

PERANCANGAN APLIKASI INFORMASI RUTE ANGKOT BERBASIS WEB MENGGUNAKAN ALGORITMA BREADTH FIRST SEARCH DI KOTA BANDUNG

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan
Progra Stara 1, Program Studi Teknik Informatika,
Universitas Pasundan Bandung

Oleh :

Raisa Isna Ainun

Nrp. 21.304.0112



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN BANDUNG
JANUARI 2026**

**LEMBAR PENGESAHAN
LAPORAN TUGAS AKHIR**

Telah disetujui dan disahkan Laporan Tugas Akhir, dari :

Nama : Raisa Isna Ainun

Nrp. : 21.304.0112

Dengan Judul :

“Perancangan Aplikasi Informasi Rute Angkot Berbasis Web Menggunakan
Algoritma Breadth First Search Di Kota Bandung”



Bandung, 21 Januari 2026
Menyetujui, Pembimbing Utama,



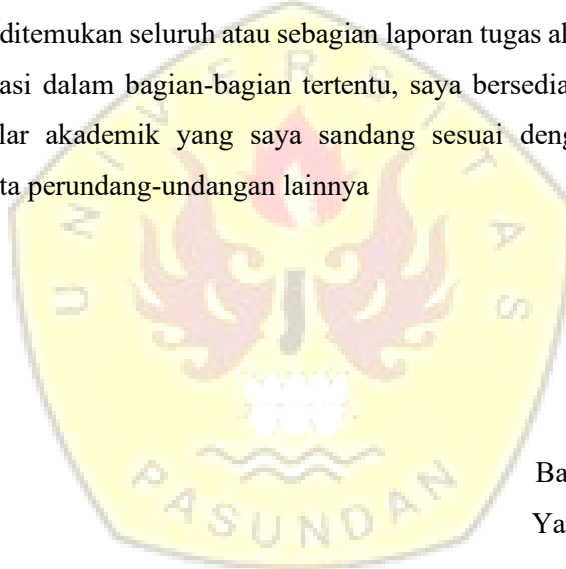
(Ade Sukendar, ST., MT.)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa:

1. Tugas akhir ini sepenuhnya merupakan hasil karya orisinal saya dan belum pernah diajukan untuk memperoleh gelar akademik di Universitas Pasundan Bandung maupun institusi pendidikan lainnya.
2. Tugas akhir ini murni merupakan ide, konsep, dan hasil penelitian saya sendiri, tanpa melibatkan bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini, tidak terdapat karya atau pendapat orang lain kecuali pada bagian-bagian tertentu yang dikutip secara jelas dengan mencantumkan sumber sesuai dengan norma, aturan, dan etika penulisan karya ilmiah, serta dirujuk dalam Daftar Pustaka.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya



Bandung, 21 Januari 2026

Yang membuat pernyataan,



(Raisa Isna Ainun)

NRP. 21.304.0112

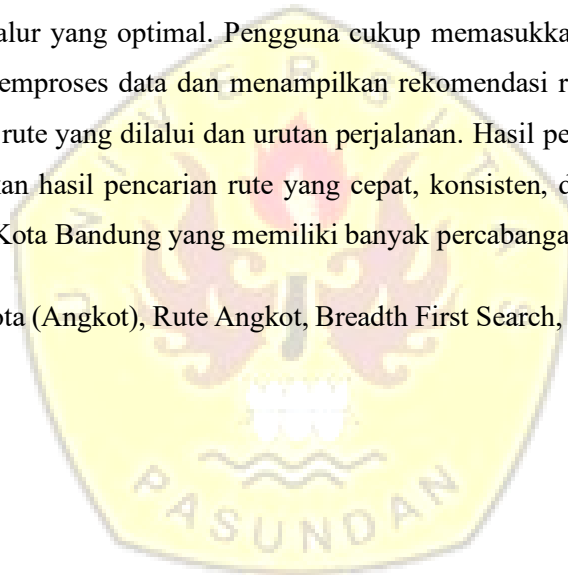
ABSTRAK

Angkutan kota (angkot) merupakan salah satu moda transportasi umum yang berperan penting dalam menunjang mobilitas masyarakat di Kota Bandung. Setiap angkot memiliki rute tertentu yang menghubungkan berbagai wilayah melalui jalur dan titik pemberhentian yang telah ditetapkan. Namun, keterbatasan informasi mengenai rute angkot yang mudah diakses dan dipahami sering menjadi kendala bagi masyarakat, khususnya pengguna baru, dalam menentukan jalur perjalanan yang efisien.

Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem pencarian rute angkot berbasis web yang mampu memberikan rekomendasi rute berdasarkan titik awal dan titik tujuan pengguna. Sistem ini menerapkan algoritma Breadth First Search (BFS) untuk menentukan jalur angkot dengan jumlah lintasan paling sedikit. Rute angkot dimodelkan dalam bentuk graf, di mana setiap titik pemberhentian atau persimpangan direpresentasikan sebagai node, sedangkan keterhubungan antar titik direpresentasikan sebagai edge.

Algoritma BFS dipilih karena mampu menelusuri graf secara sistematis dan menyeluruh sehingga menghasilkan jalur yang optimal. Pengguna cukup memasukkan titik awal dan titik tujuan, kemudian sistem akan memproses data dan menampilkan rekomendasi rute angkot beserta informasi pendukung, seperti nama rute yang dilalui dan urutan perjalanan. Hasil pengujian menunjukkan bahwa sistem mampu memberikan hasil pencarian rute yang cepat, konsisten, dan sesuai dengan kebutuhan pencarian rute angkot di Kota Bandung yang memiliki banyak percabangan jalur.

Kata Kunci: Angkutan Kota (Angkot), Rute Angkot, Breadth First Search, Pencarian Rute Angkot



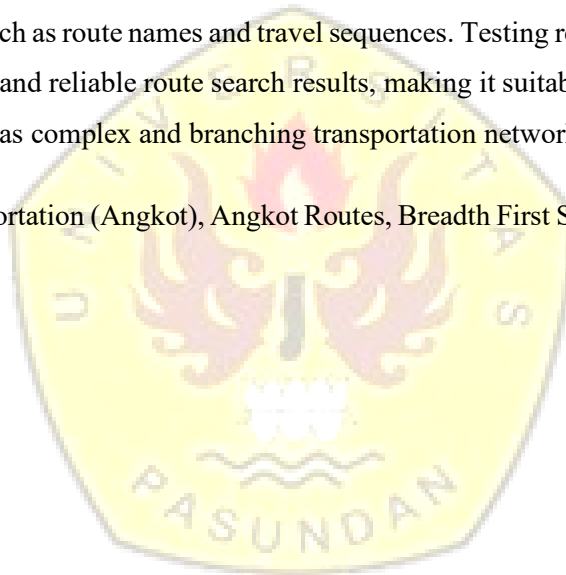
ABSTRACT

Public Transportation (angkot) play an important role in supporting public mobility in Bandung City. Each angkot operates on specific routes that connect various areas through predetermined roads and stopping points. However, limited access to clear and understandable route information often becomes an obstacle for users, especially new passengers, in determining efficient travel paths.

This study aims to design and develop a web-based angkot route search system that provides route recommendations based on the user's starting point and destination. The system applies the Breadth First Search (BFS) algorithm to determine angkot routes with the minimum number of traversed paths. Angkot routes are modeled as a graph, where each stop or intersection is represented as a node, and the connections between them are represented as edges.

The BFS algorithm is chosen for its ability to systematically and comprehensively explore the graph, resulting in optimal route selection. Users only need to input their starting point and destination, after which the system processes the data and displays recommended angkot routes along with supporting information such as route names and travel sequences. Testing results indicate that the system produces fast, consistent, and reliable route search results, making it suitable for angkot route searching in Bandung City, which has complex and branching transportation networks.

Keywords: Public Transportation (Angkot), Angkot Routes, Breadth First Search, Angkot Route Search



KATA PENGANTAR

Ucapan dan rasa syukur penulis layangkan ke hadirat Ilahi Robbi, yang telah berkenan menguatkan penulis untuk membuat Laporan Tugas Akhir dengan judul “Perancangan Rute Angkot Menggunakan Algoritma Breadth First Search Di kota Bandung”.

Adapun penulisan laporan ini bertujuan untuk memenuhi salah satu syarat kelulusan Program Strata 1, di Program Studi Teknik Informatika Universitas Pasundan.

Penulis menyadari laporan ini dapat terwujud berkat bantuan dan dorongan dari berbagai pihak. Maka pada kesempatan ini penulis sampaikan terima kasih yang sebesar-besarnya atas segala bantuan yang penulis terima baik secara moril maupun materil, sehingga penulis dapat menyelesaikan laporan ini kepada :

1. Kepada Allah SWT atas rahmat dan karuniannya.
2. Kepada Kedua Orang tua dan keluarga, atas doa, dukungan yang tidak pernah henti.
3. Keapad Bapak Ade Sukendar ST.,MT yang telah memberikan arahan, kritik, Saran serta masukan selaman proses penyusunan tugas akhir ini.
4. Kepada Bapak Wanda Gusdya Purnama, ST., MT. selaku Koordinator Tugas Akhir.
5. Seluruh civitas academica Program Studi Teknik Informatika Universitas Pasundan Bandung atas ilmu, bimbingan, dan pengalaman akademik yang telah diberikan kepada penulis selama masa perkuliahan.
6. Kepada Teman Saya Rika Febrianti, Adawiyah, Putri Azizah dan Teman seperjuangan lainnya yang telah memberikan dukungan semangat, masukan, dan bantuan teknis dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa laporan ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Akhir kata, Semoga laporan ini dapat memberikan manfaat bagi penulis dan pengembangan ilmu pengetahuan dan teknologi.

