to Medan's Merdeka Square. Based on the results of this test, the researchers obtained one output result, namely a route using public transportation with routes KPUM 10 & KPUM 78 with a total distance of 7970 meters. This is because other outputs do not meet the criteria requirements of the specified parameters. These findings indicate that this system operates according to the planned design.

**Keywords :** Public Transportation, Decision Support System, Route Search, OpenTripPlanner.



## DAFTAR ISI

PERSETUJUAN .....	iii
PERNYATAAN .....	iv
PENGHARGAAN .....	v
ABSTRAK.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xiii
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	6
1.3. Batasan Masalah.....	7
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	7
1.6. Metodologi Penelitian .....	8
1.7. Sistematika Penulisan.....	9
BAB 2 LANDASAN TEORI.....	11
2.1. Sistem Transportasi Angkutan Umum .....	11
2.2. Kemacetan .....	12
2.3. Sistem Pendukung Keputusan .....	12
2.4. Sistem Database OpenStreetMap .....	12
2.5. Format GTFS (General Transit Feed Specification) .....	13
2.6. Sistem Framework OpenTripPlanner .....	15
2.7. REST API.....	18
2.8. Sistem Framework Flutter .....	21
BAB 3 ANALISIS DAN PERANCANGAN SISTEM .....	22
3.1. Analisis Sistem.....	22
3.2. Analisis Permasalahan.....	22
3.3. Analisis Kebutuhan Sistem .....	23
3.3.1. Kebutuhan Fungsional .....	23
3.3.2. Kebutuhan non-fungsional .....	23
3.4. Perancangan Sistem.....	24
3.4.1. Perancangan Arsitektur Sistem .....	24

3.4.2.	Perancangan Flowchart .....	25
3.4.3.	Use-case Diagram .....	26
3.4.4.	Activity Diagram .....	26
3.4.5.	Sequence Diagram .....	28
3.5.	Perancangan Antarmuka Sistem (Interface).....	28
3.6.1	Tampilan Halaman Depan .....	29
3.6.2	Tampilan Halaman Output dari Halaman Perencana Rute .....	30
3.6.3	Tampilan Halaman Detail Rute .....	32
3.6.4	Tampilan Halaman Daftar Rute .....	33
3.6.5	Tampilan Halaman Detail Rute Trayek .....	34
3.6.6	Tampilan Halaman Tempat Favorit .....	35
BAB 4 IMPLEMENTASI DAN PENGUJIAN .....		37
4.1.	Implementasi Sistem .....	37
4.1.1.	Aplikasi Pendukung .....	37
4.1.2.	Perangkat Pendukung .....	38
4.2.	Pengujian Sistem .....	38
4.2.1.	Pengujian dan Perbandingan Algoritma A* dengan OpenTripPlanner .	39
4.2.2.	Pengujian OpenTripPlanner tanpa Parameter Tambahan .....	40
4.2.3.	Pengujian OpenTripPlanner dengan Parameter Tambahan .....	47
BAB 5 KESIMPULAN DAN SARAN .....		50
5.1.	Kesimpulan.....	50
5.2.	Saran.....	50
DAFTAR PUSTAKA .....		51

## DAFTAR TABEL

Tabel 2.1. Format file-file GTFS .....	14
Tabel 2.2. Perbedaan Fitur OTP1 & OTP2.....	15
Tabel 2.3. Perbedaan Syarat Penggunaan OTP1 & OTP2 .....	16
Tabel 4.1 Aplikasi Pendukung .....	37
Tabel 4.2 Perangkat Pendukung .....	38
Tabel 4.3. Biaya dari Waypoint yang terlibat .....	45
Tabel 4.4 Waypoint dan biaya total dari setiap rencana perjalanan.....	45
Tabel 4.5 Jumlah Transfers, Jarak Berjalan, dan Jarak Total .....	46
Tabel 4.6 Biaya baru yang telah ditimbang (Weighted Cost).....	46

## DAFTAR GAMBAR

Gambar 2.1. Contoh Rute .....	11
Gambar 2.2. Database Rute yang sudah di-input.....	13
Gambar 2.3. Contoh Penggunaan REST API .....	19
Gambar 3.1. Diagram Arsitektur Sistem.....	24
Gambar 3.2. Flowchart Sistem.....	25
Gambar 3.3. Use Case Diagram.....	26
Gambar 3.4. Activity Diagram.....	27
Gambar 3.5. Sequence Diagram .....	28
Gambar 3.6. Tampilan Halaman Depan .....	29
Gambar 3.7. Tampilan Halaman Output.....	31
Gambar 3.8. Tampilan Detail Rute .....	32
Gambar 3.9. Tampilan Halaman Daftar Rute .....	33
Gambar 3.10. Tampilan Halaman Detail Rute Trayek .....	34
Gambar 3.11. Tampilan Halaman Tempat Favorit .....	36
Gambar 4.1. Lokasi Koordinat Node Asal & Tujuan .....	39
Gambar 4.2. Output Rute Terpendek dengan Algoritma A* .....	40
Gambar 4.3. Kumpulan Penjelasan dari Ilustrasi.....	44
Gambar 4.4. Tampilan halaman output pada aplikasi smartphone .....	48
Gambar 4.5. Tampilan halaman detail rute pada aplikasi smartphone .....	49

**DAFTAR LAMPIRAN**

SURAT PERMOHONAN IZIN PENELITIAN .....	53
SURAT KETERANGAN RISET BADAN RISET DAN INOVASI DAERAH .....	54
SURAT KETERANGAN RISET DINAS PERHUBUNGAN KOTA MEDAN .....	55

## **BAB 1**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Kota Medan adalah kota terbesar ketiga di Indonesia dengan jumlah penduduk 2.435.252 pada tahun 2020 dan laju pertumbuhan penduduk sebesar 0,79% pada tahun 2021. Panjang jalan di Kota Medan adalah 3.279,5 KM dengan jumlah kendaraan sebanyak 288.378 pada tahun 2020. Menurut Wakil Wali Kota Medan Akhyar Nasution memprediksi bahwa kota Medan akan macet total pada tahun 2024 berdasarkan statistik rasio jumlah kendaraan pribadi dan kendaraan umum di Kota Medan adalah 97,8% banding 2,2% ditahun 2016 (Badan Pusat Statistik Kota Medan 2022). Kemacetan adalah suatu kondisi dimana jumlah kendaraan yang mencoba menggunakan jalan dalam satu waktu melebihi kemampuan jalan tersebut untuk menerima beban pada tingkat pelayanan secara umum (Aftabuzzaman 2007).

Berdasarkan statistik tersebut dapat dilihat bahwa masyarakat kota Medan lebih memilih untuk menggunakan kendaraan pribadi dibandingkan kendaraan umum sehingga menyebabkan kemacetan. Statistik tersebut juga didukung oleh penelitian terdahulu berjudul “Analisis Pengaruh Jumlah Transportasi Umum dan Transportasi Online Terhadap Kemacetan Di Kota Medan”, menyebutkan bahwa hal yang melatarbelakangi rendahnya minat masyarakat Kota Medan dalam menggunakan transportasi umum adalah kurangnya ketersediaan dan juga rendahnya fasilitas transportasi umum itu sendiri. Dalam penelitian itu disebutkan juga bahwa hal yang memicu kemacetan selain dari kurangnya penggunaan transportasi umum adalah buruknya infrastruktur Kota Medan, dilanggarnya Peraturan Menteri tentang kuota transportasi online sehingga jumlah transportasi online melebihi kuota yang seharusnya. Perilaku pengemudi transportasi umum yang tidak taat terhadap peraturan lalu lintas juga menjadi salah satu penyebabnya, berhenti disembarang tempat untuk menaik-turunkan penumpang, dan juga kebiasaan Ngetem (Istilah untuk angkutan atau kendaraan umum yang sedang berhenti untuk menunggu penumpang) (Adam B. Nababan 2018).

Berlandaskan uraian sebelumnya, dapat disimpulkan bahwa terdapat tiga peran yang menyebabkan kemacetan, yaitu Operator angkutan umum, Pengguna angkutan umum & Pemerintahan atau penegak hukum. Setiap peran tersebut tentunya mempunyai solusinya masing-masing. Operator angkutan umum dapat lebih mematuhi peraturan lalu lintas sehingga tidak menghambat arus lalu lintas. Penegak hukum mungkin dapat menindak oknum-oknum yang melanggar peraturan dan mempertegas peraturan-peraturan yang berlaku. Dan terakhir, pengguna atau masyarakat dapat meningkatkan kesadaran untuk lebih menggunakan transportasi umum. Dalam hal ini penulis yang berperan sebagai masyarakat dan juga mahasiswa di bidang ilmu komputer berencana untuk membuat sebuah sistem yang berfungsi untuk memetakan transportasi umum dengan harapan untuk meningkatkan penggunaan transportasi umum dalam masyarakat.

Penelitian terdahulu yang berjudul “Mind the Map! The Impact of Transit Maps on Path Choice in Public Transit” membahas dampak dari pemetaan rute angkutan umum di London menyebutkan bahwa terdapat 4 hal yang dapat disampaikan melalui pemetaan yang baik ; Distortion, Restoration, Codification, & Cognition. Distortion adalah pemetaan angkutan umum yang sebelumnya kurang akurat secara geografis dalam hal arah dan jarak dapat ditingkatkan kualitasnya. Lalu, Restoration adalah memulihkan fitur geografis yang sebelumnya terdistorsi agar terdapat kesinambungan antara peta angkutan umum dengan peta yang sebenarnya. Codification adalah proses konversi dari rute dan fasilitas angkutan umum menjadi peta transit melalui simbol ataupun sintaks. Dan terakhir, Cognition adalah dampak kognitif dari calon pengguna angkutan umum ketika mendapatkan visualisasi dari peta transit yang berpotensi meningkatkan daya tarik, baik dari penggunaan angkutan umum itu sendiri ataupun citra dari agency angkutan umum-nya. Penelitian ini juga membuktikan perbandingan antara angkutan umum tanpa pemetaan yang baik dengan pemetaan yang tidak baik dalam studi kasus di London Underground membuktikan bahwa pemetaan angkutan umum yang baik dapat meningkatkan minat penggunaannya hingga dua kali lipat (Zhan Guo 2011).

Berdasarkan latar belakang diatas, pada penelitian ini dirancang sebuah sistem yang memberikan rekomendasi rute angkutan umum yang ada di Kota

Medan dengan tujuan menentukan rute optimal berdasarkan jarak terpendek. Pada sistem diberikan inputan berupa dua titik pada peta rute yang menghasilkan rekomendasi beberapa rute angkutan umum beserta nomor trayek angkutan umum dan jarak yang ditempuh dari tiap rute yang dihasilkan.

Sistem rekomendasi penentuan rute angkutan umum ini merupakan implementasi dari Computer Based Information System (CBIS) yaitu Decision Support System (DSS) atau Sistem Pendukung Keputusan (SPK). Computer Based Information System adalah sistem terstruktur yang mengandalkan perangkat keras dan perangkat lunak dari komputer untuk memproses, mengumpulkan, menyimpan, maupun mendistribusikan sumber daya yang tersedia (Danimir Mandic 2012). Dan Sistem Pengambilan Keputusan adalah salah satu jenis dari Computer Based Information System yang digunakan disemua tingkat organisasi untuk membuat keputusan dan memecahkan masalah yang sudah sudah sebagian terstruktur (Khodashahri 2013).

Pada penelitian ini penulis ingin membuat sebuah sistem yang dapat menentukan rute terpendek dari dua lokasi berbeda di Kota Medan yang dapat ditempuh dengan transportasi umum dengan OpenTripPlanner. OpenTripPlanner atau OTP adalah sistem perangkat lunak perencanaan perjalanan multimoda. Sistem ini juga merupakan perangkat lunak open-source, perangkat lunak open-source adalah perangkat lunak yang tersedia dengan lisensi yang memungkinkan pengguna untuk melihat, memodifikasi, dan mendistribusikan kode sumbernya. Dengan kata lain, perangkat lunak open-source mendorong kolaborasi, transparansi, dan pembangunan berbasis komunitas. Berbeda dengan perangkat lunak berpemilik, yang kode sumbernya dibatasi dan dikontrol oleh satu entitas. (Bruce Perens 1999).

OTP dibuat pertama kali pada tahun 2009 dan masih berkembang hingga saat ini. OTP dirancang sedemikian rupa untuk menerima input dari open data sources seperti OpenStreetMap (OSM) dan General Transit Feed Specification (GTFS). Memungkinkan pengguna atau pengembang untuk merencanakan perjalanan yang dapat menggabungkan beberapa moda transportasi, seperti bersepeda atau berjalan kaki untuk mencapai transportasi umum. Hal ini juga dapat menerapkan model elevasi digital pada jaringan jalan OSM, sehingga memungkinkan, misalnya, permintaan rute ramah sepeda. OTP berfungsi di



seluruh dunia selama ada cakupan peta OSM di wilayah tersebut. Dukungan untuk jadwal transit dan penghitungan ketinggian bergantung pada data yang tersedia.

Pada awal terbentuknya, OTP bekerja dengan menggunakan algoritma A\* dan algoritma Dijkstra, namun pada tahun 2018, ditambahkan algoritma RAPTOR untuk menggantikan algoritma Dijkstra untuk menangani pemetaan angkutan umum sambil mempertahankan algoritma A\* untuk rute di jalan untuk berjalan kaki, bersepeda, dan mengemudi. Dengan demikian, sistem ini akan menyesuaikan algoritma yang digunakan tergantung dengan kasus yang diberikan (docs.opentripplanner.org, n.d.).

Algoritma A\* adalah salah satu algoritma perencanaan jalur yang paling terkenal, yang dapat diterapkan pada metrik ruang konfigurasi atau topologi. Algoritma ini menggunakan pencarian heuristik dan pencarian jalur terpendek. Dalam hal pencarian, algoritma ini bekerja dengan baik di berbagai media. Algoritma ini pertama kali dipublikasikan oleh Peter Hart, Nils Nilsson dan Bertram Raphael pada tahun 1968 . Algoritma A\* digunakan karena algoritma ini dapat diubah bentuknya menjadi algoritma pencarian jalur lainnya dengan cara mengevaluasi Node, Vertices, & Edges beserta heuristic yang digunakannya. Dengan kata lain, algoritma A\* dapat dimodifikasi sesuai heuristic yang digunakan sehingga dapat berfungsi secara efisien (Daniel Foead 2021).