Bandung, Januari 2026

Raisa Isna Ainun

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR ISTILAH.....	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xi
DAFTAR SIMBOL.....	xii
BAB 1 PENDAHULUAN.....	1-1
1.1 Latar Belakang	1-1
1.2 Identifikasi Masalah.....	1-2
1.3 Tujuan Tugas Akhir.....	1-2
1.4 Lingkup Tugas Akhir	1-2
1.5 Metodologi Tugas Akhir	1-4
1.6 Sistematika Penelitian.....	1-6
BAB 2 LANDASAN TEORI.....	2-7
2.1 Teori Yang Digunakan.....	2-7
2.1.1 Website.....	2-7
2.1.2 Angkutan Kota.....	2-8
2.1.3 Algoritma Breadth First Search	2-8
2.1.4 Graph.....	2-10
2.1.5 Trayek	2-11
2.1.6 Waterfall.....	2-12
2.1.7 Framework	2-13
2.1.8 Laravel	2-14
2.1.9 Tailwind CSS.....	2-15
2.1.10 Leaflet.js.....	2-16
2.2 Penelitian Terdahulu.....	2-18
BAB 3 SKEMA PENULISAN.....	3-1
3.1 Alur Penyelesaian	3-1
3.2 Perumusan Masalah.....	3-3
3.2.1 Analisis Sebab Akibat	3-3
3.2.2 Analisis Solusi Masalah	3-3
3.2.3 Kerangka Berpikir Teoritis.....	3-4
BAB 4 ANALISIS DAN PERANCANGAN.....	4-1
4.1 Analisis Perancangan	4-1
4.1.1 Model Analisis.....	4-1

4.1.2 Model Use Case.....	4-4
4.2 Perancangan	4-5
4.2.1 Diagram Skenario	4-6
4.2.2 Diagram Sequence	4-7
4.2.3 Pemodelan Kelas Analisis	4-9
BAB 5 IMPLEMENTASI DAN PENGUJIAN.....	5-1
5.1 Kontruksi Perangkat	5-1
5.1.1 Kontruksi Kakas Perangkat Lunak.....	5-1
5.1.2 Kontruksi Kakas Perangkat Keras	5-1
5.2 Pengkodean	5-2
5.2.1 Struktur Aplikasi.....	5-2
5.3 Implementasi AntarMuka.....	5-4
5.4 Pengujian Perangkat Lunak.....	5-8
BAB 6 KESIMPULAN DAN SARAN.....	6-1
6.1 Kesimpulan	6-1
6.2 Saran.....	6-1
DAFTAR PUSTAKA.....	i
LAMPIRAN	A-1



DAFTAR ISTILAH

No	Istilah	Keterangan
1.	Algoritma	Urutan langkah-langkah logis dan terstruktur yang digunakan untuk menyelesaikan suatu masalah secara sistematis hingga menghasilkan keluaran yang diinginkan.
2.	FIFO (First In, First Out)	Metode pemrosesan data di mana elemen yang pertama masuk akan diproses atau dikeluarkan terlebih dahulu.
3.	Node	Simpul pada graf yang merepresentasikan suatu titik, objek, atau lokasi tertentu dalam suatu jaringan atau sistem.
4.	Edge	Penghubung antar node dalam graf yang merepresentasikan hubungan atau jalur antara dua simpul.
5.	Queue	Struktur data berurutan yang menerapkan prinsip FIFO (First In, First Out), di mana elemen pertama masuk diproses lebih dahulu.
6.	Dequeue	Operasi pada struktur data queue untuk menghapus atau mengambil elemen terdepan dari antrian.
7.	Visited	Penanda yang digunakan dalam algoritma pencarian untuk menunjukkan bahwa suatu node telah dikunjungi sehingga tidak diproses kembali.
8.	Fishbone	Diagram sebab-akibat yang digunakan untuk mengidentifikasi dan menganalisis penyebab suatu masalah.
9.	Database	kumpulan data yang tersusun secara terstruktur dan saling berhubungan, yang disimpan secara sistematis agar dapat diakses, dikelola, dan diperbarui dengan mudah
10.	Requirement	kebutuhan atau spesifikasi yang harus dipenuhi oleh suatu sistem agar dapat berfungsi sesuai tujuan yang diharapkan.
11.	Design	Proses perancangan sistem yang menggambarkan struktur, tampilan, dan alur kerja sebelum sistem dikembangkan.
12.	Testing	Proses pengujian sistem untuk memastikan fungsi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan.
13.	Use Case	Gambaran interaksi antara pengguna dan sistem yang menunjukkan fungsi-fungsi yang dapat dilakukan oleh sistem.
14.	Sequence	diagram yang menggambarkan urutan interaksi

DAFTAR TABEL

Tabel 2. 1 Pseudocode Algoritma Breadth First Search.....	2-10
Tabel 2. 2 Penelitian Terdahulu.....	2-18
Tabel 3. 1 Alur Penyelesaian.....	3-1
Tabel 4. 1 Diagram Aktivitas	4-2
Tabel 4. 2 Identifikasi User Requirement	4-3
Tabel 4. 3 Identifikasi Software Requirement.....	4-3
Tabel 4. 4 Function Requirement	4-4
Tabel 4. 5 Use Case Aplikasi Rute Angkot.....	4-5
Tabel 4. 6 Definisi Aktor	4-5
Tabel 4. 7 Diagram Sequence Melakukan Pencarian Rute	4-8
Tabel 4. 8 Diagram Sequence Melihat Detail Informasi.....	4-8
Tabel 4. 9 Diagram Sequence Milih Daftar Trayek.....	4-9
Tabel 4. 10 Mengidentifikasi Kelas – Kelas Analisis	4-10
Tabel 4. 11 Menentukan Atribut.....	4-10
Tabel 4. 12 Menentukan Operasi.....	4-11
Tabel 4. 13 Antarmuka Halaman Home.....	4-19
Tabel 4. 14 Antarmuka Halaman Pencarian Rute.....	4-20
Tabel 4. 15 Antarmuka Halaman Detail Informasi.....	4-21
Tabel 4. 16 Antarmuka Halaman Daftar Trayek.....	4-22
Tabel 5. 1 Kakas Perangkat Lunak.....	5-1
Tabel 5. 2 Kakas Perangkat Keras.....	5-2
Tabel 5. 3 View.....	5-2
Tabel 5. 4 Controlller.....	5-3
Tabel 5. 5 Model	5-3
Tabel 5. 6 Services	5-3
Tabel 5. 7 Implementasi Antarmuka Halaman Home.....	5-4
Tabel 5. 8 Implementasi Antarmuka Halaman Pencarian.....	5-5
Tabel 5. 9 Implementasi Antarmuka Halaman Detail	5-6
Tabel 5. 10 Implementasi Antarmuka Halaman Daftar Trayek.....	5-7
Tabel 5. 11 Pengujian Perangkat Lunak	5-8

DAFTAR GAMBAR











Gambar 1. 1 Metodologi Tugas Akhir.....	1-4
Gambar 2. 1 Model Waterfal Presman.....	2-13
Gambar 2. 2 Laravel.....	2-15
Gambar 2. 3 TailwindCSS	2-16
Gambar 2. 4 Contoh Map Leaflet.js	2-17
Gambar 3. 1 Analisis Sebab Akibat.....	3-3
Gambar 3. 2 Analisis Solusi Masalah.....	3-4
Gambar 3. 3 Kerangka Berpikir Teoritis	3-5
Gambar 4. 1 Diagram Aktivitas.....	4-2
Gambar 4. 2 Use Case Aplikasi Rute Angkot	4-5
Gambar 4. 3 Diagram Sequence Melakukan Pencarian Rute.....	4-8
Gambar 4. 4 Diagram Sequence Melihat Detail Informasi	4-8
Gambar 4. 5 Diagram Sequence Milih Daftar Trayek	4-9
Gambar 4. 6 Diagram Kelas.....	4-16
Gambar 4. 7 Skema Basis Data.....	4-17
Gambar 4. 8 Contoh Graph BFS	4-18
Gambar 4. 9 Perancangan Arsitektur.....	4-18
Gambar 4. 10 Antarmuka Halaman Home.....	4-19
Gambar 4. 11 Antarmuka Halaman Pencarian Rute	4-20
Gambar 4. 12 Antarmuka Halaman Detail Informasi.....	4-21
Gambar 4. 13 Antarmuka Halaman Daftar Trayek.....	4-22
Gambar 5. 1 Implementasi Antarmuka Halaman Home.....	5-5
Gambar 5. 2 Implementasi Antarmuka Halaman Pencarian	5-6
Gambar 5. 3 Implementasi Antarmuka Halaman Detail.....	5-7
Gambar 5. 4 Implementasi Antarmuka Halaman Daftar Trayek.....	5-7

DAFTAR LAMPIRAN

Lampiran 1 Berita Acara.....	A-1
Lampiran 2 Laporan Wawancara	A-2
Lampiran 3 Kuesioner.....	A-3
Lampiran 4 Dokumentasi	A-6
Lampiran 5 Kodingan	A-8
Lampiran 6 Daftar Rute.....	A-10



DAFTAR SIMBOL

Simbol	Nama Simbol	Keterangan
	Activity	Simbol ini digunakan untuk menunjukkan aktivitas yang dilakukan oleh aktor yang terlibat.
	Transition	Simbol ini digunakan untuk merepresentasikan alur aktivitas.
	Off Page Connector	Simbol ini digunakan untuk keluar / masuk prosedur atau proses dalam lembar / halaman yang lain.
	Input / Output	Simbol ini digunakan masukan / keluaran dari aktifitas.
	Start state	Simbol ini digunakan untuk merepresentasikan awal dimulainya sebuah aktivitas.
	End state	Simbol ini digunakan untuk merepresentasikan akhir dari sebuah aktivitas.
	Actor	Simbol ini digunakan untuk merepresentasikan orang yang berinteraksi dengan sistem/aplikasi.
	View Object	Simbol ini digunakan untuk merepresentasikan antarmuka sistem yang berinteraksi dengan aktor/pengguna.
	Controller Object	Simbol ini digunakan untuk merepresentasikan objek yang mengendalikan aliran jalannya sistem atau use case.
	Model Object	Simbol ini digunakan untuk merepresentasikan informasi yang tersimpan dan dipertahankan untuk jangka waktu yang lama, seperti data dalam database.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kota Bandung merupakan salah satu kota besar di Indonesia, yang bertindak sebagai ibu kota provinsi Jawa Barat, dengan populasi mencapai 2.843.977 orang dan luas wilayah sebesar 167,7 km². Tercatat sekitar 1,25 juta kendaraan bermotor, di mana 94% merupakan kendaraan pribadi. Jenis kendaraan pribadi ini sebagian besar adalah sepeda motor, yang jumlahnya mencapai 895.000 unit, sedangkan jumlah mobil pribadi mencapai 282.000 unit. Menurut data dari Dinas Perhubungan Kota Bandung pada tahun 2015, terdapat 36 rute angkutan umum di Bandung dengan total armada angkutan kota sebanyak 5.521 unit, dan total panjang rute mencapai 1.114,9 km untuk setiap perjalanan. Dari data yang ada, terlihat bahwa jumlah kendaraan pribadi jauh lebih banyak dibandingkan jumlah angkutan umum yang tersedia di Kota Bandung[ADI17].

Dalam kehidupan sehari-hari, masyarakat memanfaatkan transportasi umum yang efisien untuk memenuhi kebutuhan mobilitas. Salah satu jenis transportasi ini adalah angkutan umum. Dengan mempertimbangkan berbagai keuntungan yang ditawarkan oleh penggunaan angkot, banyak individu masih menunjukkan minat yang tinggi. Angkot beroperasi dengan sistem yang mengikuti rute tertentu. Setiap rute terdiri dari beberapa jalan. Rute yang banyak dilalui oleh masyarakat kemudian diorganisir menjadi sebuah "jurusan." Semua angkot yang menuju jurusan yang sama cenderung memiliki waktu tempuh yang hampir serupa. Transportasi merupakan kegiatan pergerakan yang sangat vital dalam kehidupan masyarakat Indonesia. Angkutan umum di kawasan perkotaan berfungsi sebagai bagian integral dari sistem transportasi kota yang memegang peranan penting dalam mendukung mobilitas penduduk. Fungsi ini menjadikan angkutan umum di area perkotaan sebagai komponen yang sangat strategis dan diharapkan dapat memenuhi semua kebutuhan aktivitas Masyarakat [MUH17].

Hal ini belum terwujud karena berbagai hambatan yang ada. Rendahnya jumlah pengguna transportasi umum dipengaruhi oleh beberapa faktor, termasuk keterbatasan dalam ketersediaan serta distribusi informasi mengenai angkutan umum yang masih kurang memadai. Masyarakat cenderung mengandalkan informasi yang mereka ketahui sebelumnya atau bertanya kepada sopir angkutan umum dan orang di sekitar mereka. Perkembangan teknologi informasi memberikan peluang untuk menerapkan sistem informasi yang menyertakan data spasial. Dengan adanya data spasial tersebut, para pencari informasi tidak hanya akan memperoleh teks dan angka, tetapi juga gambaran tentang lokasi asal informasi tersebut. Salah satu aspek yang membuat pencarian rute lebih jelas adalah penerapan algoritma pencarian rute [FED10].

Dalam mengatasi isu transportasi umum di kota Bandung, telah dikembangkan sebuah situs web yang dapat membantu dalam pencarian rute angkutan kota dengan memberikan informasi yang lebih komprehensif, seperti Rute, Tarif, warna kendaraan, jarak, dan Gambar.

Mengingat banyaknya rute yang ada dan beragamnya jarak tempuh, penting untuk terlebih dahulu menentukan jalur yang paling efisien berdasarkan lokasinya. Menemukan rute terpendek sangat membantu dalam mengidentifikasi jarak minimum dan menemukan berbagai alternatif rute yang tersedia. Penelitian ini berfokus pada penentuan rute jalur terpendek melalui Metode BFS (Breadth First Search). Algoritma pencarian ini merupakan metode yang melakukan eksplorasi secara menyeluruh, yakni dengan mengunjungi node dalam urutan preorder, di mana suatu node dikunjungi terlebih dahulu sebelum melanjutkan ke semua node yang terhubung dengan node tersebut. Proses pencarian dilaksanakan dengan pendekatan Breadth First Search (BFS), yang diawali dari node root di tingkat teratas. Pencarian dimulai dari node pada level satu, dan akan terus berlanjut tanpa batas kedalaman hingga menemukan node yang diinginkan [ADE24].

1.2 Identifikasi Masalah

1. Bagaimana Algoritma *Breadth First Search* melakukan Pencarian Rute Angkot dikota Bandung?
2. Bagaimana Algoritma *Breadth First Search* menyajikan rincian informasi seperti rute, tarif, kode, warna, dan gambar angkot?
3. Bagaimana Algoritma *Breadth First Search* menyajikan peta yang mencakup titik-titik dari rute yang ada?

1.3 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah untuk Merancang Website yang memberikan Informasi Rute Angkot di kota Bandung menggunakan algoritma *Breadth First Search* (BFS) yang dapat memberikan informasi rute angkot secara detail termasuk rute yang dilewati, harga, kode, warna angkot, gambar pendukung, dan peta yang dibutuhkan selama perjalanan.

1.4 Lingkup Tugas Akhir

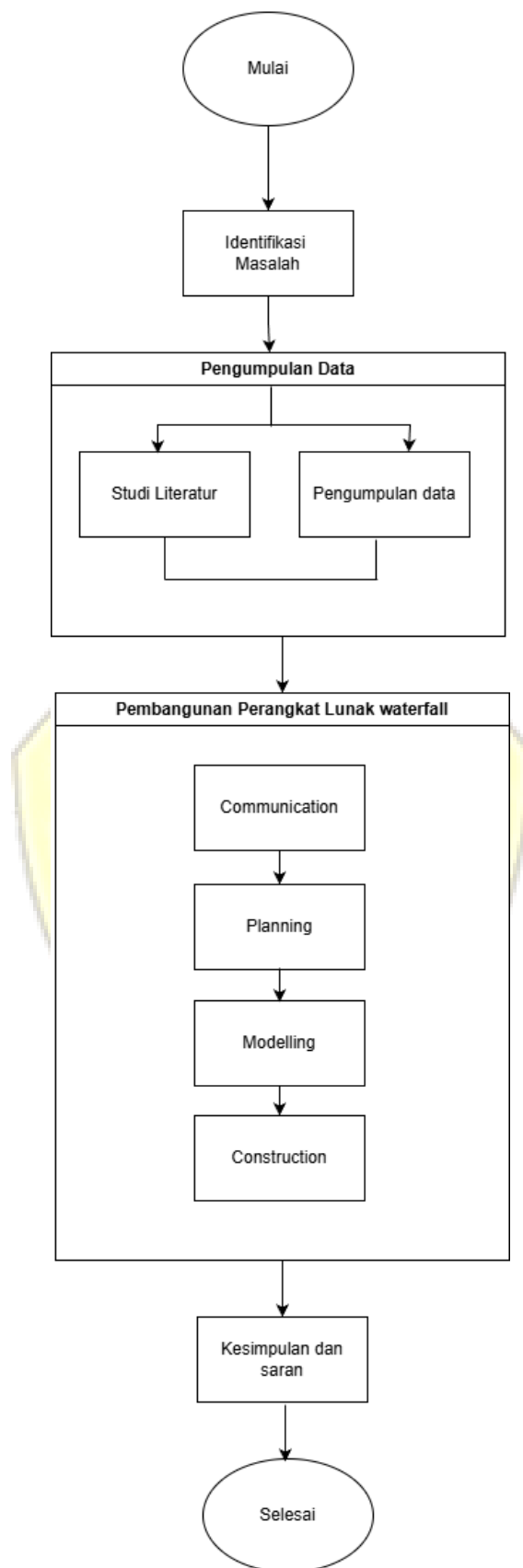
Dengan adanya batasan tersebut, penelitian ini difokuskan pada perancangan website informasi angkot di Kota Bandung dengan penerapan algoritma *Breadth First Search* di kota bandung sebagai berikut :

1. Penelitian ini berfokus pada pembuatan aplikasi berbasis website yang menyajikan informasi rute angkot di kota bandung.
2. Informasi yang ditampilkan berupa nama angkot, kode, tarif, jarak, rute yang dilewati dan map yang menampilkan titik titik rute.
3. Metode pencarian rute yang digunakan dalam penelitian ini adalah Algoritma *Breadth First Search* (BFS) untuk menentukan jalur terpendek berdasarkan jumlah titik yang dilalui.
4. Metodologi pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model Waterfall, tahapan Communication, Planning, Modeling, dan Construction.
5. Pengguna hanya dapat melakukan pencarian rute dari titik awal ke titik tujuan, serta melihat hasil rute yang direkomendasikan.