Algoritma RAPTOR adalah algoritma pemetaan angkutan umum berbasis ronde yang diciptakan oleh Martin C. Carlisle, Terry Wilson, Jeff Humphries, dan Jason Moore. Algoritma ini beroperasi secara berbeda dari algoritma Dijkstra, memproses setiap rute dalam jaringan satu kali per putaran. Algoritma ini dapat dipercepat dengan aturan pemangkasan dan paralelisasi, serta dapat disesuaikan untuk jalur berjalan kaki tanpa batas di angkutan umum, sehingga memperpanjang sebagian perjalanan sebanyak satu perjalanan dalam setiap putaran. Hal ini menjadikan RAPTOR alat yang ampuh untuk perutean angkutan umum yang efisien (Delling and Pajor, 2012).

Penelitian terdahulu dengan judul “Analysis of Dijkstra’s Algorithm and A\* Algorithm in Shortest Path Problem” oleh Dian Rachmawati dan Lysander Gustin membandingkan kinerja Algoritma Dijkstra dan Algoritma A\* dalam

menyelesaikan permasalahan jalur terpendek. Penulis menemukan bahwa kedua algoritma memiliki kinerja yang serupa ketika diterapkan pada peta skala kota atau regional. Namun Algoritma A\* mengungguli Algoritma Dijkstra ketika digunakan untuk menyelesaikan peta skala besar. Hal ini menunjukkan bahwa pilihan algoritma dapat berdampak signifikan terhadap efisiensi penyelesaian masalah jalur terpendek, terutama dalam konteks skala besar. (Dian Rachmawati 2020).

Penelitian terdahulu lainnya oleh Esat Dere dan Akif Durdu dengan judul “Usage of the A\* Algorithm to Find the Shortest Path in Transportation Systems” yang menggunakan peta dari Google Map yang telah di konversi dalam bentuk Grid mengeksplorasi penerapan A\* algoritma dalam sistem transportasi. Penulis mendemonstrasikan bagaimana algoritma A\* dapat digunakan untuk menghitung jalur terpendek dalam sistem transportasi umum, seperti rute bus. Algoritma tersebut membuat “daftar tertutup” untuk mencatat area yang telah dievaluasi, “daftar terbuka” untuk mencatat area yang berdekatan dengan area yang telah dievaluasi, dan menghitung jarak yang ditempuh dari “titik awal” dengan perkiraan jarak ke “titik tujuan”. Penelitian ini menyimpulkan bahwa algoritma A\* efisien dan efektif dalam menyelesaikan permasalahan jalur terpendek pada sistem transportasi skala besar. (Esat Dere 2018).

Terakhir, penelitian terdahulu berjudul “Round-Based Public Transit Routing” oleh Delling, Pajor, dan Werneck, memperkenalkan RAPTOR (Round-Based Public Transit Optimized Router), sebuah algoritma baru untuk menghitung semua perjalanan optimal Pareto dalam jaringan angkutan umum yang dinamis. Penulis mendemonstrasikan bahwa RAPTOR menghitung semua perjalanan optimal Pareto antara dua lokasi acak dengan urutan besarnya lebih cepat dibandingkan pendekatan sebelumnya, yang dengan mudah mengoptimisasi aplikasi interaktif. Ini merupakan peningkatan yang signifikan dibandingkan algoritma lain yang menganggap ini sebagai masalah grafik dan menyelesaikannya menggunakan varian algoritme Dijkstra, yang menghasilkan waktu kueri yang tinggi atau solusi yang kurang optimal. Para penulis juga menyoroti bahwa RAPTOR dapat dengan mudah diperluas untuk menangani waktu keberangkatan yang fleksibel atau kriteria tambahan secara bebas, seperti

zona tarif. Hal ini menunjukkan kinerja dan fleksibilitas algoritma RAPTOR yang lebih unggul dibandingkan algoritma lainnya (Delling, Pajor, and Werneck, 2012).

Dari uraian diatas, penulis sebagai masyarakat pengguna angkutan umum dan juga mahasiswa ilmu komputer berencana untuk membuat aplikasi berbasis mobile yang tujuan dasarnya adalah untuk memetakan peta angkutan umum dengan harapan meningkatkan penggunaan angkutan umum oleh masyarakat. Aplikasi ini nantinya juga dibuat dari kombinasi beberapa algoritma, framework & software yang bekerja secara independen dan dapat diperbaharui dikemudian hari seandainya dibutuhkan perbaikan atau pengembangan. Karena seyogianya saya atau kita sebagai peneliti atau pelajar harus selalu terbuka dengan perkembangan teknologi dan zaman, algoritma atau framework yang dianggap baik atau efisien pada saat ini mungkin akan digantikan dengan penemuan lain dimasa depan (fukurabilitas). Oleh karena itu penulis merancang aplikasi ini sedemikian rupa sehingga setiap komponen-nya dapat diubah dan diperbaharui tanpa mempengaruhi kompatibilitas dengan komponen lainnya.

## **1.2. Rumusan Masalah**

Rumusan masalah pada penelitian ini adalah banyaknya jenis angkutan umum dan kurangnya infrastruktur yang ada di Kota Medan, menyebabkan masalah tersendiri bagi para penggunanya yaitu sulitnya menentukan rute transportasi umum yang harus digunakan untuk berpindah dari dua lokasi berbeda di Kota Medan.

Sistem Pendukung Keputusan Berbasis Mobile dengan OpenTripPlanner diharapkan dapat menjadi solusi untuk meningkatkan minat guna angkutan umum, sehingga mengurangi angka kemacetan di Kota Medan.

### 1.3. Batasan Masalah

Adapun beberapa batasan masalah dalam skripsi ini sebagai berikut :

1. Menggunakan sistem *OpenTripPlanner & OpenStreetMap*
2. Aplikasi menggunakan *OpenStreetMap* sebagai tempat penyimpanan rute
3. Referensi data yang digunakan diambil dari Dishub Kota Medan dan juga dari jurnal yang berkaitan dengan pemetaan angkutan umum dan *Shortest-path Algorithm*
4. Penyusunan program hanya ditujukan untuk pemetaan rute angkutan umum di Kota Medan
5. Program dirancang dengan menggunakan *Flutter*
6. Program yang dirancang berbasis *cross-platform*, dapat diakses dari *Android, iOS*, dan website
7. Aplikasi berfungsi secara *online*

### 1.4. Tujuan Penelitian

Tujuan dari penelitian ini yaitu membangun sebuah aplikasi untuk membantu memetakan rute angkutan umum dari dua titik di Kota Medan dan menghasilkan beberapa output optimal sesuai moda transportasi ( Berjalan kaki, Angkot, & Bus dalam kota ) yang di input oleh pengguna.

### 1.5. Manfaat Penelitian

Manfaat dari penelitian ini yaitu sebagai berikut :

1. Memberikan informasi berupa rute angkutan umum di Kota Medan
2. Memberikan pengetahuan tentang pengolahan bentuk data rute angkutan umum menjadi bentuk data yang lebih dapat dibaca oleh komputer
3. Memberikan informasi tentang penerapan *OpenTripPlanner & OpenStreetMap* dalam membantu memetakan rute angkutan umum
4. Dapat dijadikan referensi dalam pengembangan sistem prediksi dengan topik *Shortest-path Algorithm*

## 1.6. Metodologi Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut :

### 1. Studi Literatur

Pada tahap studi pustaka, penelitian dimulai dengan mencari referensi dari berbagai sumber terpercaya dan melakukan peninjauan pustaka melalui buku-buku, jurnal, artikel ilmiah, dan penelitian-penelitian lainnya yang berhubungan dengan Transportasi Umum, *Shortest-path Algorithm*, Algoritma A\*, Algoritma RAPTOR, & OpenTripPlanner

### 2. Pengumpulan Data

Setelah melakukan studi pustaka, maka langkah selanjutnya adalah mengumpulkan data-data yang diperlukan dari Dinas Perhubungan Kota Medan.

### 3. Perancangan Sistem

Analisis dan perancangan sistem dilakukan untuk menentukan metode yang akan digunakan, efisiensi perangkat lunak, & merancang sistem yang mudah dimengerti dan digunakan segala kalangan. Sistem Pendukung Keputusan tersebut nantinya akan diaplikasikan melalui Smartphone.

### 4. Implementasi Sistem

Pada tahap ini, rancangan yang telah dibuat akan diimplementasikan kedalam framework flutter sehingga bisa dijalankan di Smartphone sesuai dengan rancangan.

### 5. Pengujian Sistem

Pada tahap ini, sistem yang sudah selesai dalam bentuk perangkat lunak berbasis mobile akan diuji untuk mengetahui kekurangan atau kegagalan apabila ada.

### 6. Dokumentasi

Pada tahap ini, penelitian yang telah dilakukan, didokumentasikan mulai dari tahap analisa sampai kepada pengujian dalam bentuk skripsi.

## 1.7. Sistematika Penulisan

Beberapa langkah yang diambil untuk menyelesaikan penelitian ini adalah sebagai berikut :

### **BAB 1 PENDAHULUAN**

Bab ini menjelaskan latar belakang dari penulisan skripsi “Sistem Pendukung Keputusan Pemilihan Angkutan Umum Di Kota Medan Berbasis Mobile Menggunakan Opentripplanner”, rumusan dan batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan prosedur penulisan.

### **BAB 2 LANDASAN TEORI**

Bab ini memberikan penjelasan terkait pengertian serta membahas sistem transportasi angkutan umum, kemacetan, sistem pendukung keputusan, sistem *database OpenStreetMap*, Sistem Framework *OpenTripPlanner*, Format *GTFS (General Transit Feed Specification)*, dan Sistem Framework *Flutter*.

### **BAB 3 ANALISIS DAN PERANCANGAN SISTEM**

Bab ini akan membahas analisis masalah yang diteliti dan perancangan sistem yang akan dibuat.

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM**

Dalam bab ini, pembahasan tentang implementasi dan pelaksanaan perancangan sistem, serta hasil dari analisis dan perancangan yang telah dibuat, dibahas. Sistem yang telah dibangun akan diuji untuk menunjukkan hasilnya.

### **BAB 5 KESIMPULAN DAN SARAN**

Bab terakhir dari penelitian ini berisi kesimpulan dari bab-bab sebelumnya, yang sebagian besar berdasarkan bab-bab sebelumnya. Bab ini juga mencakup rekomendasi yang mendukung hasil dari percobaan dan

pengujian yang dilakukan pada sistem yang telah dibangun. Saran-saran ini diharapkan dapat bermanfaat bagi pembaca dan pengembang.

## BAB 2

### LANDASAN TEORI

#### 2.1. Sistem Transportasi Angkutan Umum

Sistem Transportasi Angkutan Umum adalah sistem yang mengatur layanan yang disediakan oleh lembaga publik atau swasta yang tersedia untuk semua orang yang membayar tarif yang ditentukan. Di daerah perkotaan, sistem transportasi umum yang umum adalah bus, troli, LRT, metro, kereta api regional dan moda lainnya, yang beroperasi pada jalur atau rute yang ditentukan jadwal yang ditetapkan dan diumumkan (Vukan R. Vuchic 2002).

Lembaga yang mengatur sistem ini di Indonesia adalah Kementerian Perhubungan Republik Indonesia. Sumber data yang akan digunakan pada penelitian ini berasal dari Dinas Perhubungan Kota Medan. Berikut adalah contoh dari data yang diberikan Dinas Perhubungan Kota Medan.

##### KOPERASI PENGANGKUTAN UMUM MEDAN

###### Trayek 02

Keluar : Term.Amplas – Jl.Rivai A.Manaf– Jl. Menteng VII – Jl. P. Denai – Jl. AR. Hakim – Jl. Halat – Jl. SM. Raja – Jl. Sakti Lubis – Jl. B. Katamso – Jl. Karya Jaya – Jl. Karya Jasa – Jl. Pintu Air – Sp. Pos – Jl. Jamin Ginting - Jl. Iskandar Muda – Jl. Gajah Mada – Jl. S. Parman – Jl. Glugur – Jl.G. Subroto – Jl. Ayahanda – Jl. Pabrik Tenun – Jl. Sekip – Jl. Gereja – Jl. Karya – Jl. Kapt. Sumarsono – Jl. Pertempuran – Jl. KL. Yos Sudarso – Rumah Potong Hewan

Masuk : Rumah Potong Hewan – Jl. KL. Yos Sudarso – Jl. Pertempuran – Jl. Kapt. Sumarsono – Jl. Karya – Jl. Gereja – Jl. Sekip – Jl. Sikambing – Jl. Adam Malik – Jl.G.Subroto – Jl. Iskandar Muda – Jl. Jamin Ginting – Sp. Pos – Jl. Pintu Air – Jl. Karya Jasa – Jl. B. Katamso – Jl. Sakti Lubis – Jl. SM. Raja – Jl. Halat – Jl. AR. Hakim – Jl. Denai – Jl. Menteng VII – Jl.Rivai A.Manaf– Term.Amplas

###### Trayek 03

Keluar : Sp.IKIP/Sp.Pasar V /Jl. Williem Iskandar/Batas Kota Medan – Jl. Perjuangan – Jl. Rakyat – Jl. Purwo – Jl. P. Kemerdekaan – Jl. Prof. HM. Yamin, SH – Jl. Timor – Jl. Veteran – Jl. Bangka – Jl. Riau – Jl. Karimun – Term.Sambu – Jl. Pandan – Jl. Haryono MT – Jl.Surakarta – Jl.Bandung – Jl.Cirebon – Jl.SM.Raja – Jl.Rivai A.Manaf -Term.Amplas – Jl.Rivai A.Manaf- Jl.SM.Raja - Batas Kota Medan

Masuk : Jl.SM.Raja/Batas Kota Medan – Jl.SM.Raja – Jl.Rivai A.Manaf - Term. Amplas – Jl.Rivai A.Manaf - Jl.SM.Raja – Jl.Rahmadsyah – Jl.Sutomo – Jl.P. Kemerdekaan – Jl.Purwo – Jl.Rakyat – Jl.Perjuangan – Jl.Williem Iskandar – Jl.Williem Iskandar/ SP.IKIP/SP.Pasar V/Batas kota Medan

**Gambar 2.1. Contoh Rute**



## 2.2. Kemacetan

Kemacetan adalah kondisi dimana arus lalu lintas melambat di bawah kecepatan yang dapat diterima. Hal ini disebabkan karena jumlah mobil yang mencoba menggunakan jalan tersebut lebih banyak daripada kapasitas jaringan lalu lintas. Apabila kepadatan dan kecepatan lalu lintas pada suatu sarana transportasi tinggi dibandingkan dengan kondisi acuan yang mempunyai kepadatan rendah dan kecepatan tinggi maka dikatakan dalam keadaan macet (Aftabuzzaman 2007).