6. Pengembangan aplikasi dilakukan menggunakan Laravel 11 sebagai framework backend untuk pengelolaan data dan logika sistem.
7. Visualisasi peta dan rute angkot dalam aplikasi memanfaatkan Leaflet.js sebagai framework JavaScript untuk pemetaan.
8. MySQL digunakan sebagai sistem manajemen basis data untuk menyimpan, mengelola, dan mengorganisasi seluruh data yang dibutuhkan oleh aplikasi.
9. Wilayah penelitian dibatasi pada Kota Bandung sesuai dengan ketersediaan data rute angkot yang digunakan, sehingga hasil penelitian tidak mencakup wilayah di luar Kota Bandung.
10. Data nama angkot yang digunakan dalam penelitian ini diperoleh dari Dinas Perhubungan (Dishub) Kota Bandung sebagai sumber data resmi



1.5 Metodologi Tugas Akhir



Gambar 1. 1 Metodologi Tugas Akhir

Berikut Adalah penjelasan gambar 1.1 Metodologi Tugas akhir, Sebagai berikut;

1. Identifikasi Masalah

Pada tahap ini dilakukan proses pengamatan dan analisis awal untuk menentukan permasalahan yang akan diangkat sebagai topik penelitian. Permasalahan yang ditemukan menjadi dasar dalam penyusunan tugas akhir, sehingga fokus penelitian dapat diarahkan secara jelas sesuai dengan kondisi dan kebutuhan yang ada.

2. Pengumpulan Data

Pada tahap ini, data dikumpulkan melalui dua aktivitas, yaitu:

- Studi Pustaka, yang bertujuan untuk menganalisis teori dan metode dari berbagai jurnal yang relevan, sehingga dapat memperoleh dasar teori dan mengidentifikasi permasalahan yang ada.
- Pengumpulan informasi, yang dilakukan dengan mengedarkan kuesioner kepada para pengguna angkot dan meminta informasi mengenai rute angkot dari dinas perhubungan (dishub).

3. Perancangan Aplikasi

Pada tahap ini dilakukan proses perancangan dan pembangunan aplikasi informasi rute angkot dengan menggunakan metode *Waterfall*. Metode ini diterapkan secara sistematis dan berurutan, Adapun tahapan yang diikuti dalam metode *Waterfall* pada penelitian ini adalah sebagai berikut:

➤ Communication (Komunikasi)

Tahap ini merupakan proses awal dalam pengembangan sistem yang bertujuan untuk menggali kebutuhan pengguna. Kegiatan yang dilakukan meliputi wawancara dengan pihak Supir Angkot serta penyebaran kuesioner kepada pengguna Angkot untuk memperoleh informasi mengenai permasalahan, kebutuhan, dan harapan terhadap sistem yang akan dibangun.

➤ Planning (Perencanaan)

Pada tahap ini dilakukan perencanaan pengembangan sistem berdasarkan hasil dari tahap komunikasi. Perencanaan mencakup penentuan tujuan sistem, ruang lingkup penelitian, kebutuhan fungsional dan nonfungsional, serta penyusunan jadwal dan tahapan kerja yang akan dilaksanakan.

➤ Modelling (Pemodelan)

Tahap pemodelan berfokus pada perancangan sistem analisis dan antarmuka. Kegiatan yang dilakukan meliputi pembuatan desain arsitektur sistem, perancangan basis data, serta pemodelan menggunakan diagram seperti UML untuk menggambarkan alur proses, struktur data, dan interaksi antara pengguna dengan sistem.

➤ Construction (Pembangunan)

Tahap ini merupakan proses implementasi dari hasil perancangan yang telah dibuat. Kegiatan yang dilakukan meliputi penulisan kode program, integrasi basis data, serta penerapan algoritma yang digunakan.

4. Kesimpulan dan Saran

Pada tahap ini disajikan bagian akhir yang memuat rangkuman hasil penelitian beserta evaluasi terhadap sistem yang telah dibangun. Kesimpulan menggambarkan pencapaian tujuan penelitian dan menjawab permasalahan yang telah dirumuskan sebelumnya. Sementara itu, saran berisi rekomendasi yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan sistem maupun penelitian selanjutnya agar menjadi lebih optimal dan berkelanjutan.

1.6 Sistematika Penelitian

Bagian ini menguraikan isi dari berbagai bab yang ada dalam laporan akhir. Penjelasan ini bertujuan untuk memberikan pemahaman yang jelas mengenai struktur dan isi setiap bab dalam laporan akhir. Berikut adalah sistematika penulisan yang diterapkan :

BAB 1 PENDAHULUAN

Bab I berisi penjelasan umum mengenai usulan penelitian yang dilakukan dalam penyusunan tugas akhir. Bab ini mencakup latar belakang masalah, identifikasi masalah, tujuan penelitian, ruang lingkup tugas akhir, metodologi yang digunakan, serta sistematika penulisan laporan tugas akhir.

BAB 2 LANDASAN TEORI

Bab II membahas landasan teori yang relevan dan mendukung penelitian ini. Bab ini mencakup teori-teori yang digunakan serta tinjauan terhadap penelitian-penelitian sebelumnya yang berkaitan dengan topik penelitian.

BAB 3 SKEMA PENELITIAN

Bab III menjelaskan metodologi penelitian yang digunakan, meliputi tahapan penyelesaian tugas akhir, kerangka berpikir teoritis, serta skema analisis yang diterapkan.

BAB 4 ANALISIS DAN PERANCANGAN

Bab IV membahas analisis sistem yang berfokus pada identifikasi kebutuhan dan fitur yang diperlukan oleh pengguna. Selain itu, bab ini juga mencakup perancangan antarmuka pengguna UI untuk menghasilkan desain sistem yang mudah digunakan dan fleksibel.

BAB 5 IMPLEMENTASI

Bab V membahas implementasi sistem berdasarkan hasil perancangan pada bab sebelumnya, sesuai dengan metode yang digunakan dalam pengembangan website.

BAB 6 PENUTUP

Bab VI merupakan bab penutup yang berisi kesimpulan dari hasil penelitian berdasarkan permasalahan yang telah diidentifikasi serta keterkaitan seluruh tahapan penelitian yang telah dilakukan. Bab ini juga menyajikan saran yang diharapkan dapat menjadi bahan pertimbangan bagi pengembangan sistem maupun penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Teori Yang Digunakan

Teori yang digunakan dalam penelitian ini berkaitan dengan konsep-konsep dasar yang relevan dalam perancangan website rute angkot di Kota Bandung. Teori-teori tersebut berfungsi sebagai landasan ilmiah dalam proses perancangan dan implementasi aplikasi informasi rute angkutan kota (angkot), sehingga sistem yang dibangun memiliki dasar konseptual yang jelas dan terarah.

2.1.1 Website

Penemu website adalah Sir Timothy John “Tim” Berners-Lee, sedangkan website yang tersambung dengan jaringan, pertama kali muncul pada tahun 1991. Website adalah untuk mempermudah tukar-menukar dan memperbarui informasi kepada sesama peneliti di tempat bekerja. Website atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut hyperlink, sedangkan teks yang dijadikan media penghubung disebut hypertext. Dalam perkembangannya Web 2.0 diaplikasikan sebagai bentuk penyajian halaman web yang bersifat sebagai program dekstop pada umumnya seperti windows. Fungsi fungsi pada penerapannya sudah bersifat seperti dekstop, seperti drag and drop, auto-complete, serta fungsi lainnya. [FEB12].

Adapun Jenis-jenis web berdasarkan sifat atau stylenya yaitu:

1. Website Dinamis, merupakan sebuah website yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, NET dan memanfaatkan database MySQL atau MSSQL.
2. Website Statis, merupakan website yang contentnya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan database. Misalnya: web profile organisasi, dan lain-lain.

Fungsinya,website terbagi beberapa yaitu:

1. Personal website, website yang berisi informasi pribadi seseorang
2. Commercial website, website yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
3. Government website, website yang dimiliki oleh instansi pemerintahan, pendidikan yang bertujuan memberikan pelayanan kepada pengguna.
4. Non-Profit Organization website, dimiliki oleh organisasi yang bersifat non profit atau bisnis.

2.1.2 Angkutan Kota

Transportasi atau transportare berasal dari kata latin, di mana trans berarti sisi lain atau seberang, sementara portare berarti membawa atau mengangkut. Dari kedua istilah ini, transport dapat diartikan sebagai memindahkan sesuatu dari satu lokasi ke lokasi lainnya. Transportasi dapat dimanfaatkan sebagai sarana untuk meraih keuntungan finansial dalam berbagai bisnis dan interaksi sosial. Transportasi publik adalah fasilitas transportasi yang sangat penting untuk mendukung aktivitas serta mobilitas mayoritas masyarakat di area perkotaan. Meskipun sebuah kota sudah sangat maju, tetap saja angkutan umum masih diperlukan. Tujuan dari pengadaan transportasi publik adalah untuk menyediakan layanan angkutan yang aman, cepat, terjangkau, dan nyaman bagi masyarakat. Karena bersifat massal, terdapat keperluan akan kesamaan di antara penumpang dalam hal asal dan tujuan [SIN22].