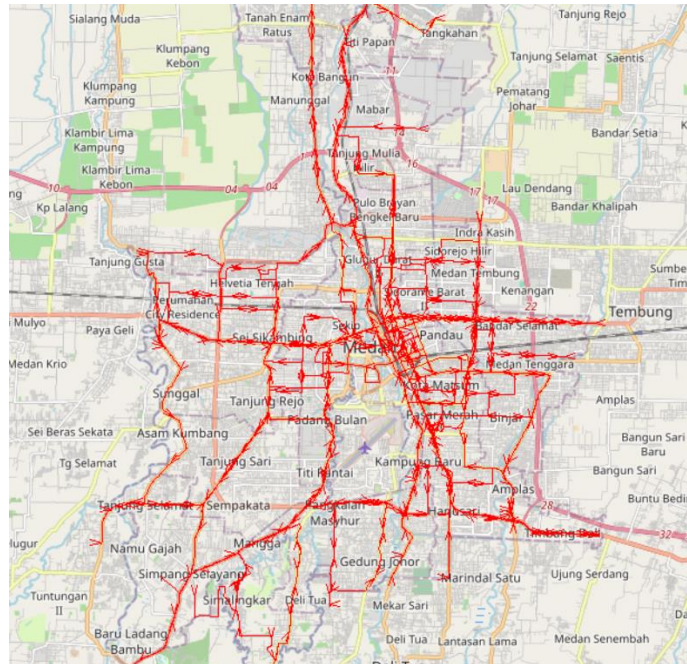
## 2.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem informasi berbasis komputer yang membantu pengambilan keputusan perusahaan atau organisasi. Sistem Pendukung Keputusan (SPK) membantu tingkat manajemen, operasi, dan perencanaan suatu organisasi dalam mengambil keputusan yang mungkin berubah dengan cepat dan sulit untuk ditentukan sebelumnya. Sistem berbasis pengetahuan adalah bagian dari SPK. Ketika SPK dikembangkan dengan benar, SPK merupakan sistem perangkat lunak interaktif yang dimaksudkan untuk membantu pengambil keputusan dalam mengumpulkan data yang relevan dari berbagai sumber, termasuk dokumen, data yang belum diproses, pengalaman pribadi, dan model bisnis, untuk mengidentifikasi dan mengatasi masalah. dan

## 2.4. Sistem Database *OpenStreetMap*

*OpenStreetMap* adalah database geografis *open-source* dan gratis buatan relawan yang paling terkenal di dunia, *OpenStreetMap* (OSM) banyak digunakan sebagai ilustrasi Volunteered Geographic Information (VGI). Sejak peluncurannya pada tahun 2004, OSM telah mengumpulkan lebih dari 8 juta pengguna terdaftar pada tahun 2022. Pada saat ini banyak situs web dan aplikasi, seperti Facebook, Craigslist, Foursquare, dan Strava menggunakan peta dasar OSM sebagai peta referensi untuk melapisi informasi mereka atau menggunakan data OSM. OSM telah berkontribusi secara signifikan terhadap berbagai skenario bencana lingkungan dan kemanusiaan (Alison Moore 2022).

Data rute yang sebelumnya sudah didapatkan dari Dinas Perhubungan Kota Medan di-*tracing* secara *manual* dan diubah formatnya menjadi format yang dapat diproses dan disimpan di OpenStreetMap. Berikut gambar dari hasil dari data yang sudah diproses.



**Gambar 2.2.** Database Rute yang sudah di-*input*

## 2.5. Format *GTFS* (*General Transit Feed Specification*)

*GTFS*, singkatan dari *General Transit Feed Specification*, adalah format standar yang dirancang oleh Google pada tahun 2005 untuk menyusun jadwal transportasi umum dan data geografis yang relevan. Spesifikasi ini berfungsi sebagai kerangka terstruktur untuk mengkonsolidasikan beragam informasi transit, termasuk rute, pemberhentian, jadwal, dan tarif berbagai moda transportasi umum seperti bus, kereta bawah tanah, kereta api, dan feri.

Dengan menyediakan format seragam untuk data transit, *GTFS* memungkinkan pengembang mengakses dan mengintegrasikan informasi ini ke dalam aplikasi mereka dengan lancar. Integrasi ini memfasilitasi pengembangan layanan dan aplikasi terkait transit yang bertujuan membantu pengguna dalam perencanaan perjalanan dan navigasi melalui sistem transportasi umum. Khususnya, *GTFS* telah diadopsi secara luas di kalangan perusahaan angkutan

umum secara global, dan banyak dari mereka mempublikasikan datanya dalam format *GTFS* untuk aksesibilitas yang lebih luas.

Standardisasi ini telah memicu munculnya beragam aplikasi dan layanan yang berpusat pada transportasi umum yang memanfaatkan data *GTFS*. Aplikasi ini menawarkan fungsionalitas seperti pembaruan angkutan umum secara real-time, perencanaan perjalanan, dan optimalisasi rute, sehingga meningkatkan pengalaman pengguna secara keseluruhan dalam menavigasi jaringan angkutan umum (Google, 2005).

Berikut merupakan format dari *file-file GTFS* beserta *Fields* yang diperlukan. Data rute yang sebelumnya telah disimpan di database *OpenStreetMap* diubah kembali formatnya menjadi format *GTFS*.

**Tabel 2.1.** Format *file-file GTFS*

GTFS Files	Required Fields
agency.txt	agency_id, agency_name, agency_url, agency_timezone
stops.txt	stop_id, stop_name, stop_lat, stop_lon
routes.txt	route_id, route_short_name, route_long_name, route_type
trips.txt	route_id, service_id, trip_id, trip_headsign
stop_times.txt	trip_id, arrival_time, departure_time, stop_id, stop_sequence
calendar.txt	service_id, monday, tuesday, wednesday, thursday, friday, saturday, sunday, start_date, end_date
calendar_dates.txt	service_id, date, exception_type (1 for added, 2 for removed)

## 2.6.Sistem Framework OpenTripPlanner

*OpenTripPlanner* adalah perencana rute multi-modal *open-source* dan *cross-platform* yang ditulis dalam bahasa *Java*. Sistem ini menggunakan data *Open Street Map (OSM)* yang diimpor untuk perutean pada jaringan jalan dan jalur serta mendukung sistem perutean angkutan umum *multi-modal* melalui *feed GTFS* yang diimpor. Sistem ini juga dapat menerapkan model elevasi digital pada jaringan jalan *OSM*, sehingga memungkinkan, misalnya, permintaan rute ramah sepeda.

*Framework OpenTripPlanner* ini juga masih berkembang hingga saat penulisan penelitian ini, dan versi yang terbaru saat ini adalah versi 2.4. Secara garis besar, penggunaan *framework* ini dibagi menjadi 2, OTP1 (Versi 1.x) & OTP2 (Versi 2.x). Kedua versi tersebut memiliki syarat, keunggulan, dan kelemahannya masing-masing. Berikut adalah table yang menjelaskan perbedaan fitur dan syarat dari kedua versi tersebut (OpenTripPlanner, n.d.).

**Tabel 2.2.** Perbedaan Fitur OTP1 & OTP2

Fitur	OTP 1.x	OTP 2.x
Algoritma Perutean	<i>Generalized cost A*</i>	<i>Generalized cost A* &amp; Multi-criteria range-RAPTOR</i>
Modularitas	Arsitektur monolitik	Arsitektur modular, memungkinkan untuk penyesuaian lebih mudah
Format Data	Mendukung berbagai format data tetapi mungkin memerlukan pemrosesan tambahan	Dukungan yang ditingkatkan untuk berbagai format data termasuk <i>GTFS</i> , <i>GTFS-Realtime</i> , dan <i>OpenStreetMap</i>
Integrasi Data Real-time	Dukungan terbatas untuk integrasi data <i>real-time</i>	Dukungan yang ditingkatkan untuk data real-time, termasuk <i>GTFS-Realtime</i> , membuat pembaruan real-time lebih efisien

Fitur	OTP 1.x	OTP 2.x
Aksesibilitas	Fitur aksesibilitas dasar	Fitur aksesibilitas yang ditingkatkan dan opsi yang lebih banyak
Kinerja	Kinerja yang cukup baik tetapi mungkin menurun dengan dataset besar	Kinerja yang ditingkatkan, terutama dengan dataset besar
Skalabilitas	Skalabilitas terbatas karena arsitektur monolitik	Skalabilitas yang ditingkatkan karena desain modular
Kustomisasi	Opsi kustomisasi terbatas	Kemampuan kustomisasi yang meningkat melalui arsitektur modular dan dukungan <i>API</i> yang ekstensif

**Tabel 2.3.** Perbedaan Syarat Penggunaan OTP1 & OTP2

Persyaratan GTFS	OTP 1.x	OTP 2.x
Dukungan <i>GTFS</i>	Dukungan <i>GTFS</i> dasar dengan fungsionalitas terbatas	Dukungan <i>GTFS</i> yang ditingkatkan dengan kemampuan parsing yang lebih baik
Dukungan <i>GTFS-Realtime</i>	Dukungan terbatas atau tidak ada untuk <i>GTFS-Realtime</i>	Dukungan yang ditingkatkan untuk feed <i>GTFS-Realtime</i>
Pembaruan Jadwal	Data jadwal statis digunakan untuk perencanaan rute	Pembaruan jadwal dinamis dimungkinkan dengan <i>GTFS-Realtime</i>
Validasi Data	Kemampuan validasi data terbatas	Validasi data yang ditingkatkan memastikan kepatuhan <i>GTFS</i>
Kompatibilitas	Kompatibilitas dengan versi data <i>GTFS</i> hingga sejauh tertentu tergantung pada implementasinya	Kompatibilitas yang lebih baik dengan berbagai versi data <i>GTFS</i> , termasuk iterasi terbaru

Persyaratan GTFS	OTP 1.x	OTP 2.x
Integrasi Data	Membutuhkan integrasi dan pemrosesan data <i>GTFS</i> secara manual sebelum digunakan dengan <i>OTP</i>	Mekanisme integrasi data yang ditingkatkan untuk penggunaan data <i>GTFS</i> yang lebih lancar dengan <i>OTP</i>
Pembaruan Real-time	Pembaruan <i>real-time</i> mungkin memerlukan konfigurasi dan penyiapan tambahan	Pembaruan <i>real-time</i> terintegrasi lebih lancar, memungkinkan penyesuaian rute dinamis berdasarkan data langsung

Berdasarkan syarat diatas, data rute yang penulis dapatkan dari Dinas Perhubungan Kota Medan tidak memenuhi untuk menggunakan OTP2, oleh karena itu pada penelitian ini penulis akan menggunakan OTP1. Algoritma perutean OTP1 menggunakan algoritma *Generalized cost A\**, algoritma ini menggunakan algoritma *A\** yang dimodifikasi agar menggeneralisasi *cost* antara beberapa jenis moda transportasi. Berikut merupakan rumus dari algoritma *A\** dan algoritma *Generalized cost A\** yang akan digunakan dalam penelitian ini.

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

Dengan :

$f(n)$  = Total biaya dari node, yang merupakan jumlah dari biaya dari lintasan dari node awal ke node n ( $g(n)$ ) dan perkiraan heuristik dari biaya dari node n ke node tujuan ( $h(n)$ ).

$g(n)$  = Biaya dari lintasan dari node awal ke node n. Ini mewakili biaya aktual yang dikeluarkan sejauh ini untuk mencapai node n dari node awal.

$h(n)$  = Perkiraan heuristik dari biaya dari node n ke node tujuan. Ini adalah perkiraan dari biaya aktual dari node n ke node tujuan. Heuristik ini membantu dalam memandu pencarian menuju node tujuan dengan efisien.

$$f(n) = Transfers \times 0.65 + Walk Distance \times 0.3 \\ + ( Total Distance / 100 ) \times 0.05$$

Dengan :

$f(n)$  = Total biaya dari node awal hingga node akhir.

*Transfers* = Jumlah perubahan moda transportasi yang dilakukan selama menempuh lintasan dari node awal hingga node akhir.

*Walk Distance (m)* = Jumlah lintasan yang ditempuh dengan berjalan kaki selama menempuh lintasan dari node awal hingga node akhir.

*Total Distance (m)* = Total jarak lintasan yang ditempuh selama menempuh lintasan dari node awal hingga node akhir.

Perlu diingat bahwa framework ini menggunakan database OpenStreetMap sebagai node & vertices dalam algoritma yang akan digunakan. Dan tujuan dari penyesuaian atau generalisasi ini adalah karena framework ini dirancang secara khusus untuk penggunaan transportasi umum (Transit), sehingga jarak atau biaya (  $f(n)$  ) yang sebelumnya dihasilkan pada algoritma A\* tidak dapat digunakan karena algoritma tersebut mengasumsikan biaya dari jenis moda transportasi apapun menjadi sama, sehingga hanya menghasilkan jarak yang terdekat. Oleh karena itu, *output* dari algoritma *Generalized cost A\** yang dihasilkan *framework* ini mungkin saja menghasilkan jarak yang lebih panjang namun biaya yang lebih sedikit.

## 2.7. REST API

Representational State Transfer (REST) adalah gaya arsitektur yang banyak digunakan untuk mendesain API web. Berasal dari proposal Fielding lebih dari dua dekade lalu, REST telah mengalami pertumbuhan dan evolusi yang substansial. Tujuan mendasarnya adalah untuk memfasilitasi evolusi independen dan pemeliharaan komponen klien dan server, sehingga meningkatkan keandalan layanan dibandingkan dengan pendekatan yang memerlukan pengembangan tersinkronisasi dari kedua sisi antarmuka. Terlepas dari manfaatnya, desain API yang buruk dalam arsitektur REST dapat menyebabkan kerentanan keamanan dan ketidakpuasan pengguna, seperti yang disoroti oleh kekhawatiran industri dan munculnya layanan yang didedikasikan

untuk memantau kesalahan API. Meskipun banyak rekomendasi desain, bukti empiris mengenai kemanjurannya bagi pengguna REST API masih langka, dan alat untuk menegakkan rekomendasi ini masih kurang.

REST API mewarisi persyaratan struktural dari standar HTTP dan tidak bergantung pada bahasa, memungkinkan klien dan server diimplementasikan dalam berbagai bahasa pemrograman seperti JavaScript, TypeScript, Python, Ruby, dan Go. Pembagian antara klien dan server membawa implikasi signifikan terhadap kemampuan modifikasi, kemampuan berevolusi, dan kinerja, karena organisasi yang menghosting server mengeluarkan biaya hosting. Akibatnya, memprediksi alat yang paling efektif untuk pengembang REST API dan memilih praktik yang kondusif untuk membangun aplikasi klien yang andal menimbulkan tantangan. Meskipun terdapat banyak pengetahuan tentang desain API yang baik secara umum, menerjemahkan prinsip-prinsip ini ke dalam desain REST API yang efektif memerlukan pertimbangan yang cermat karena persyaratan strukturalnya yang unik dan kemandirian bahasa (Coblenz et al., n.d.).

Pada aplikasi ini REST API akan digunakan sebagai alat komunikasi antara perangkat *User* dengan *Backend Server* dari *Framework OpenTripPlanner*. Berikut adalah contoh penggunaan REST API pada aplikasi ini.

```

    TruifLocation from,
    TruifLocation to,
    List<TransportMode> transportModes,
) async {
    final Uri request = Uri.parse(
        otpEndpoint + planPath,
    ).replace(queryParameters: {
        "fromPlace": from.toString(),
        "toPlace": to.toString(),
        "date": _todayMonthDayYear(),
        "time": '12:00:00',
        "numItineraries": "7",
        "maxWalkDistance": "1000",
        "mode": _parseTransportModes(transportModes),
    });
    final response = await _fetchRequest(request);
    if (response.statusCode == 200) {
        return compute(_parsePlan, utf8.decode(response
    ) else {
        throw FetchOnlineResponseException('Server Err

```

**Gambar 2.3.** Contoh Penggunaan REST API



Berikut adalah beberapa contoh parameter yang digunakan dalam komunikasi ini, parameter ini juga dapat digunakan untuk menyesuaikan *cost* atau biaya dari *output* yang dikembalikan *OpenTripPlanner* sesuai keperluan pengembang aplikasi. Pada penelitian ini penulis menambahkan parameter “numItineraries : 7” , “maxWalkDistance : 1000”, & “mode : \_parseTransportModes(transportModes)”. Fungsi dari ketiga parameter ini adalah membatasi jumlah *output* rute menjadi 7, membatasi jarak berjalan kaki menjadi 1500 meter, dan hanya menggunakan moda transportasi berjalan kaki & angkot.

**Tabel 2.3.** Parameter *OpenTripPlanner*

Parameter	Deskripsi	Penggunaan
routerId	Identifikasi untuk instansi router tertentu	Digunakan untuk menentukan instansi router yang digunakan untuk perencanaan perjalanan
fromPlace	Titik awal perjalanan	Menentukan lokasi awal untuk perencanaan perjalanan
toPlace	Tujuan perjalanan	Menentukan lokasi tujuan untuk perencanaan perjalanan
date	Tanggal perjalanan	Menetapkan tanggal untuk perencanaan perjalanan
time	Waktu perjalanan	Menetapkan waktu untuk perencanaan perjalanan
arriveBy	Kendala waktu tiba	Menentukan apakah perjalanan harus sampai pada waktu tertentu
departBy	Kendala waktu keberangkatan	Menentukan apakah perjalanan harus berangkat pada waktu tertentu

Parameter	Deskripsi	Penggunaan
mode	Preferensi mode perjalanan	Menentukan mode transportasi yang akan digunakan
maxWalkDistance	Jarak maksimum untuk berjalan kaki	Menetapkan jarak maksimum yang pengguna bersedia tempuh dengan berjalan kaki
numItineraries	Jumlah opsi perjalanan yang akan dihasilkan	Menentukan jumlah opsi perjalanan yang akan dikembalikan
wheelchair	Preferensi aksesibilitas kursi roda	Menentukan apakah perjalanan harus mempertimbangkan aksesibilitas kursi roda
bikeSafe	Preferensi keselamatan sepeda	Menentukan preferensi untuk rute yang ramah sepeda
bikeTrafficStress	Preferensi stres lalu lintas sepeda	Menentukan tingkat toleransi stres lalu lintas untuk bersepeda
bikeSpeed	Kecepatan rata-rata bersepeda	Menetapkan kecepatan rata-rata bersepeda
maxTransfers	Jumlah maksimum transfer yang diizinkan	Membatasi jumlah transfer dalam opsi perjalanan

## 2.8. Sistem Framework Flutter

*Flutter* adalah sebuah *tools* atau "alat" yang memungkinkan *developer* menggunakan satu bahasa pemrograman dan basis kode untuk membuat aplikasi *cross-platform* (iOS, Android, Web). *Flutter* adalah *framework open-source* dan gratis yang pertama kali dibuat oleh Google. *Flutter* adalah software development kit (SDK) yang bersifat *cross-platform* dan dapat digunakan untuk membuat aplikasi seluler untuk platform utama seperti Windows, Linux, Mac, iOS, dan Android. Meskipun merupakan alat atau *framework* baru yang dirilis pada Mei 2017, *flutter* dengan cepat muncul sebagai pilihan utama bagi pengembang hibrida atau *cross-platform* karena kemudahan pengembangan dan penerapan aplikasi, serta perpustakaan UI-nya yang luas (Kurale & Bala 2023).

## BAB 3

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Analisis Sistem

Analisis sistem adalah tahap awal dalam pengembangan sistem yang bertujuan untuk memahami tujuan dan kebutuhan sistem serta mengidentifikasi masalah yang mungkin muncul. Analisis sistem juga membantu dalam memecahkan masalah dengan membagi sistem menjadi bagian-bagian yang lebih terdefinisi dan memuda.

Oleh karena itu, analisis sistem adalah bagian penting dari proses pengembangan sistem. Ini memungkinkan pemahaman yang lebih baik tentang kebutuhan dan kebutuhan pengguna sistem serta memudahkan proses perancangan dengan membaginya menjadi bagian-bagian yang lebih dapat dikelola.

#### 3.2. Analisis Permasalahan

Analisis masalah adalah proses yang dilakukan untuk menemukan masalah dan mencari solusinya. Pada penelitian ini akan dibahas bagaimana proses dan kinerja dari *framework OpenTripPlanner* dalam menghasilkan rute optimal dari dua titik yang di-*input* oleh *user* pada aplikasi *mobile*.

Aplikasi ini serupa dengan aplikasi navigasi lainnya dimana aplikasi akan menghasilkan beberapa *output* agar *user* dapat memilih rute mana yang lebih diminati. Alasan penulis merancang aplikasi ini untuk menghasilkan beberapa *output* adalah karena keberagaman preferensi setiap orang atau *user*. Meskipun sebelumnya telah dibuat penyesuaian *cost* untuk setiap jenis moda transportasi, masih terdapat faktor-faktor atau variabel-variabel lain dari sisi *user* yang harus dibuat menjadi pertimbangan. Contohnya seperti biaya, kesehatan atau stamina, dan lain-lain. Maksudnya adalah, sebagian besar *user* mungkin sudah setuju dan mengikuti *output* dari aplikasi ini, namun mungkin ada juga beberapa *user* yang mempunyai kondisi atau keterbatasan, contohnya seperti biaya transfer atau pergantian mode transportasi, atau keterbatasan fisik dalam berjalan kaki. Oleh karena itu aplikasi ini akan menghasilkan beberapa *output*.

### 3.3. Analisis Kebutuhan Sistem

Analisis kebutuhan adalah analisis yang dilakukan untuk mengetahui apa yang diperlukan untuk membangun aplikasi yang diinginkan. Analisis kebutuhan ini dibagi menjadi dua, yaitu:

#### 3.3.1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang mengandung informasi dan proses yang diperlukan sistem, seperti input, proses, output, dan penyimpanan data. Adapun kebutuhan-kebutuhan tersebut adalah sebagai berikut

- a. Sistem mampu memberikan informasi rute seluruh trayek yang tersedia
- b. Sistem mampu menerima *input user* berupa dua titik koordinat
- c. Sistem mampu memberikan *output* berupa rute-rute optimal yang dapat ditempuh dari kedua titik yang di-*input user*
- d. Sistem mampu menampilkan rute-rute tersebut secara berurut sesuai *cost* yang dihasilkan

#### 3.3.2. Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tak mempengaruhi fungsionalitas sistem atau aplikasi secara langsung, seperti tampilan aplikasi, kontrol, dan lain-lain. Adapun kebutuhan-kebutuhan tersebut adalah sebagai berikut

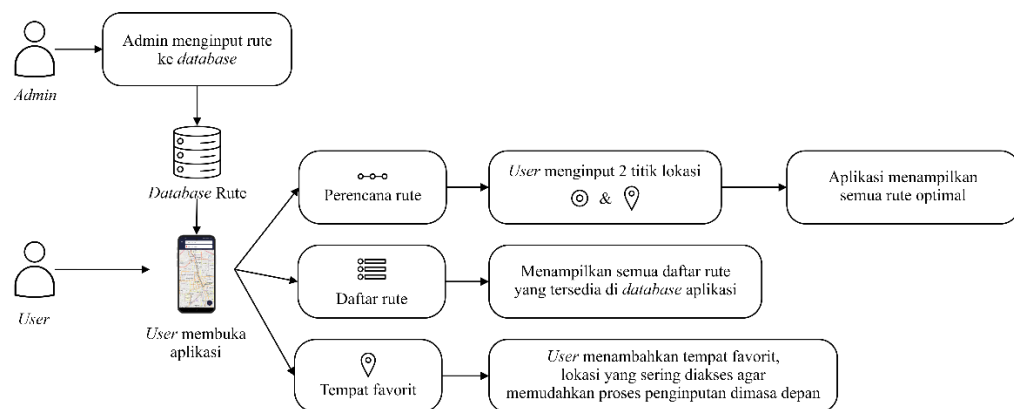
- a. Beberapa fitur aplikasi masih dapat digunakan dalam keadaan *offline*
- b. Aplikasi mudah dioperasikan oleh berbagai kalangan usia
- c. Tampilan dan menu aplikasi bersifat *user friendly* sehingga tidak membingungkan pengguna aplikasi
- d. Pergantian moda transportasi ditandai oleh perbedaan warna rute sehingga memudahkan pengguna untuk menentukan pilihan rute

### 3.4. Perancangan Sistem

Dalam proses merancang sistem, terdapat langkah-langkah yang dilakukan untuk memudahkan pemodelan sistem yang akan dibangun. Langkah ini bertujuan untuk menggambarkan kondisi serta komponen-komponen yang akan berperan dalam sistem tersebut. Proses pemodelan sistem ini meliputi perancangan Arsitektur Sistem, perancangan *Flowchart*, perancangan *Use-Case Diagram*, perancangan *Activity Diagram*, dan perancangan *Sequence Diagram*.

#### 3.4.1. Perancangan Arsitektur Sistem

Perancangan Arsitektur Sistem adalah suatu bentuk perencanaan yang menggambarkan proses, alur, dan interaksi antara komponen-komponen dalam sistem. Seluruh rancangan tersebut diilustrasikan dalam arsitektur sistem yang dijelaskan dalam Gambar 3.1.

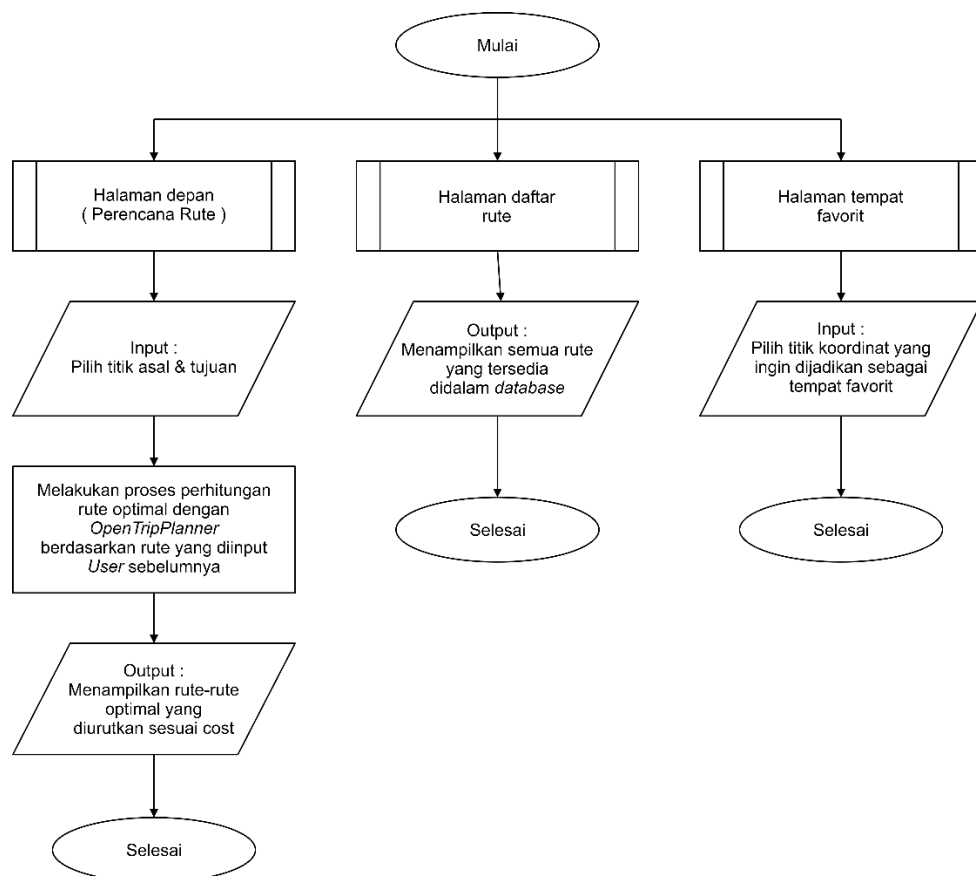


**Gambar 3.1.** Diagram Arsitektur Sistem

Pada gambar 3.1. terlihat bahwa tiga fitur utama dari sistem ini adalah Perencana Rute, Daftar Rute, & Tempat Favorit. Fitur yang ditampilkan secara *default* pada halaman depan adalah Perencana Rute, dan fitur lainnya dapat diakses melalui *side navigation bar*. Fitur yang menggunakan framework *OpenTripPlanner* adalah Perencana Rute, & Daftar Rute, sementara fitur Tempat Favorit bekerja secara lokal di perangkat User.

### 3.4.2. Perancangan Flowchart

Dalam perancangan sistem, diperlukan *Flowchart* yang menjelaskan cara kerja sistem secara rinci. *Flowchart* digunakan untuk mengilustrasikan setiap langkah dan keputusan yang terjadi dalam proses yang akan diimplementasikan dalam sebuah program. Keseluruhan rancangan sistem akan diuraikan melalui *Flowchart* yang terdapat dalam Gambar 3.2.