Angkutan adalah proses pemindahan penumpang dan barang dari satu lokasi ke lokasi lain. Menurut peraturan, angkutan umum adalah angkutan yang memungut biaya dari pengguna, sementara rute adalah sekumpulan jalan yang secara berkesinambungan menghubungkan satu lokasi ke lokasi lainnya. Dalam penelitian ini, terdapat dua rute angkutan umum, yaitu rute angkot dan rute damri [SEL15].

Angkutan kota adalah angkutan dari suatu tempat ke tempat lain dalam wilayah kota dengan mempergunakan mobil bus umum dan atau mobil penumpang umum yang terikat dalam trayek tetap dan teratur. Tujuan utama keberadaan angkutan kota adalah untuk menyelenggarakan pelayanan yang baik dan layak bagi masyarakat. angkutan dari suatu tempat ke tempat lain dalam wilayah suatu kota dengan menggunakan mobil bis umum dan/atau mobil penumpang umum yang terikat pada trayek tetap dan teratur. Dapat juga angkutan kota berupa angkutan massal atau mass rapid transit yang dapat mengangkut penumpang dalam jumlah banyak dalam satu kali perjalanan. Mobil penumpang umum (MPU) adalah setiap kendaraan umum yang dilengkapi sebanyak banyaknya delapan tempat duduk, tidak termasuk tempat duduk pengemudi, dengan maupun tanpa baik perlengkapan pengangkutan bagasi [MSB17].

2.1.3 Algoritma Breadth First Search

Breadth First Search merupakan sebuah metode pencarian yang menggunakan pendekatan melebar dengan cara mengunjungi simpul secara urutan pre-order, artinya mengunjungi satu simpul terlebih dahulu sebelum melanjutkan ke semua simpul yang berdekatan dengannya. Setelah itu, simpul yang belum dikunjungi yang berdekatan dengan simpul-simpul yang telah dikunjungi akan diakses, dan proses ini akan terus berlanjut. Algoritma ini memerlukan antrian q untuk menyimpan simpul yang sudah dijelajahi. Simpul-simpul ini sangat penting sebagai referensi untuk menjelajahi simpul-simpul di sekitarnya. Setiap simpul yang sudah dikunjungi akan dimasukkan ke dalam antrian hanya satu kali. Selain itu, algoritma ini juga membutuhkan tabel boolean untuk merekam simpul yang telah dijelajahi sehingga tidak ada simpul yang dijelaskan lebih dari satu kali [DEL16].

Metode algoritma BFS memiliki karakteristik pencarian yang menyamping dan dimulai dari kondisi awal. Data yang pertama kali masuk akan menjadi yang pertama keluar. Dalam penerapannya pada pemrograman, digunakan struktur data yang disebut antrian, yang dikenal dengan istilah FIFO (First In, First Out), di mana elemen yang masuk terlebih dahulu akan keluar terlebih dahulu [YUL24].

Algoritma Breadth First Search (BFS) adalah teknik pencarian jalur yang melibatkan kunjungan ke setiap titik dengan cara mengakses node pada level awal terlebih dahulu. Selanjutnya, node-node pada level berikutnya akan dikunjungi secara berurutan. Jika titik tujuan masih belum ditemukan, proses ini dapat diulang di setiap cabang dari setiap titik hingga titik yang dicari berhasil ditemukan. Secara ringkas, algoritma BFS berfungsi untuk menjelajahi dan mencari seluruh node dalam struktur data baik itu tree ataupun graph. Algoritma BFS memerlukan antrian yang dikenal sebagai q, yang bertindak sebagai penyimpanan untuk node yang telah dikunjungi. Node-node ini penting sebagai acuan untuk mengunjungi node-node yang berdekatan. Setiap node yang dikunjungi hanya dimasukkan ke dalam antrian sekali. Hal ini dikarenakan algoritma BFS memerlukan tabel Boolean untuk menyimpan informasi tentang node yang telah dikunjungi, sehingga setiap node tidak dikunjungi lebih dari satu kali[SUT23].

Kelebihan Algoritma Breadth First Search

Berikut adalah beberapa kelebihan yang perlu diketahui mengenai algoritma pencarian breadth first search, sebagai Berikut;

1. Algoritma Breadth First Search Tidak akan menjumpai jalan buntu.
2. Algoritma Breadth First Search Dapat dipastikan menemukan solusi yang dicari.
3. Apabila terdapat satu solusi maka breadth first search dapat menemukannya.
4. Dapat menjamin menemukan solusi apabila memang ada solusinya dan solusi yang telah ditemukan pasti dianggap paling baik.

Kekurangan Algoritma Breadth First Search

Adapun kekurangan yang ada dalam breadth first search yang perlu dipahami, antara lain sebagai berikut.

1. Algoritma Breadth First Search Memerlukan waktu yang cukup lama untuk memeriksa semua node yang ada.
2. Apabila node jauh maka membutuhkan waktu yang cukup lama untuk memeriksa lintasan.
3. Membutuhkan memori atau tempat penyimpanan yang cukup besar, hal tersebut disebabkan oleh metode ini dapat memeriksa secara menyeluruh node yang ada dalam satu pohon.

Cara Kerja Algoritma Breadth First Search

1. Pertama – tama Menentukan Titik awal (Start) dan Titik tujuan (goal)
2. Masukkan Titik awal ke dalam antrian (queue)
3. Ambil node dari antrian awal, kemudian periksa kembali apakah node tersebut merupakan solusi.
4. Apabila node tersebut merupakan solusi, maka pencarian dapat diselesaikan dan hasil dapat dikembalikan.

5. Namun apabila node tersebut bukan merupakan solusi, maka dapat memasukkan semua node yang bersebelahan dengan node anak ke dalam antrian.
6. Setelah node masuk ke antrian dan di periksa, tandai node sebagai visited (sudah dikunjungi)
7. Apabila antrian kosong dan setiap simpul telah diperiksa, maka pencarian dapat diselesaikan dan hasil dapat dikembalikan sebagai solusi tidak ditemukan.
8. Langkah terakhir, ulangi kembali pencarian dari langkah kedua sampai titik tujuan.

Tabel 2. 1 Pseudocode Algoritma Breadth First Search

Pseudocode Algoritma Breadth First Search
<pre> PROCEDURE BFS(graph, start): Buat queue Q Buat set visited = kosong Enqueue start ke Q Tambahkan start ke visited WHILE Q tidak kosong: u = Dequeue Q Proses u // Kunjungi atau cetak u UNTUK setiap tetangga v dari u: JIKA v tidak dalam visited: Enqueue v ke Q Tambahkan v ke visited END WHILE END PROCEDURE </pre>

2.1.4 Graph

Graf adalah pelajaran fundamental dalam matematika diskrit dan ilmu komputer. Ini menggambarkan objek yang terpisah dan keterkaitan di antara objek-objek tersebut. Representasi graf secara visual ditunjukkan dengan menggunakan node, lingkaran, atau titik untuk melambangkan objek, sementara hubungan antar objek diwakili oleh garis atau sisi. Dalam artikel yang ditulis oleh Ida Bagus, dijelaskan bahwa graf terdiri dari sepasang himpunan vertex atau simpul dan edges atau sisi, di mana setiap sisi terhubung dengan satu atau dua simpul. Menurut artikel oleh Mira Kusmira, graf yang memiliki sisi (node) dan simpul (edge) pada graf [MAN22],

Dapat dibedakan menjadi dua kategori yaitu;

A. Graf Sederhana

Graf Sederhana adalah graf yang tidak memiliki simpul atau sisi berulang.

B. Graf Tak Sederhana

Graf Tak Sederhana adalah graf yang memiliki simpul atau sisi berulang. Ada dua jenis graf tak sederhana, yaitu: multigraph (graf ganda) dan pseudograph (graf semu).

Dalam graf, berdasarkan orientasi pada sisi, graf terbagi menjadi dua jenis yaitu:

- A. Graph Tidak Berarah Graf tidak berarah adalah sisi yang tidak memiliki orientasi tertentu. Urutan dari pasangan simpul dalam graf tidak menjadi hal yang penting.

- B. Graf Berarah Graf berarah adalah graf di mana setiap sisi memiliki orientasi tertentu. Sisi-sisi yang berarah ini biasanya disebut busur.

2.1.5 Trayek

Trayek/rute angkutan umum didefinisikan sebagai tempat-tempat dimana angkutan umum secara tetap melayani penumpang yaitu dengan menaikkan dan menurunkannya. Suatu rute biasanya merupakan suatu lintasan tetap dari angkutan umum yang melewati beberapa daerah, dimana angkutan umum secara rutin melayani penumpang dan dilain pihak calon penumpang menggunakan angkutan pada rute tersebut. Rute angkutan umum biasanya ditempatkan di lokasi dimana memang diperkirakan ada calon penumpang yang akan dilayani. Dalam suatu[MSB17].