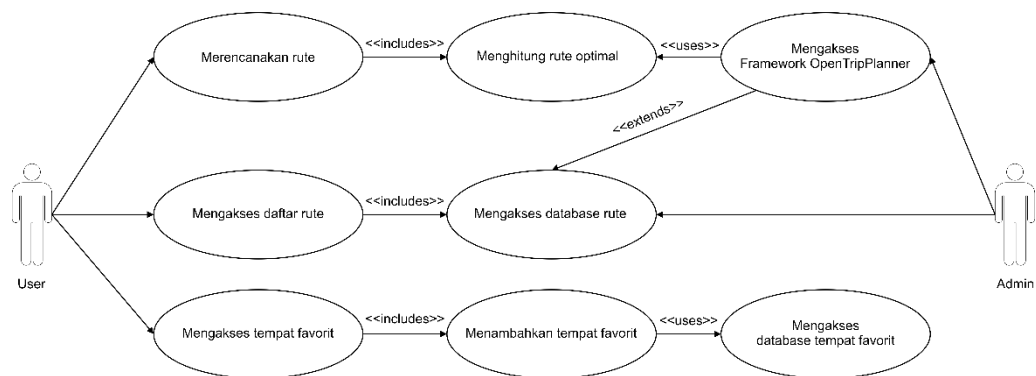


**Gambar 3.2.** *Flowchart* Sistem

Pada gambar 3.2. dijelaskan bahwa terdapat tiga *predefined process* atau tiga halaman utama dalam sistem ini, yaitu Perencana Route sebagai Halaman utama, Halaman Daftar Rute, & Halaman Tempat Favorit. Seperti yang dapat dilihat, alur kerja utama dari sistem ini adalah Halaman depan (Perencana Route), dua komponen lainnya merupakan komponen pendukung untuk memudahkan akses halaman depan tersebut.

### 3.4.3. Use-case Diagram

Diagram yang diperlukan untuk mengilustrasikan fungsi-fungsi yang diharapkan ada dalam sistem yang akan dibangun disebut sebagai *Use-Case Diagram*. *Use-Case Diagram* adalah representasi visual yang menggambarkan perilaku atau kebiasaan sistem. Dalam kata lain, diagram ini digunakan untuk menunjukkan interaksi antara aktor (*User* atau sistem lainnya) dengan sistem.



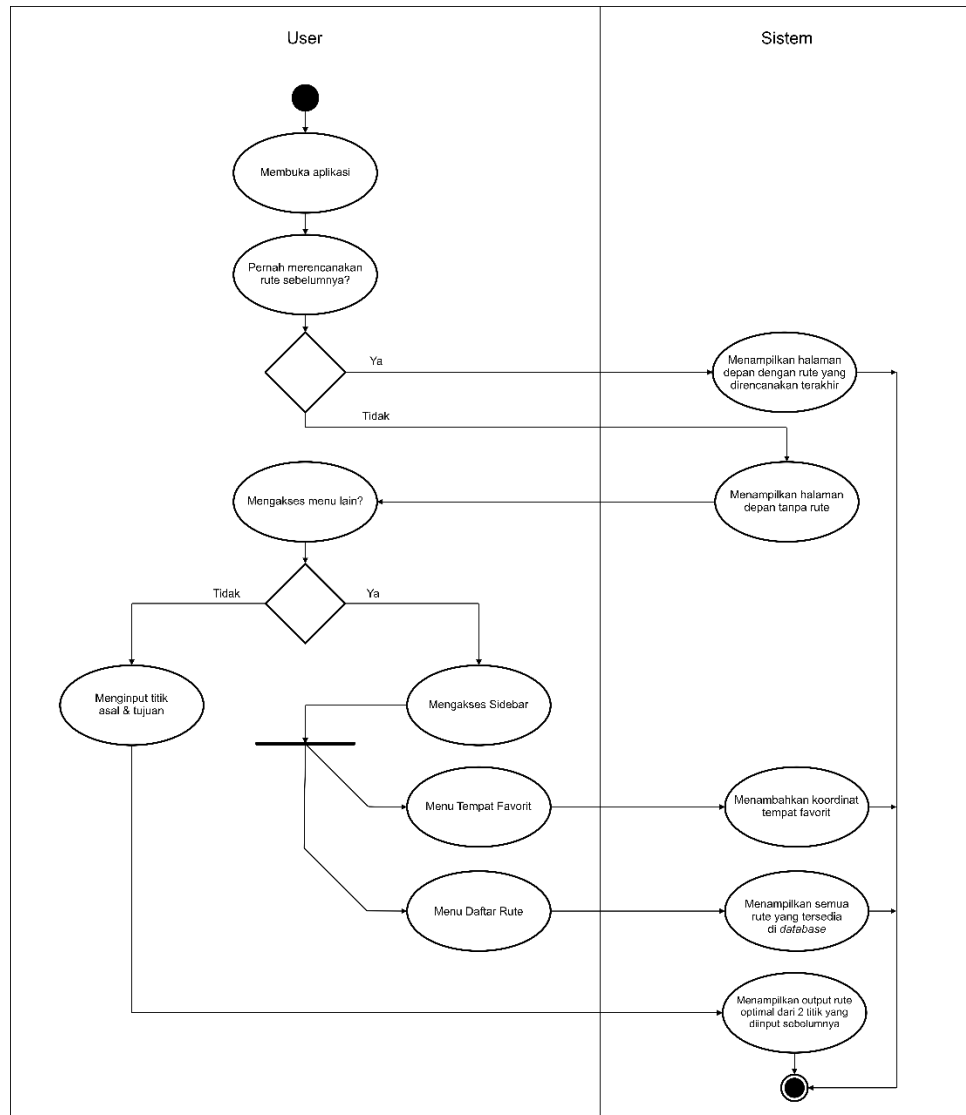
**Gambar 3.3.** Use Case Diagram

Pada gambar 3.3. terlihat bahwa terdapat dua aktor pada sistem ini. Aktor tersebut adalah *User* dan *Admin*. Pada gambar tersebut dijelaskan juga bahwa kedua aktor tersebut tidak terlibat secara langsung. Pada kasus *User*, *User* diberikan opsi untuk memilih salah satu fitur yang sudah disediakan. Sementara pada kasus *Admin*, *Admin* dapat melakukan penginputan database rute, dan juga melakukan penyesuaian pada *Framework OpenTripPlanner*, penyesuaian ini dapat berupa peningkatan versi, penambahan *parameter*, dan sebagainya. Sehingga, meskipun *User* & *Admin* tidak terlibat secara langsung, proses atau cara bekerja dari fitur-fitur yang dapat diakses oleh *User* diatur, diawasi, dan dipelihara secara langsung oleh *Admin*.

### 3.4.4. Activity Diagram

*Activity Diagram* adalah representasi visual yang menggambarkan berbagai aktivitas yang dilakukan oleh sistem yang akan dibangun. Diagram ini menjelaskan bagaimana setiap alur dimulai, pilihan yang mungkin terjadi, dan cara alur tersebut berakhir. Dalam sistem yang akan dibuat, terdapat dua jenis

*Activity Diagram* yang digunakan: satu berdasarkan interaksi pengguna atau *User*, dan yang lainnya berdasarkan interaksi admin.



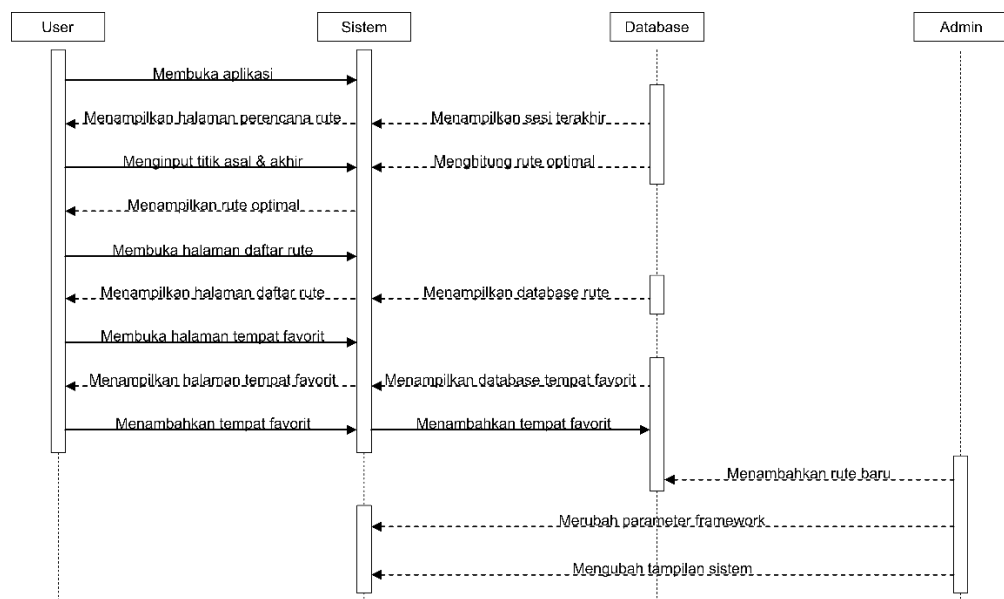
**Gambar 3.4.** Activity Diagram

Pada gambar 3.4. dijelaskan alur aktivitas dari sistem ini. Sistem ini dirancang sedemikian rupa sehingga sistem dapat menyimpan sesi terakhir dalam penggunaan aplikasi ini, dan sesi tersebut akan menjadi halaman depan dalam penggunaan aplikasi selanjutnya.



### 3.4.5. Sequence Diagram

Pembuatan *Sequence Diagram* adalah suatu langkah penting dalam perancangan karena diagram ini menjadi panduan untuk setiap tahap pemrograman dan mencakup alur kontrol dari sistem yang sedang dibangun. Proses pembuatan diagram urutan membutuhkan waktu yang cukup lama agar menghasilkan desain yang baik dan terstruktur.



**Gambar 3.5.** Sequence Diagram

Gambar 3.5. mengilustrasikan interaksi antara objek-objek yang bekerja pada sistem ini. Ilustrasi tersebut menjelaskan interaksi tersebut dengan menggunakan anak panah. Pesan yang dikirimkan ditandai dengan anak panah dari kiri ke kanan, dan pesan balasan ditandai dengan anak panah putus-putus dari kanan ke kiri.

### 3.5. Perancangan Antarmuka Sistem (Interface)

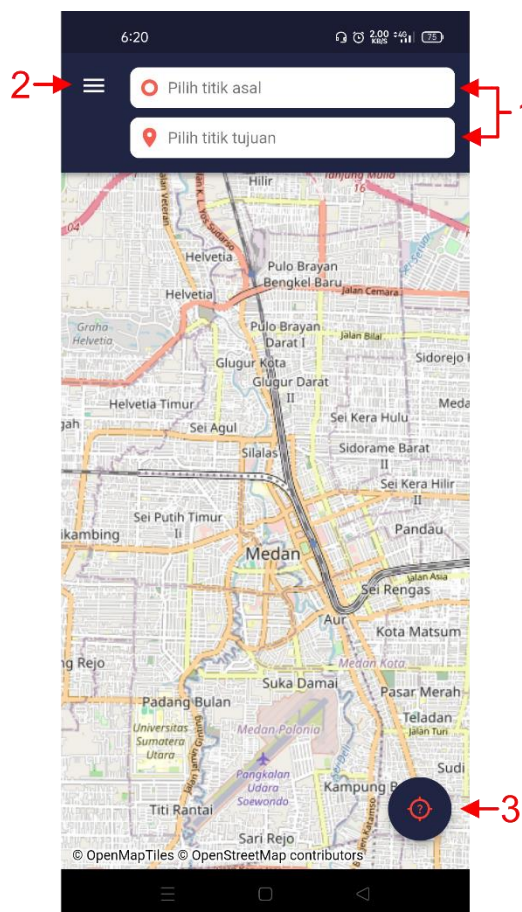
*Interface*, yang juga dikenal sebagai antarmuka, telah menjadi komponen integral dalam sistem yang berfungsi sebagai medium interaksi antara sistem dan pengguna. Dalam merancang antarmuka, diperlukan pemahaman yang jelas

tentang struktur sistem terkait dengan antarmuka yang akan dibangun sebelum memulai tahap implementasi.

Perancangan antarmuka sistem memerlukan tampilan yang sederhana agar memberikan kenyamanan dan kemudahan bagi pengguna dalam menggunakan sistem. Hal ini penting untuk memastikan bahwa hasil akhir sistem memberikan pengalaman yang baik bagi pengguna.

### 3.6.1 Tampilan Halaman Depan

Pada saat membuka aplikasi, aplikasi akan menampilkan halaman depan (Perencana Rute) sebagai halaman *default*, terdapat dua tombol dan dua form input pada halaman ini, berikut merupakan tampilan dari Halaman Depan dari aplikasi ini



**Gambar 3.6.** Tampilan Halaman Depan

Berikut keterangan dari Gambar 3.6. diatas :

1. Form input titik asal & tujuan

Form ini merupakan form yang akan diisi oleh *User* lalu sistem dapat mengkalkulasi rute-rute optimal dari kedua titik tersebut. Form ini juga akan terisi secara otomatis berdasarkan *input* dari sesi terakhir ketika *User* mengakses halaman ini.

2. Tombol *Side Navigation Bar*

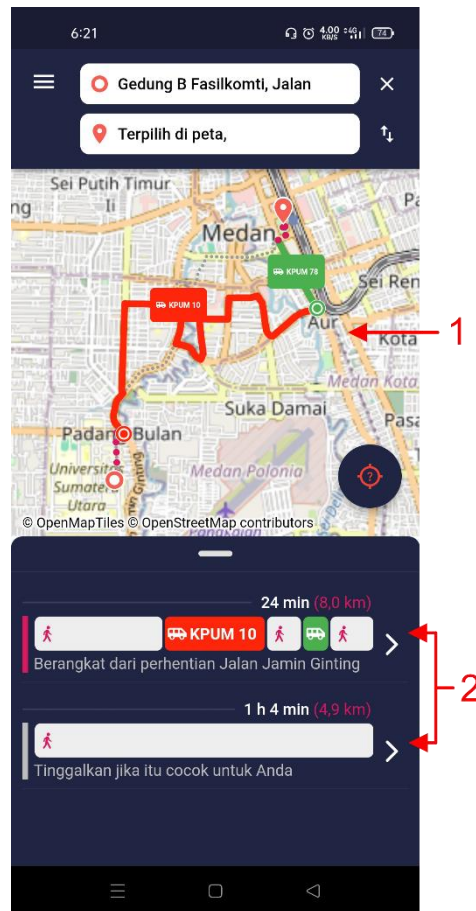
Tombol ini digunakan sebagai navigasi jika *User* ingin mengakses halaman lain.

3. Tombol Lokasi Saya

Tombol ini digunakan untuk menampilkan lokasi *User* pada saat menggunakan aplikasi tersebut. Sistem akan meminta *User* untuk mengizinkan akses lokasi pada perangkat saat *User* menekan tombol ini.

### 3.6.2 Tampilan Halaman Output dari Halaman Perencana Rute

Halaman ini adalah halaman yang ditampilkan oleh sistem ketika *User* sudah mengisi *input* pada form di halaman sebelumnya. Adapun tampilan dari halaman *Output* adalah sebagai berikut.