Trayek Angkutan merupakan jalur untuk kendaraan umum atau rute untuk memberikan layanan transportasi orang menggunakan bus yang memiliki titik awal dan akhir perjalanan yang tetap, rute yang tetap, serta jadwal yang tetap atau tidak terjadwal. Sesuai dengan Peraturan Pemerintah No 41 Pasal 7 tahun 1993[ARY18], jaringan trayek terbagi menjadi :

- A. Trayek Antarkota Antar Provinsi adalah rute yang melintasi lebih dari satu bagian wilayah Provinsi Daerah Tingkat I.
- B. Trayek Antarkota Dalam Provinsi (AKDP) adalah rute yang menghubungkan antar Daerah Tingkat II di dalam satu wilayah Provinsi Daerah Tingkat I.
- C. Trayek Kota adalah rute yang sepenuhnya berlokasi dalam satu wilayah Kotamadya Daerah Tingkat II atau rute yang terletak di Daerah Khusus Ibukota Jakarta.
- D. Trayek Pedesaan adalah rute yang sepenuhnya berada di dalam satu wilayah Kabupaten Daerah Tingkat II.
- E. Trayek Pedesaan adalah rute yang sepenuhnya berada di dalam satu wilayah Kabupaten Daerah Tingkat II.
- F. Trayek Lintas Batas Negara adalah rute yang melewati batas negara.

Berdasarkan Keputusan Menteri Perhubungan Nomor KM. 35 Tahun 2003, trayek angkutan umum terdiri dari:

- A. Trayek Utama Trayek utama memiliki jadwal yang tetap dan teratur.

Trayek ini melayani angkutan antar kawasan utama, antar kawasan utama dan pendukung dengan ciri perjalanan ulang alik secara tetap.

- B. Trayek Cabang

Sama halnya dengan sistem pengoperasian pada trayek utama namun trayek cabang ini beroperasi pada kawasan pendukung, antara kawasan pendukung dan pemukiman.

- C. Trayek Ranting

Trayek ranting tidak memiliki jadwal yang tetap. Wilayah pelayanannya pada kawasan pemukiman penduduk. Sedangkan moda yang digunakan berupa mobil penumpang.

D. Trayek Langsung

Trayek langsung memiliki jadwal yang tetap. Melayani kawasan utama angkutan dengan antara kawasan pendukung dan kawasan pemukiman, dan berhenti pada tempat-tempat yang telah ditetapkan untuk angkutan kota untuk menaik turunkan pemunpang.

2.1.6 Waterfall

Model Waterfall, yang juga dikenal sebagai linear sequential model, merupakan model proses pengembangan perangkat lunak yang menerapkan pendekatan sistematis dan berurutan. Proses pengembangan dimulai dari spesifikasi kebutuhan oleh pelanggan dan dilanjutkan secara bertahap melalui perencanaan, pemodelan, konstruksi, hingga implementasi dan pemeliharaan sistem. Pressman menjelaskan bahwa model ini mengasumsikan alur kerja yang mengalir secara linear, di mana setiap fase harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase berikutnya. Model ini merupakan paradigma tertua dalam rekayasa perangkat lunak dan telah digunakan selama puluhan tahun dalam berbagai proyek pengembangan sistem. Model Waterfall umumnya digunakan ketika kebutuhan sistem telah dipahami dengan baik, jelas, dan relatif stabil, misalnya pada pengembangan ulang atau penyesuaian sistem yang sudah ada, seperti perubahan pada perangkat lunak akuntansi akibat regulasi pemerintah yang baru .

Berikut Tahapan Model Waterfall menurut Pressman:

1. Communication (Komunikasi)

Pada tahap ini dilakukan interaksi dengan pelanggan untuk memahami kebutuhan sistem. Kegiatan meliputi pengumpulan kebutuhan, identifikasi masalah, dan penentuan tujuan sistem.

2. Planning (Perencanaan)

Tahap ini mencakup estimasi sumber daya, penjadwalan proyek, dan perencanaan aktivitas pengembangan. Hasil tahap ini berupa rencana kerja proyek yang terstruktur.

3. Modeling (Pemodelan / Analisis dan Desain)

Kebutuhan yang telah dikumpulkan diterjemahkan ke dalam bentuk model analisis dan desain sistem, baik secara struktural maupun fungsional.

4. Construction (Konstruksi)

Pada tahap ini dilakukan proses pengkodean dan pengujian. Sistem mulai diwujudkan dalam bentuk perangkat lunak yang dapat dijalankan.

5. Deployment (Implementasi dan Pemeliharaan)

Perangkat lunak diserahkan kepada pengguna untuk digunakan. Pada tahap ini juga dilakukan perbaikan kesalahan dan dukungan berkelanjutan terhadap sistem.

Kelebihan Waterfall

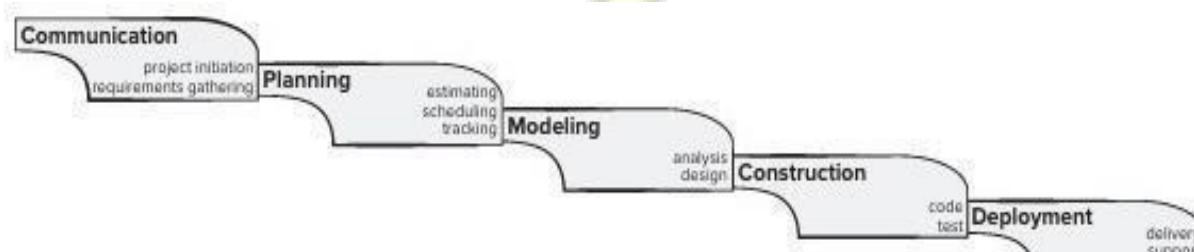
1. Model ini mudah dipahami dan direncanakan.
2. Cocok digunakan pada proyek kecil dengan kebutuhan yang sudah jelas.
3. Proses analisis dan pengujian bersifat lurus dan terstruktur.
4. Memberikan struktur kerja yang jelas dan sistematis.

5. Setiap tahap memiliki keluaran (dokumen) yang terdefinisi dengan baik.
6. Memudahkan pengelolaan proyek karena alur kerja dapat diprediksi.
7. Cocok untuk lingkungan yang membutuhkan dokumentasi formal dan kontrol ketat.

Kekurangan Waterfall

Pressman menekankan bahwa meskipun Waterfall memiliki struktur yang baik, model ini sering menimbulkan berbagai permasalahan dalam praktik, antara lain:

1. Proyek nyata jarang mengikuti alur sekuensial yang kaku.
2. Pelanggan sering kesulitan menyatakan seluruh kebutuhan secara lengkap di awal proyek.
3. Pelanggan harus menunggu lama karena perangkat lunak yang berfungsi baru tersedia di akhir proyek.
4. Kesalahan besar sering baru terdeteksi ketika sistem telah hampir selesai.
5. Tidak mampu mengakomodasi perubahan dengan baik.
6. Pengujian dilakukan pada tahap akhir proses.
7. Persetujuan pelanggan baru terjadi di akhir proyek.



Gambar 2. 1 Model Waterfal Presman

2.1.7 Framework

Framework merupakan suatu kerangka kerja dalam pengembangan perangkat lunak. Framework juga dapat diartikan sebagai kumpulan skrip, terutama berupa class dan function, yang dirancang untuk membantu developer atau programmer dalam menanggapi berbagai permasalahan pemrograman, seperti koneksi ke basis data, pemanggilan variabel, pengelolaan file, dan kebutuhan lainnya. Dengan adanya framework, pekerjaan developer menjadi lebih terfokus, terstruktur, dan lebih cepat dalam membangun sebuah aplikasi. Framework menyediakan komponen pemrograman yang siap digunakan kembali kapan saja, sehingga programmer tidak perlu menulis ulang skrip yang sama untuk tugas yang serupa. Sebagai contoh, dalam pembuatan aplikasi web berbasis Ajax, setiap proses biasanya memerlukan *XML HTTP Request*. Kehadiran framework telah mempermudah hal tersebut dengan menyediakan objek khusus yang siap digunakan untuk operasi Ajax berbasis PHP. Contoh tersebut hanyalah sebagian kecil, karena pada kenyataannya cakupan dan manfaat framework jauh lebih luas dalam mendukung pengembangan aplikasi secara efisien[YUD19].

Framework adalah struktur yang tersusun dari serangkaian kode generik dan berfungsi untuk mengembangkan sistem dan aplikasi. Kerangka kerja ini berperan sebagai template atau model yang menyediakan fungsi cerdas dan elemen struktur standar untuk memudahkan pekerjaan developer. Karena menjadi semacam “fondasi” dalam mengerjakan proyek, framework membantu developer agar

tidak perlu memulai proses development dari nol. Ada berbagai komponen siap pakai dan solusi yang mudah disesuaikan dalam framework untuk menyederhanakan proses pengembangan[FAR25].

Kegunaan dan Fungsi Framework

Berikut adalah beberapa kegunaan utama framework yang membuatnya banyak dimanfaatkan dalam pemrograman adalah:

1. Memudahkan proses pengembangan software atau aplikasi. Dengan menyediakan struktur dasar yang memudahkan proses pemrograman sistem, framework sangat membantu dalam hal waktu dan keamanan.
2. Framework yang umumnya bersifat open-source biasanya sering menjalani pengujian dan pengoptimalan secara teratur.
3. Framework juga punya konsistensi tinggi karena merupakan coding yang diuji secara ekstensif dan sudah terbukti berhasil.
4. Menentukan dan menstandarkan langkah-langkah pemrograman
5. Meningkatkan konsistensi dalam proses pengembangan dan dalam aplikasi yang dibuat

2.1.8 Laravel

Laravel adalah framework PHP open source yang kuat dan mudah dipahami, hal ini mengikuti pola desain model-view-controller. Laravel menggunakan kembali komponen kerja berbeda yang ada untuk membantu dalam pengembangan membuat aplikasi web. Laravel juga menawarkan serangkaian fungsi yang menggabungkan fitur dasar kerangka PHP seperti Codeigniter, Yoo dan Bahasa pemrograman lainnya seperti Ruby on Rails. Laravel merupakan sebuah framework PHP yang bersifat open source dan menggunakan pola arsitektur Model-View-Controller (MVC). Laravel dikembangkan oleh Taylor Otwell dengan tujuan untuk mempermudah proses pengembangan aplikasi web melalui sintaks yang sederhana, ekspresif, dan terstruktur. Laravel menyediakan berbagai komponen bawaan seperti routing, authentication, middleware, ORM (Eloquent), templating engine (Blade), serta sistem manajemen basis data yang terintegrasi. Dengan adanya fitur-fitur tersebut, pengembang dapat membangun aplikasi web yang aman, terorganisir, dan mudah dipelihara. Framework ini dirancang untuk mempercepat proses pengembangan aplikasi tanpa mengurangi kualitas sistem, karena Laravel mendorong penggunaan praktik pengkodean yang rapi, modular, dan sesuai standar pengembangan perangkat lunak modern. [ADM23].