**Gambar 3.7.** Tampilan Halaman Output

Adapun keterangan dari Gambar 3.7. diatas adalah :

1. *Output* rute-rute optimal

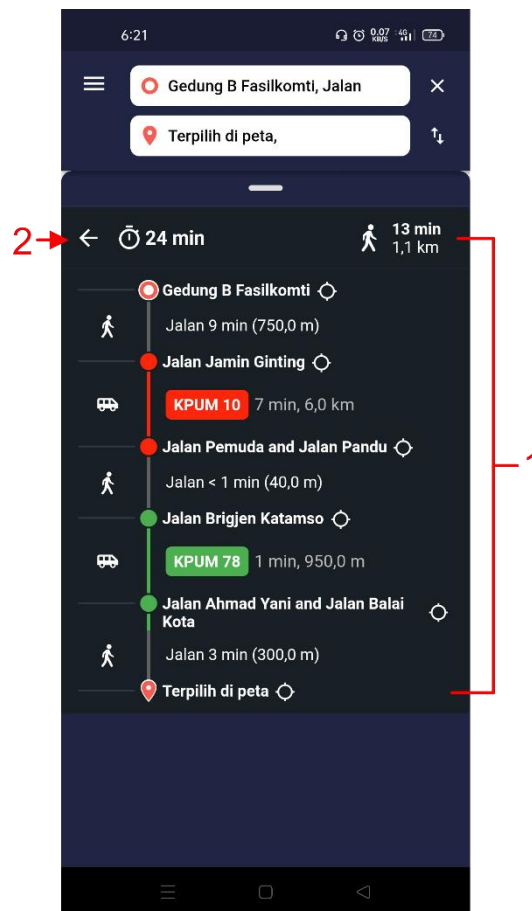
Bagian ini akan menampilkan seluruh rute optimal dari kedua titik yang sebelumnya diinput *User*. Rute yang ditempuh dengan berjalan kaki ditandai dengan garis putus-putus, dan pergantian moda transportasi ditandai dengan perbedaan warna rute.

2. Menu detail rute

Menu ini digunakan jika *User* hendak mengetahui penjelasan dan keterangan lebih detail dari rute yang ditampilkan pada poin No.1. Penjelasan ini berupa jarak, waktu tempuh, moda transportasi, dan sebagainya. Bagian ini juga dapat ditekan untuk menuju ke halaman selanjutnya yang menampilkan penjelasan lebih detail.

### 3.6.3 Tampilan Halaman Detail Rute

Halaman ini merupakan halaman yang menjelaskan tahapan perpindahan secara mendetail dari kedua titik yang sebelumnya diinput *User*.



**Gambar 3.8.** Tampilan Detail Rute

Adapun penjelasan dari keterangan yang terdapat dalam Gambar 3.8. adalah sebagai berikut :

#### 1. Halaman Detail Rute

Halaman ini menjelaskan tahapan yang harus ditempuh *User* untuk mencapai titik tujuan dari titik asal. Pada halaman ini terdapat jarak tempuh berjalan kaki, durasi berjalan kaki, durasi perjalanan total, nama dan jenis moda transportasi, koordinat perpindahan moda transportasi, dan jarak tempuh setiap moda transportasi.

## 2. Tombol Kembali

Tombol ini digunakan untuk kembali ke halaman sebelumnya (Halaman Output).

### 3.6.4 Tampilan Halaman Daftar Rute

Halaman ini merupakan halaman yang dapat diakses *User* melalui *Side Navigation Bar* dan berisikan seluruh daftar rute trayek yang tersedia dalam aplikasi ini.



**Gambar 3.9.** Tampilan Halaman Daftar Rute

Berikut keterangan dari gambar 3.9. yang terdapat diatas :

#### 1. Menu Daftar Rute

Pada menu ini ditampilkan seluruh trayek yang terdapat dalam aplikasi ini. Perbedaan moda transportasi ditandai dengan perbedaan ikon dan warna. Bagian ini juga dapat ditekan jika *User* hendak melihat detail

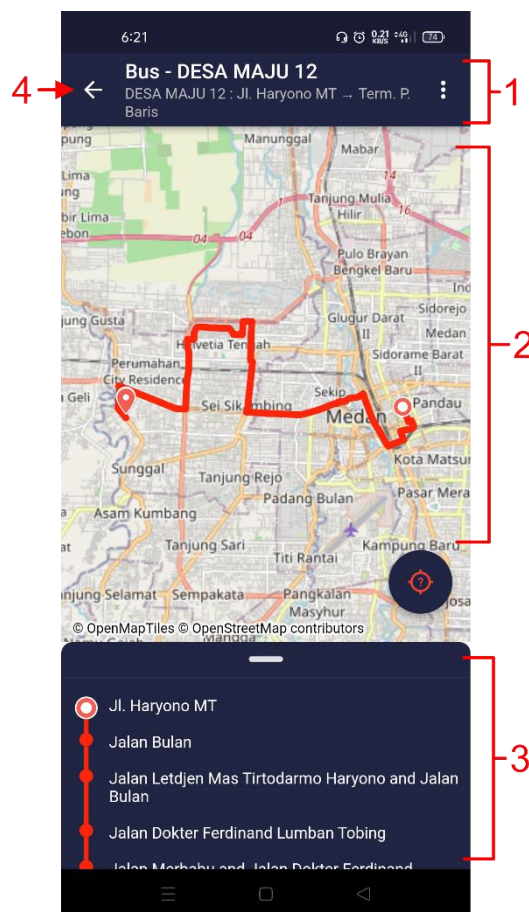
dari rute trayek tersebut. *User* akan diarahkan ke halaman selanjutnya (Halaman Detail Trayek Rute) ketika menekan salah satu trayek tersebut.

## 2. Form Pencarian Trayek

Form ini digunakan jika *User* ingin mencari detail dari trayek tertentu yang namanya sudah diketahui *User* sebelumnya.

### 3.6.5 Tampilan Halaman Detail Rute Trayek

Halaman ini adalah halaman yang ditampilkan aplikasi ketika *User* menekan salah satu rute trayek yang ditampilkan pada halaman sebelumnya (Halaman Daftar Rute). Halaman ini berisikan informasi tentang rute dari trayek yang dipilih *User*.



**Gambar 3.10.** Tampilan Halaman Detail Rute Trayek

Berikut ini adalah penjelasan dari keterangan yang terdapat pada Gambar 3.10. :

1. Judul atau Nama Trayek

Bagian ini menampilkan jenis transportasi trayek, nama trayek, dan lokasi awal & tujuan dari trayek tersebut.

2. Lintasan Trayek

Pada bagian ini ditampilkan lintasan dari lokasi awal hingga tujuan yang dilalui trayek yang dipilih.

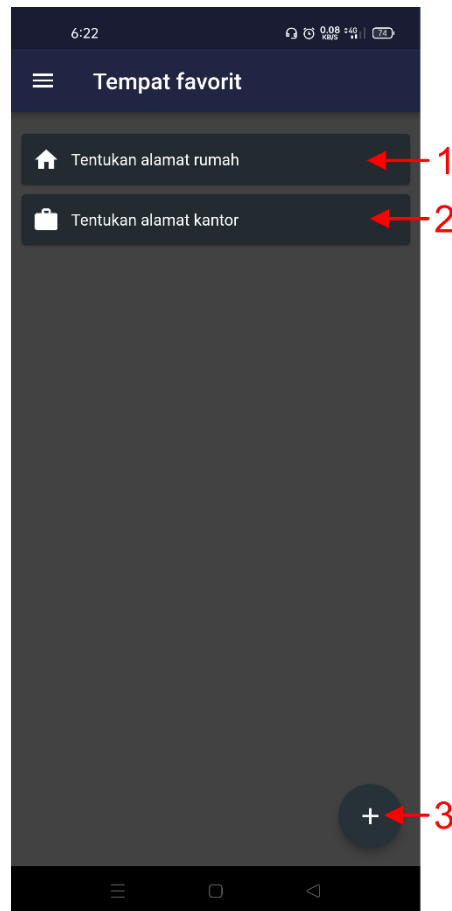
3. Menu Detail Rute Trayek

Menu ini menampilkan perhentian dan nama jalan sepanjang rute yang dilalui trayek yang dipilih.

### 3.6.6 Tampilan Halaman Tempat Favorit

Halaman ini merupakan fitur ketiga dan terakhir dari aplikasi ini, halaman ini dapat diakses *User* melalui *Side Navigation Bar*. Semua inputan yang terdapat pada halaman ini bersifat lokal, tersimpan secara lokal pada perangkat *User*. Fungsi dari halaman ini adalah untuk mempermudah pemilihan koordinat pada halaman utama (Halaman Perencana Rute). Secara *default*, halaman ini sudah menambahkan dua titik atau alamat, alamat rumah, dan alamat kantor. *User* hanya diperlukan untuk menambahkan koordinat atau titik dari kedua alamat tersebut agar tersimpan dalam aplikasi.





**Gambar 3.11.** Tampilan Halaman Tempat Favorit

Adapun keterangan yang terdapat pada Gambar 3.11. dijelaskan sebagai berikut :

1. Alamat Rumah

Tombol ini digunakan untuk menambahkan koordinat atau titik dari alamat rumah sesuai inputan *User*.

2. Alamat Kantor

Tombol ini digunakan untuk menambahkan koordinat atau titik dari alamat kantor sesuai inputan *User*.

3. Tombol menambahkan alamat baru

Tombol ini digunakan jika *User* hendak menambahkan alamat lain kedalam aplikasi. Selanjutnya *User* akan diminta untuk menamai alamat tersebut dan menginput koordinat seperti alamat rumah & kantor sebelumnya.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1. Implementasi Sistem

Dalam implementasi sistem ini, penulis memanfaatkan sejumlah aplikasi pendukung yang mendukung pembuatan dan beberapa perangkat tambahan untuk merancang serta mengoperasikan aplikasi yang telah selesai dibuat. Adapun aplikasi dan perangkat tersebut dijelaskan sebagai berikut.

##### 4.1.1. Aplikasi Pendukung

Berikut beberapa aplikasi pendukung tersebut yang digunakan dalam proses pembuatan aplikasi ini.

**Tabel 4.1** Aplikasi Pendukung

Nama Aplikasi	Keterangan
Visual Studio Code	Aplikasi yang digunakan untuk melakukan pemrograman dan penyesuaian <i>Framework OpenTripPlanner</i>
CorelDraw 2021	Aplikasi yang digunakan untuk mendesain rancangan dan komponen-komponen dalam sistem aplikasi ini
Android Studio	Aplikasi yang digunakan untuk melakukan pemrograman <i>Flutter</i>

#### 4.1.2. Perangkat Pendukung

Adapun perangkat-perangkat pendukung yang digunakan akan dijelaskan dalam tabel berikut.

**Tabel 4.2** Perangkat Pendukung

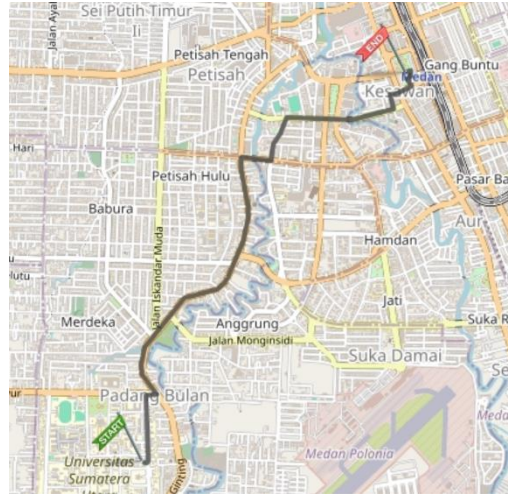
Nama Perangkat	Spesifikasi	Keterangan
Laptop	<ul style="list-style-type: none"> <li>- Windows 10</li> <li>- RAM 8 GB</li> <li>- Processor Intel i3</li> </ul>	Perangkat yang digunakan untuk merancang keseluruhan aplikasi
Virtual Private Server	<ul style="list-style-type: none"> <li>- Ubuntu</li> <li>- RAM 2 GB</li> <li>- Processor 2 Core</li> <li>- Bandwidth 10 MBps</li> </ul>	Perangkat yang digunakan untuk menjalankan <i>Framework OpenTripPlanner</i> pada <i>Backend Server</i>
Smartphone	<ul style="list-style-type: none"> <li>- Android 11</li> <li>- RAM 6 GB</li> </ul>	Perangkat yang digunakan untuk menjalankan dan mendemonstrasikan aplikasi

#### 4.2. Pengujian Sistem

Tahap pengujian sistem merupakan proses untuk mengevaluasi kesuksesan sistem yang telah dibangun sesuai dengan desain yang telah direncanakan sebelumnya. Berikut adalah proses pengujian sistem yang penulis lakukan dalam pembuatan aplikasi ini dengan membandingkan Algoritma A\* dengan *Framework OpenTripPlanner*. Adapun pada pengujian ini penulis menghitung biaya rute dari Kompleks FASILKOM-TI USU menuju Lapangan Merdeka Medan sebagai ilustrasi.



dasar. Berikut penjelasan dan perbandingan antara Algoritma A\* dengan Algoritma *Generalized Cost A\** yang digunakan pada aplikasi ini.



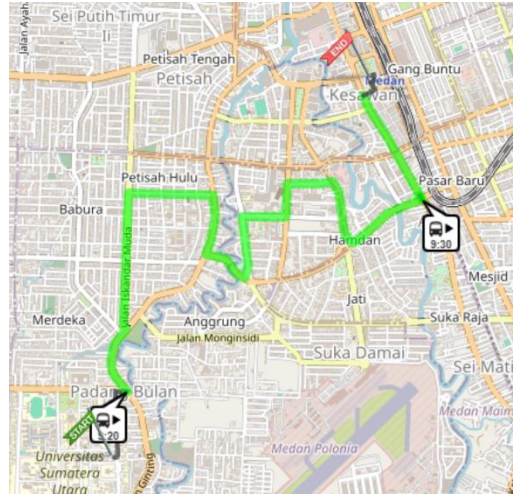
**Gambar 4.2.** Output Rute Terpendek dengan Algoritma A\*

Gambar diatas didapatkan dengan menggunakan *Framework OpenTripPlanner* dan menjalankannya hanya dengan menggunakan Algoritma A\*, tanpa penyesuaian biaya, parameter, dan lain-lain. Dari perhitungan tersebut dihasilkan bahwa jarak terdekat dari Kompleks FASILKOM-TI menuju Lapangan Merdeka Medan adalah 4,87 KM yang ditempuh dengan berjalan kaki. Namun pada penelitian aplikasi ini penulis beranggapan bahwa jarak tersebut tidak relevan untuk digunakan karena studi kasus dari aplikasi ini adalah transportasi umum. Sehingga perlu dilakukan penyesuaian biaya sesuai dengan variabel-variabel yang terlibat dalam studi kasus tersebut, seperti jenis moda transportasi, *Transfers* (Pergantian moda transportasi), *WalkDistance* (Jarak yang ditempuh dengan berjalan kaki), & Total jarak tempuh.

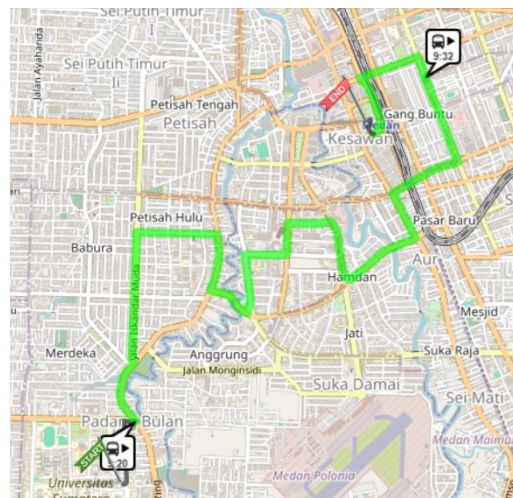
#### 4.2.2. Pengujian *OpenTripPlanner* tanpa Parameter Tambahan

Salah satu fitur atau manfaat dari menggunakan *Framework OpenTripPlanner* adalah pengembang aplikasi dapat menambahkan parameter tambahan untuk lebih menyesuaikan biaya (*Cost*) sesuai keperluan, agar menghasilkan *output* yang diharapkan. Adapun *output* dari pengujian

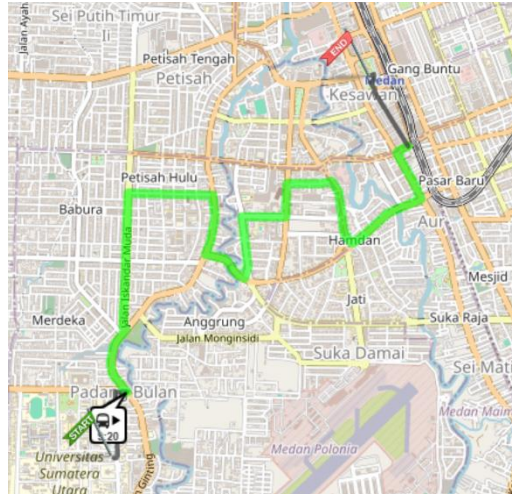
*Framework OpenTripPlanner* sebelum penambahan parameter tambahan adalah sebagai berikut.



(a)



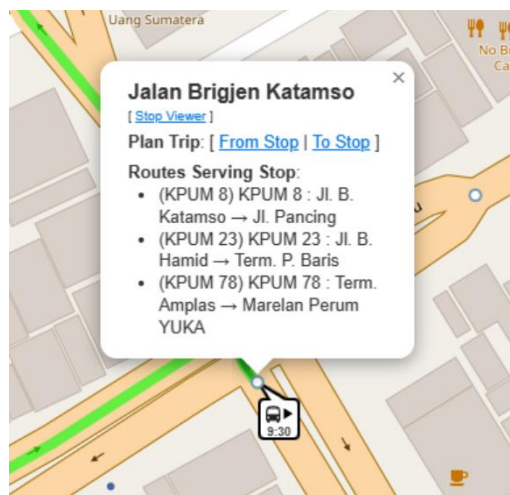
(b)



(c)



(d)



(e)

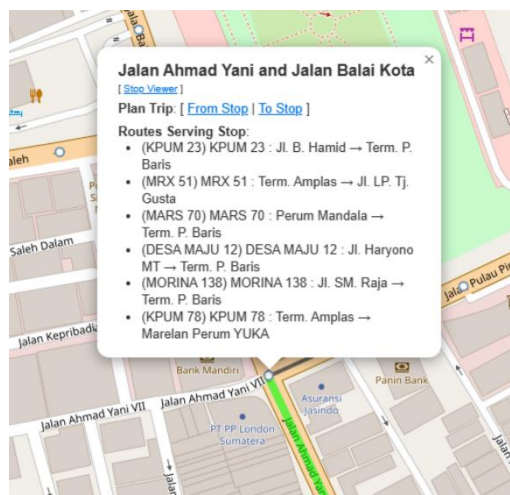




(f)



(g)

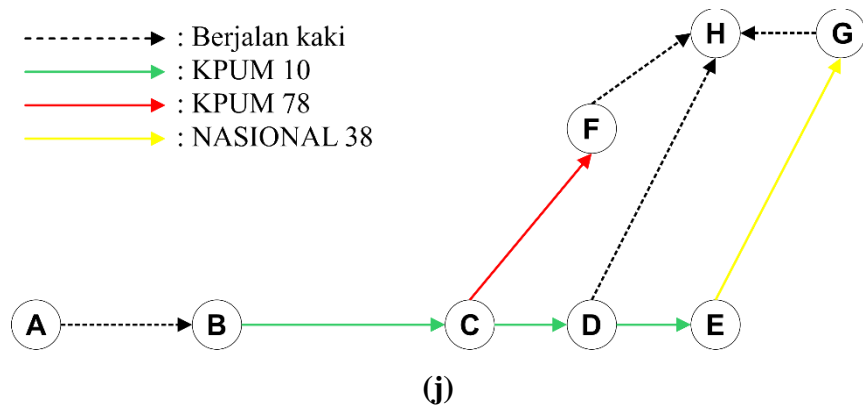


(h)





(i)



(j)

**Gambar 4.3.**

- (a) Output Rute Rencana Perjalanan 1.  
 (b) Output Rute Rencana Perjalanan 2.  
 (c) Output Rute Rencana Perjalanan 3.  
 (d) Perhentian pertama dari ketiga Rencana Perjalanan atau node B.  
 (e) Perhentian kedua dari Rencana Perjalanan 1 atau node C.  
 (f) Perhentian kedua dari Rencana Perjalanan 2 atau node D.  
 (g) Perhentian kedua dari Rencana Perjalanan 3 atau node E.  
 (h) Perhentian ketiga dari Rencana Perjalanan 1 atau node F.  
 (i) Perhentian ketiga dari Rencana Perjalanan 3 atau node G.  
 (j) Ilustrasi ketiga rute.

Gambar diatas didapatkan dengan menggunakan *Framework OpenTripPlanner* dan menjalankannya dengan penyesuaian biaya. Berdasarkan Gambar 4.3., dapat dilihat bahwa pengujian ini menghasilkan 3 jenis rute rencana perjalanan, adapun perhentian atau node-node yang terlibat dalam ketiga rencana perjalanan ini terdapat dalam Gambar 4.3 (d), (e), (f), (g), (h), & (i), dengan node A sebagai node asal (Kompleks FASILKOM-TI) dan node H sebagai node tujuan (Lapangan Merdeka Medan). Dan pada gambar 4.3 (j) penulis membuat graf sebagai bentuk ilustrasi sederhana dari ketiga rencana perjalanan tersebut. Adapun perhitungan dan perbandingan biaya dari output tersebut dijelaskan pada tabel berikut.

**Tabel 4.3.** Biaya ( $f(n)$ ) dari Waypoint (Trip) yang terlibat dengan menggunakan algoritma Generalized Cost  $A^*$ .

Waypoint (Trip)	A-B	B-C	B-D	B-E	C-F	E-G	C-F	F-H	D-H	G-H
$f(n)$	1015	1051	1088	1183	677	696	677	492	1075	288

**Tabel 4.4** Waypoint atau node yang dilalui rencana perjalanan tersebut dan biaya ( $f(n)$ ) total dari setiap rencana perjalanan

Rencana Perjalanan	Trip / Rute	$f(n)$
1	A-B-C-F-H	$f(n) = f(A-B) + f(B-C) + f(C-F) + f(F-H)$
		$f(n) = 1015 + 1051 + 677 + 492$
		$f(n) = 3235$
2	A-B-C-D-H	$f(n) = f(A-B) + f(B-D) + f(D-H)$
		$f(n) = 1015 + 1088 + 1075$
		$f(n) = 3178$
3	A-B-C-D-E-G-H	$f(n) = f(A-B) + f(B-E) + f(E-G) + f(G-H)$
		$f(n) = 1015 + 1183 + 696 + 288$
		$f(n) = 3182$

**Tabel 4.5** Jumlah Transfers, Jarak Berjalan Total (m), dan Jarak Total (m) dari setiap rencana perjalanan

Rencana Perjalanan	Transfers	Jarak Berjalan Total (m)	Jarak Total (m)
1	1	980	7970
2	0	1360	7860
3	1	1050	9900

**Tabel 4.6** Biaya (f(n)) baru yang telah ditimbang dengan Transfers, Jarak Berjalan Total dan Jarak Total (*Weighted Cost*)

Rencana Perjalanan	$f(n) = \text{Transfers} \times 0,65 + \text{Jarak Berjalan Total} \times 0,3 + (\text{Jarak Total} / 100) \times 0,05$
1	$f(n) = 1 \times 0,65 + 980 \times 0,3 + (7970/100) \times 0,05$
	$f(n) = 0,65 + 294 + 3,985$
	$f(n) = 298,635$
2	$f(n) = 0 \times 0,65 + 1360 \times 0,3 + (7860/100) \times 0,05$
	$f(n) = 0 + 408 + 3,93$
	$f(n) = 411,93$
3	$f(n) = 1 \times 0,65 + 1050 \times 0,3 + (9900/100) \times 0,05$
	$f(n) = 0,65 + 315 + 4,95$
	$f(n) = 320,6$

Berdasarkan perhitungan dan pengujian dari ilustrasi diatas, ditemukan bahwa *Framework OpenTripPlanner* melakukan dua kali perhitungan f(n) untuk mendapatkan hasil output. Perhitungan pada Tabel 4.3 & Tabel 4.4 berfungsi untuk menemukan rute-rute yang tersedia dari node A menuju node H dengan menggunakan kombinasi moda transportasi berjalan kaki dan transportasi umum yang tersedia didalam database. Pada tahap ini telah dibuat penyesuaian perbedaan biaya antara menggunakan transportasi umum dan berjalan kaki dengan skala 1 : 2,4.

Pengertian dari skala tersebut adalah, f(n) atau biaya yang dihasilkan dari rute atau waypoint yang ditempuh dengan berjalan kaki mempunyai nilai yang 2,4 kali lebih besar daripada jika jarak yang sama ditempuh dengan menggunakan transportasi umum. Dari perhitungan tersebut juga dihasilkan

bahwa rute yang paling optimal atau memiliki biaya terendah adalah Rencana Perjalanan 2.

Selanjutnya, berdasarkan output tersebut *Framework OpenTripPlanner* akan menghitung jumlah *Transfers* (Pergantian moda transportasi), Jarak Berjalan Total, & Jarak Total dari setiap Rencana Perjalanan (Tabel 4.5.). Lalu akan dilakukan perhitungan  $f(n)$  atau biaya sekali lagi dengan rumus :

$$f(n) = \text{Transfers} \times 0,65 + \text{Jarak Berjalan Total} \times 0,3 \\ + (\text{Jarak Total} / 100) \times 0,05$$

Hasil dari perhitungan ini terdapat pada Tabel 4.6. dan dihasilkan bahwa rute yang memiliki biaya terendah kali ini adalah Rencana Perjalanan 1.

#### 4.2.3. Pengujian *OpenTripPlanner* dengan Parameter Tambahan

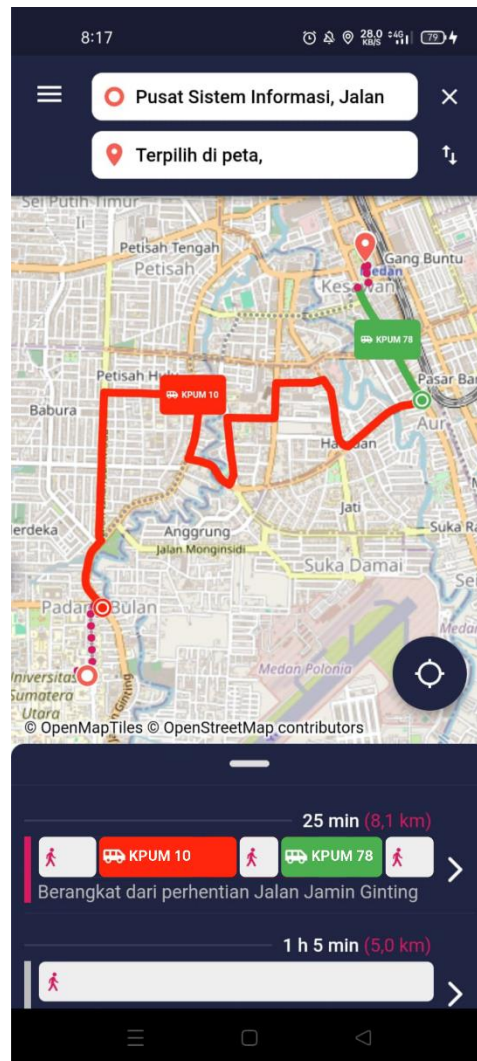
Untuk menghasilkan *Output* yang lebih optimal, penulis memutuskan untuk menambahkan beberapa parameter sesuai dengan kasus penggunaan *Framework OpenTripPlanner* tersebut. Pada kasus ini, karena moda transportasi yang digunakan adalah angkutan kota, dan trayek angkutan kota sudah cukup banyak untuk meliputi sebagian besar dari wilayah Kota Medan, penulis memutuskan untuk membatasi jumlah jarak berjalan kaki menjadi 1500. Berikut adalah parameter-parameter yang penulis gunakan dalam aplikasi ini :

- numItineraries : 7
- maxWalkDistance : 1000
- mode : \_parseTransportModes(transportModes)

Adapun kegunaan dari ketiga parameter tersebut adalah :

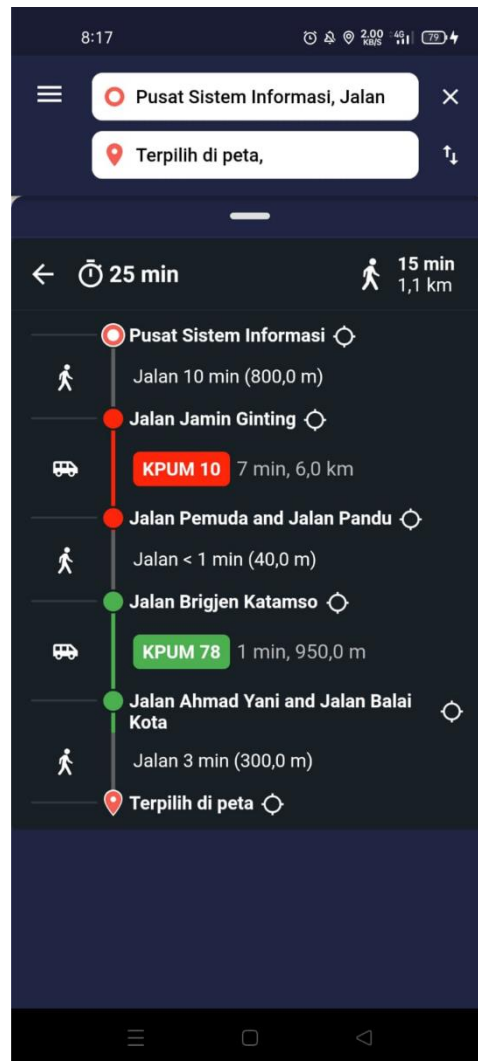
- Membatasi *Output* rute optimal menjadi 7
- Membatasi jarak tempuh berjalan kaki menjadi 1000 m
- Membatasi jenis moda transportasi yang digunakan menjadi hanya angkot & berjalan kaki

Berikut adalah output dan tampilan pada aplikasi dari *Framework OpenTripPlanner* ini setelah ditambahkan parameter-parameter tersebut.