➤ Keunggulan Laravel

Sejak di-launching pada tahun 2011 lalu, Laravel kini menjadi salah satu library yang cukup populer. Sudah banyak developer yang jatuh hati dengan kemudahan dan keunggulannya tetapi tetap saja Laravel masih belum sempurna dan masih terdapat beberapa kekurangan dalam implementasinya, berikut merupakan beberapa keunggulan dan kekurangan yang dimiliki oleh laravel

- 1 Pengkodean biasanya menjadi salah satu hal yang sangat rumit, tetapi Laravel membuat beberapa tugas menjadi lebih mudah, hal ini dikarenakan Laravel memiliki library yang luas yang memuat fitur-fitur yang fungsionalitas yang telah deprogram sebelumnya.

- 2 Laravel merupakan salah satu framework scalable yang memudahkan pengembang menggunakannya untuk aplikasi web berukuran kecil dan menengah
- 3 Laravel memiliki sistem kontrol akses bawaan yang aman dan juga menyediakan mekanisme tanggung yang memungkinkan pengembang menangani bug atau masalah dengan mudah
- 4 Migrasi data jauh lebih sederhana jika menggunakan Laravel, karena migrasi data dilakukan dengan meminimalisir waktu daripada biasanya.

Karakteristik Laravel

1. Menggunakan arsitektur MVC (Model–View–Controller) untuk memisahkan logika bisnis, tampilan, dan pengolahan data.
2. Memiliki routing system yang fleksibel dan mudah dikonfigurasi.
3. Menyediakan ORM Eloquent untuk pengelolaan basis data secara objektif.
4. Mendukung migration dan seeding database untuk pengelolaan skema data.
5. Memiliki sistem keamanan bawaan seperti proteksi CSRF, enkripsi, dan autentikasi.
6. Menyediakan template engine Blade untuk membangun antarmuka pengguna secara efisien.
7. Mendukung pengembangan berbasis API dan aplikasi modern berbasis web.



Gambar 2. 2 Laravel

2.1.9 Tailwind CSS

Tailwind CSS merupakan framework berbasis utility-first yang memungkinkan pengembang mengatur desain langsung melalui kelas utilitas pada elemen HTML. Menurut pendekatan ini memberikan fleksibilitas tinggi dalam membuat antarmuka unik tanpa terikat gaya bawaan. Sementara[14]menekankan bahwa Tailwind lebih efisien dalam membangun prototipe modern dan mudah diintegrasikan dengan kerangka kerja JavaScript[MIF25].

Tailwindcss Yaitu framework CSS berbasis utility-first, yang menyediakan kelas-kelas CSS siap pakai yang dapat digunakan langsung untuk mengatur tampilan antarmuka pengguna. Berbeda dengan framework CSS lain yang biasanya menyediakan komponen dengan style siap pakai, Tailwind memberi keleluasaan untuk mengatur setiap elemen dari awal. Pendekatan ini memudahkan pengembang menciptakan desain yang konsisten dan unik dengan kode CSS yang minimal[COD25].

Fungsi Tailwind CSS

1. Mengurangi penulisan CSS khusus: developer bisa mengatur tampilan elemen menggunakan kelas-kelas yang sudah ada langsung di HTML. Mereka juga dapat membuat desain kustom tanpa perlu menulis CSS.
2. Bisa menjaga ukuran file CSS tetap kecil: jika tidak menggunakan framework seperti Tailwind, developer harus terus menulis CSS saat menambahkan fitur dan komponen baru. Akibatnya, file CSS akan semakin besar dan berat.
3. Tidak perlu menciptakan nama kelas sendiri: developer bisa memilih kelas-kelas dari sistem desain dengan nama yang sudah ditentukan sebelumnya. Mereka tidak perlu bingung mencari nama kelas untuk style dan komponen tertentu, atau bahkan mengingat nama-nama yang rumit.
4. Bisa melakukan perubahan dengan lebih aman: dalam pendekatan tradisional, saat melakukan perubahan pada CSS kemungkinan akan merusak halaman situs secara keseluruhan. Namun, kelas utilitas dalam HTML bersifat lokal yang dapat diubah tanpa khawatir merusak hal lain di website.



Gambar 2. 3 TailwindCSS

2.1.10 Leaflet.js

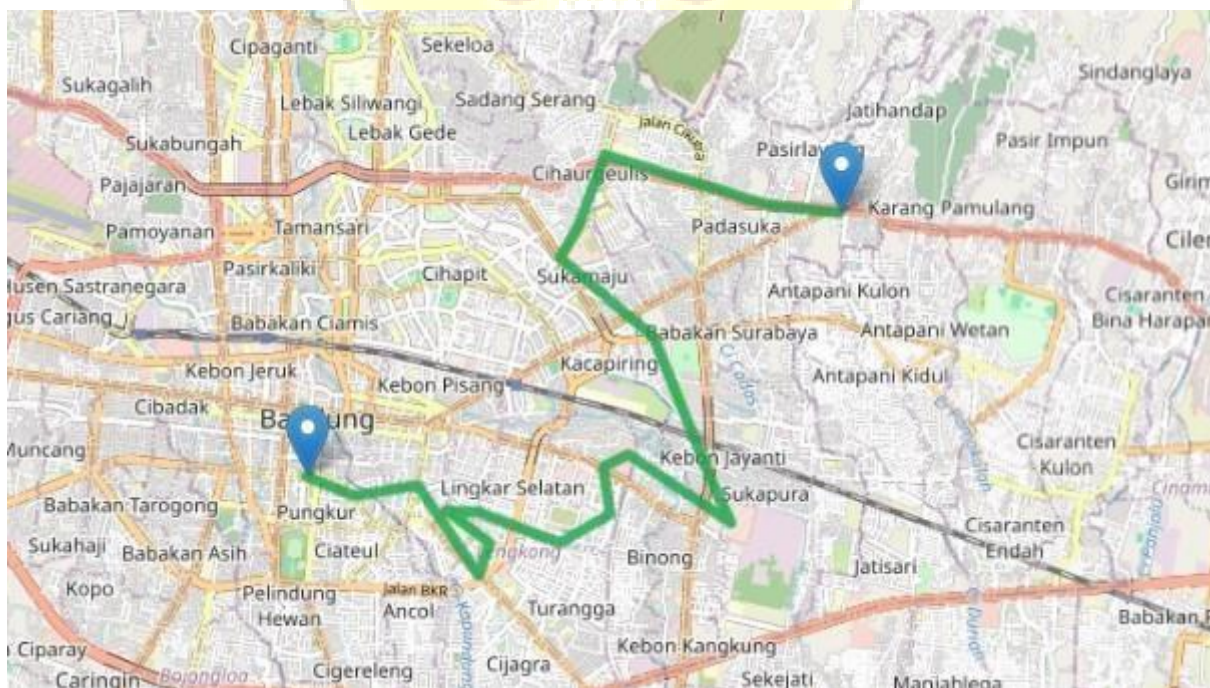
LeafletJS adalah sebuah pustaka (library) JavaScript yang ringan dan open-source yang digunakan untuk membuat peta interaktif berbasis web. Leaflet dirancang untuk memiliki performa yang tinggi dan ramah pengguna, terutama di perangkat seluler. Pustaka ini menyediakan berbagai fitur untuk membuat dan mengatur peta, menambahkan marker, pop-up, dan layer, serta berinteraksi dengan elemen-elemen peta. Leaflet awalnya dibuat oleh Volodymyr Agafonkin, namun kini dikembangkan oleh komunitas kontributor yang besar. API LeafletJS dirancang agar intuitif dan mudah dipelajari, sehingga pengembang dapat dengan cepat mengimplementasikan peta interaktif tanpa perlu memahami banyak konsep yang kompleks. Dukungan kepada banyak jenis layer, termasuk tile layers (misalnya dari OpenStreetMap atau layanan peta lainnya), marker, polylines, polygons, dan lain-lain memungkinkan fleksibilitas yang tinggi dalam visualisasi data. Geografis[BAY22].