**Gambar 4.4.** Tampilan halaman output pada aplikasi *smartphone*

Berdasarkan parameter-parameter yang penulis gunakan pada sistem ini, dapat dilihat pada Gambar 4.4. bahwa output akhir dari *Framework OpenTripPlanner* ini hanya menyisakan satu Rencana Perjalanan, yaitu Rencana Perjalanan 1. Hal itu disebabkan karena Rencana Perjalanan 1 & 2 tidak memenuhi kriteria dari salah satu parameter ( $\text{maxWalkDistance} : 1000$ ), yaitu jarak maksimal yang boleh ditempuh dengan berjalan kaki adalah 1000 m. Berdasarkan Tabel 4.5., diketahui bahwa Jarak Berjalan Total dari Rencana Perjalanan 2 adalah 1360m , & Jarak Berjalan Total dari Rencana Perjalanan 3 adalah 1050 m. Oleh karena itu kedua Rencana Perjalanan ini tereliminasi dari hasil output karena dianggap tidak memenuhi parameter.



**Gambar 4.5.** Tampilan halaman detail rute pada aplikasi *smartphone*

Selanjutnya, jika *User* menekan output tersebut, aplikasi akan menampilkan detail dari Rencana Perjalanan tersebut. Pada Gambar 4.5 terlihat informasi-informasi tentang Rencana Perjalanan tersebut berupa durasi perjalanan, durasi berjalan kaki, jarak berjalan kaki, nama trayek transportasi umum, dan lain sebagainya.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan evaluasi, perancangan, dan pengujian yang telah dilakukan dalam penelitian ini penulis menyimpulkan bahwa penggunaan *Framework OpenTripPlanner* pada sistem pendukung keputusan pemilihan angkutan umum di Kota Medan menghasilkan output yang sesuai dengan yang diharapkan. Hal ini dibuktikan dengan pengujian dan perbandingan yang kita lakukan sebelumnya. Dibandingkan dengan Algoritma A\* yang hanya menghasilkan satu output rute tercepat, *framework* ini mampu menghasilkan beberapa output sekaligus yang diurutkan berdasarkan biaya. Ketersediaan opsi juga merupakan hal yang krusial dalam merancang sebuah sistem pendukung keputusan karena setiap pengguna atau *User* dari sistem ini memiliki preferensi masing-masing. Dan dari sisi pengembang atau *developer*, *framework* ini mempunyai berbagai jenis versi, fitur, dan parameter yang bisa disesuaikan sesuai dengan kebutuhan.

#### **5.2. Saran**

Berikut beberapa saran yang penulis peroleh untuk pengembangan aplikasi ini :

1. Menambah dan melengkapi database dari rute transportasi umum sehingga meningkatkan jumlah opsi rute yang dapat dipilih *User*.
2. Menambahkan sistem pengurutan dan penyortiran pada halaman daftar rute.
3. Menambahkan informasi tentang estimasi biaya total dari output rute sehingga *User* dapat menggunakan informasi tersebut sebagai pertimbangan tambahan dalam memilih rute yang akan digunakan.

## DAFTAR PUSTAKA

- Aftabuzzaman, Md., 2007. Measuring traffic congestion- A critical review. 30th Australasian Transport Research Forum.
- Badan Pusat Statistik Kota Medan., 2022. Kota Medan Dalam Angka 2022. <https://medankota.bps.go.id/publication/2022/02/25/6b718cac3269c92f5de22b65/kota-medan-dalam-angka-2022.html>.
- Coblenz, M., Guo, W., Voozhian, K., & Foster, J. (n.d.). A Qualitative Study of REST API Design and Specification Practices. <https://www.cs.tufts.edu/~jfoster/papers/vlhcc23.pdf>
- Delling, D., Pajor, T. and Werneck, R.F., 2015. Round-Based Public Transit Routing. *Transportation Science*, 49(3), pp.591-604.
- Dere, Esat & Durdu, Akif., 2018. Usage of the A\* Algorithm to Find the Shortest Path in Transportation Systems. *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018)*.
- Docs.opentripplanner.org. (n.d.). OpenTripPlanner 2. [online] Available at: <https://docs.opentripplanner.org/en/v2.4.0/> [Accessed 19 Dec. 2023].
- Foead, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., Gunawan, E., 2021. A systematic literature review of a\* pathfinding. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2021.01.034>.
- Guo, Zhan., 2011. Mind the Map! The Impact of Transit Maps on Path Choice in Public Transit. *Transportation Research Part A: Policy and Practice*. 45. 625-639. 10.1016/j.tra.2011.04.001.
- Google., 2005. General Transit Feed Specification. <https://developers.google.com/transit/gtfs>.
- Jain, Rajni. (2016). *Decision Support Systems: an Overview*.
- Khodashahri, N. G., Sarabi, M. M., 2013. Decision support system ( DSS ). *Singaporean Journal of Business , Economics and Management Studies*, 1(6), 95–102. <https://doi.org/10.12816/0003780>.



- Kurale, R., & Bala, K. (2023). A Comparative Study of Flutter with other Cross-Platform Mobile Application Development. *International Journal of Creative Research Thoughts (IJCRT)*, ISSN: 2320-28
- Mandic, Danimir ., Pelemis, Momcilo., Pasalic, Stevo., Lalic, Nenad., 2012. Computer-based information system in education. 231-236.
- Abdulrazzaq, Layth R., 2020. Traffic Congestion: Shift from Private Car to Public Transportation. *Civil Engineering Journal* Vol. 6, No. 8.
- Moore, A. (2022). Using OpenStreetMap for your research: leveraging a massive global geographic database that emphasizes local knowledge. SFU Library. Retrieved from SFU Library website.
- Nababan, Adam B., 2018. Analisis Pengaruh Jumlah Transportasi Umum dan Transportasi Online Terhadap Kemacetan Di Kota Medan. Skripsi. Universitas Sumatera Utara. Medan.
- OpenTripPlanner. (n.d.). In OpenStreetMap Wiki. Retrieved February 5, 2024, from [<https://wiki.openstreetmap.org/wiki/OpenTripPlanner>]
- Perens, B., 1999. The open source definition. *Open sources: voices from the open source revolution*, 1, pp.171-188.
- Rachmawati, Dian., Gustin, Lysander., 2020. Analysis of Dijkstra's Algorithm and A\* Algorithm in Shortest Path Problem. *J. Phys.: Conf. Ser.* 1566 012061.
- Vuchic, V. R. (2002). *Urban public transportation systems*. University of Pennsylvania, Philadelphia, PA, USA, 5, 2532-2558.

## SURAT PERMOHONAN IZIN PENELITIAN



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS SUMATERA UTARA  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Kampus USU Medan 20155

Telepon: (061) 8221379

Laman: fasilkom-ti.usu.ac.id

Nomor : 132/UN5.2.1.14/PPM/2024

Lampiran : 1 (satu) set

Hal : Permohonan Izin Penelitian

Yth.  
Dinas Perhubungan Kota Medan

Sehubungan dengan Surat Permohonan Izin Penelitian yang diajukan mahasiswa sebagai berikut:

Nama : REKSA ANJANGSARA  
NIM : 171401097  
Program : S1  
Program Studi : Ilmu Komputer  
Semester : 13  
Alamat Mahasiswa : Jl. Nazir Alwi No.6 Merdeka, Kec. Medan Baru  
Judul : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN ANGKUTAN UMUM  
Proposal : DI KOTA MEDAN BERBASIS MOBILE MENGGUNAKAN OPENTRIPLANNER  
Lokasi Penelitian : Kantor Dinas Perhubungan Kota Medan  
Ditujukan Kepada : Dinas Perhubungan Kota Medan  
Dosen Pembimbing : Sri Melvani Hardi, S.Kom., M.Kom

Maka dengan ini kami mohon kesediaan Bapak/Ibu untuk dapat memberikan Izin Penelitian kepada mahasiswa yang tersebut di atas. Penelitian ini diperlukan mahasiswa untuk mengumpulkan data/informasi sebagai bahan untuk menyelesaikan Skripsi/Tugas Akhir.

Demikian hal ini disampaikan, atas perhatian dan kerjasamanya diucapkan terima kasih.

Medan, 15 Januari 2024  
Ditandatangani secara elektronik oleh:  
Dekan



Dr. Maya Silvi Lydia, B.Sc., M.Sc.  
NIP 197401272002122001

## SURAT KETERANGAN RISET BADAN RISET DAN INOVASI DAERAH



### PEMERINTAH KOTA MEDAN BADAN RISET DAN INOVASI DAERAH

Jalan Jenderal Besar A.H. Nasution Nomor 32, Medan Johor, Medan, Sumatera Utara 20143,  
Telepon (061) 7873439, Faksimile (061) 7873144  
Laman [brida.pemkomedan.go.id](http://brida.pemkomedan.go.id), Pos-el [brida@pemkomedan.go.id](mailto:brida@pemkomedan.go.id)

#### SURAT KETERANGAN RISET

Nomor : 000.9/0571

DASAR	: 1. Peraturan Daerah Kota Medan Nomor : 8 Tahun 2022, tanggal 30 Desember 2022 tentang Perubahan Atas Peraturan Daerah Kota Medan Nomor 15 Tahun 2016 Tentang Pembentukan Perangkat Daerah Kota Medan. 2. Peraturan Walikota Medan Nomor : 97 Tahun 2022, tanggal 30 Desember 2022 tentang Kedudukan, Susunan Organisasi, Tugas dan Fungsi, dan Tata Kerja Perangkat Daerah Kota Medan.
MENIMBANG	: Surat dari Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara Nomor 132/UN5.2.1.14/PPM/2024 Tanggal 15 Januari 2024 Perihal Permohonan Izin Penelitian.
NAMA	: <b>Reksa Anjarsara</b>
NIM	: 171401097
PROGRAM STUDI	: Ilmu Komputer
LOKASI	: Dinas Perhubungan Kota Medan
JUDUL	: <b>“Sistem Pendukung Keputusan Pemilihan Angkutan Umum di Kota Medan Berbasis Mobile Menggunakan OpenTripPlanner”</b>
LAMANYA	: 3 (Tiga) Bulan
PENANGGUNG JAWAB	: Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara

Melakukan Riset, dengan ketentuan sebagai berikut :

1. Sebelum melakukan Riset terlebih dahulu harus melapor kepada Pimpinan Perangkat Daerah lokasi yang ditetapkan.
2. Mematuhi Peraturan dan Ketentuan yang berlaku di lokasi Riset.
3. Tidak dibenarkan melakukan Riset atau aktivitas lain di luar lokasi yang telah ditetapkan.
4. **Hasil Riset diserahkan kepada Kepala Badan Riset dan Inovasi Daerah Kota Medan selambat-lambatnya 2 (dua) bulan setelah Riset dalam bentuk *softcopy* atau melalui Email [brida@pemkomedan.go.id](mailto:brida@pemkomedan.go.id).**
5. Surat keterangan Riset dinyatakan batal apabila pemegang surat keterangan tidak mengindahkan Ketentuan atau Peraturan yang berlaku pada Pemerintah Kota Medan.
6. Surat keterangan Riset ini berlaku sejak tanggal dikeluarkan.

Demikian Surat ini diperbuat untuk dapat dipergunakan sebagaimana mestinya.

Dikeluarkan di : M e d a n  
Pada Tanggal : 15 Maret 2024



Dilandatangani secara elektronik oleh :  
Kepala Badan Riset dan Inovasi Daerah Kota Medan,  
Mansursyah, S. Sos, M. AP  
Pembina Tk. I (IV/b)  
NIP 196805091989091001

#### Tembusan :

1. Wali Kota Medan.
2. Kepala Dinas Perhubungan Kota Medan.
3. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara



- Dokumen ini telah ditandatangani secara elektronik, menggunakan sertifikat elektronik yang diterbitkan BSrE.  
- UU ITE No. 11 Tahun 2008 Pasal 5 Ayat 1 "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah."

**SURAT KETERANGAN RISET DINAS PERHUBUNGAN KOTA MEDAN****PEMERINTAH KOTA MEDAN  
DINAS PERHUBUNGAN**

Jalan Pinang Baris Nomor 114A, Medan Sunggal, Medan, Sumatera Utara 20127  
Telepon (061) 8450542 Laman : Dishub.Pemkomedan.go.id  
Pos-el : dishubkotamedan1@gmail.com

Medan, 02 April 2024

Nomor : 000.9.2 / 1189  
Lampiran : -  
Hal : Selesai Riset

Yth. Dekan Fakultas Ilmu Komputer  
Dan Teknologi Informasi Universitas Sumatera Utara  
di  
Tempat

Dengan hormat ;

1. Menindak lanjuti Surat Kepala Badan Riset dan Inovasi Daerah Kota Medan; Nomor: 000.9/0571 Tanggal 15 Maret 2024 Perihal : Surat Keterangan Riset A.n Reksa Anjarsara ; NPM : 171401097 ; Jurusan ; Ilmu Komputer; Judul : Sistem Pendukung Keputusan Pemilihan Angkutan Umum di Kota Medan berbasis Mobile Menggunakan OpenTripPlanner.
2. Berkaitan dengan hal tersebut di atas, dijelaskan bahwa yang bersangkutan telah selesai melaksanakan Riset dan pengambilan data pada Dinas Perhubungan Kota Medan.
3. Demikian disampaikan untuk dapat dipergunakan sebagaimana mestinya.

An. Kepala Dinas Perhubungan Kota Medan  
Sekretaris  
U.b KASUBBAG KEPEGAWAIAN DAN UMUM  
DINAS PERHUBUNGAN KOTA MEDAN



MELDA HERLINA, SH.MSI  
PEMBINA  
NIP. 19740312 199803 2 003