Leaflet.js merupakan sebuah pustaka JavaScript yang dibuat khusus untuk membuat peta interaktif dalam aplikasi web. Pustaka ini difokuskan pada kesederhanaan, performa, dan modularitas. Dengan memanfaatkan Leaflet.js, para pengembang bisa dengan mudah menyisipkan peta interaktif ke dalam aplikasi web mereka, lengkap dengan berbagai fitur yang dapat diubah sesuai kebutuhan. Berikut adalah fitur utamanya;

1. Ringan dan Modular: Leaflet.js dibuat untuk memiliki ukuran kecil, yaitu hanya sekitar 38 KB saat diperkecil (sebelum kompresi gzip). Hal ini menjadikannya cepat dimuat dan bisa digunakan di berbagai jenis perangkat dan jaringan.
2. Kustomisasi yang Fleksibel: Para pengembang dapat dengan mudah menyesuaikan tampilan peta, seperti mengubah layout, warna, dan gaya agar sesuai dengan keperluan aplikasi mereka. Leaflet.js menawarkan API yang mudah dipahami untuk mengatur elemen seperti penanda, polyline, polygon, dan lapisan lainnya.
3. Interaktivitas Tinggi: Leaflet.js memungkinkan pengguna berinteraksi dengan peta melalui berbagai gerakan seperti zoom, pan, dan klik. Selain itu, para pengembang dapat menambahkan kontrol tambahan seperti kontrol zoom, peta dasar yang beragam, dan alat untuk pengukuran.

Leaflet.js banyak digunakan dalam beragam aplikasi web yang membutuhkan elemen peta, antara lain:

1. Aplikasi Geospasial: Untuk menampilkan data geospasial dan melakukan analisis spasial.
2. Pengembangan Web GIS: Membuat Sistem Informasi Geografis (SIG) dalam platform web.
3. Aplikasi Pariwisata: Menyediakan peta interaktif untuk membantu wisatawan dalam navigasi.
4. Platform Logistik: Untuk memantau dan mengatur jalur logistik.
5. Aplikasi Perencanaan Kota: Untuk memetakan tata kota dan merencanakan penggunaan ruang.



Gambar 2. 4 Contoh Map Leaflet.js

2.2 Penelitian Terdahulu

Tabel 2. 2 Penelitian Terdahulu

No	Peneliti	Judul	Kesimpulan	Alasan digunakan
1.	Delima Zai, Haeni Budiati, Sunneng Sandino Berutu	Simulasi Rute Terpendek Lokasi Pariwisata Di Nias Dengan Metode Breadth First Search Dan Tabu	Dapat ditarik kesimpulan bahwa aplikasi untuk mencari rute terpendek bisa diterapkan dan dimanfaatkan di pulau Nias, serta program simulasi yang dibuat mampu menampilkan jalur yang paling efisien.	Sebagai referensi penggunaan algoritma Breadth First Search dalam mengimplementasikan pada pembuatan pencarian rute
2.	Ade Zulkarnain, Munjiat Setiani Asih	Aplikasi Menemukan Rute Terdekat Lokasi Tambal Ban Menggunakan Algoritma Breadth First Search (BFS)	Dapat disimpulkan bahwa merancang sistem dengan menerapkan algoritma bfs ke dalam sebuah sistem dapat menghasilkan aplikasi yang mampu menemukan rute terdekat untuk tambal ban. Di samping itu, program sistem yang dihasilkan sangat efisien dan dapat membantu pengguna dalam menemukan rute tambal ban terdekat di sekitar mereka.	Sebagai referensi penggunaan algoritma Breadth First Search dalam mengimplementasikan pada pembuatan pencarian rute
3.	Almeiza Arvin Muzaki	Implementasi Algoritma BFS dan DFS dalam Pencarian Rute Angkutan Umum di Bandung Utara	Kesimpulan dari pengamatan yang kedua berfokus pada efektivitas penggunaan BFS dan DFS yang telah diamati. Algoritma BFS memberikan rute yang selalu lebih singkat daripada algoritma DFS di semua sampel yang diuji. Hal ini sejalan dengan karakteristik BFS yang menjelajahi secara luas, sehingga memastikan ditemukan jalur terpendek dalam sebuah graf yang lengkap. Sebaliknya,	Sebagai referensi penggunaan algoritma Breadth First Search dalam mengimplementasikan pada pembuatan pencarian rute
4.	Feddy Setio Pribadi dan Anggraini mulwinda	Pencarian Rute Terpendek dengan menggunakan Algoritma Depth First, Breadth First dan Hill climbing(Study Comparative)	Sebagai kesimpulan, algoritma yang paling unggul untuk menemukan rute terpendek dan paling efisien yang diterapkan pada jalur transportasi yang ada adalah algoritma Pencarian Pertama Lebar (Breadth First Search) dan algoritma Pendakian Bukit (Hill Climbing).	Sebagai referensi penggunaan algoritma Breadth First Search dalam mengimplementasikan pada pembuatan pencarian rute

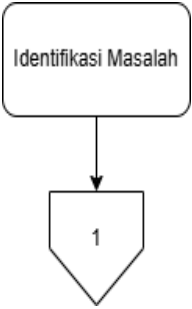
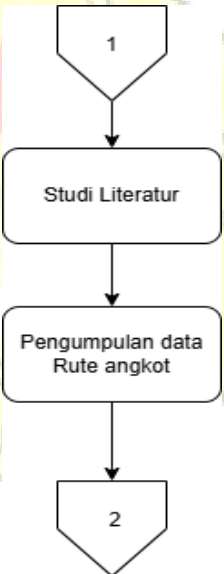
5.	Indah Adinda Marleni Aris Gunaryati	Presensi Karyawan Berbasis Web dengan Fitur Lokasi Leaflet JS menggunakan Laravel	Studi tentang Kehadiran Karyawan Berbasis Web dengan Fitur Lokasi Leaflet JS telah berhasil menciptakan sebuah aplikasi yang menawarkan berbagai keuntungan bagi organisasi. Dengan sistem yang	Sebagai referensi penggunaan Leaflet.js dalam menggunakan tampilan peta yang efisien
----	--	---	---	--



BAB 3 SKEMA PENULISAN

3.1 Alur Penyelesaian

Tabel 3. 1 Alur Penyelesaian

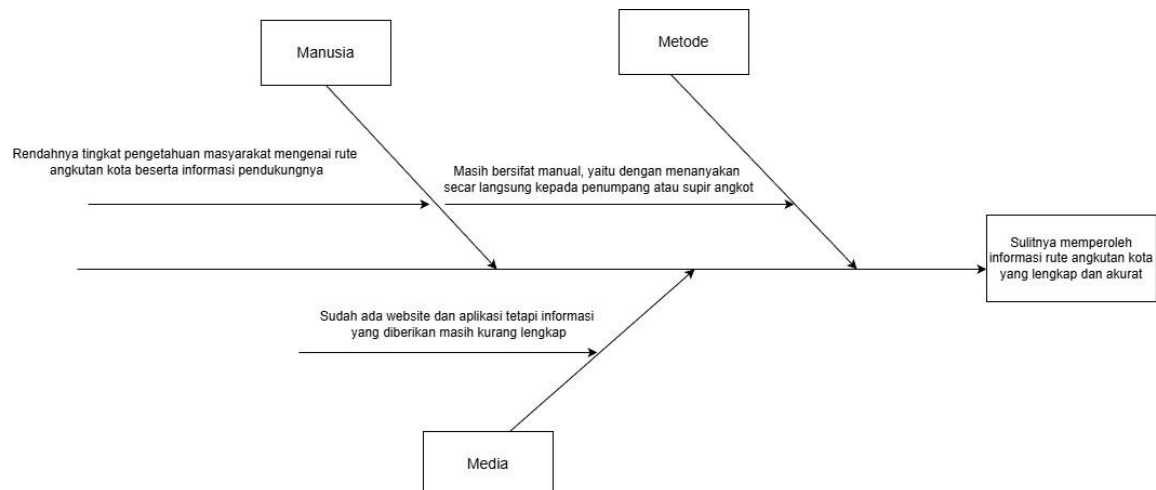
No	Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
1.	<p>Tahap 1: Mengidentifikasi masalah Tugas Akhir</p> <p>Hasil: Masalah teridentifikasi setelah melalui analisis pengumpulan data</p> <p>Kontribusi: Memastikan masalah pada tugas akhir ini lebih terarah dan tidak menyimpang dari tujuan</p>	 <pre> graph TD A[Identifikasi Masalah] --> B{1} </pre>	<p>Identifikasi Masalah Tugas Akhir</p>
2.	<p>Tahap 2: Studi literatur dengan menganalisis jurnal yang relevan dengan topik tugas akhir ini</p> <p>Hasil: Mendapatkan pemahaman lebih mengenai topik yang di ambil mengenai pencarian rute terpendek</p> <p>Kontribusi: Menyediakan dasar pengetahuan atau pemahaman dari jurnal yang relevan dengan topik yang di ambil</p>	 <pre> graph TD A{1} --> B[Studi Literatur] B --> C[Pengumpulan data Rute angkot] C --> D{2} </pre>	<p>Salah satu contoh Penerapan Beberapa jurnal Algoritma Breadth First Search "Aplikasi Menemukan Rute Terdekat Lokasi Tambal Ban Menggunakan Algoritma Breadth First Search (BFS)"</p> <p>Penerapan Leaflet.js "Presensi Karyawan Berbasis Web dengan Fitur Lokasi Leaflet JS menggunakan Laravel"</p>

<p>3.</p>	<p>Tahap 3 : Pembangunan Perangkat Lunak dengan acuan metode waterfall</p> <p>Hasilnya : Membuat website rute pencarian angkot yang memiliki informasi gambar, warna, dan tarif angkot dengan menggunakan metode waterfall</p> <p>Kontribusi : Membantu untuk memudahkan pengguna untuk mengakses informasi yang jelas tentang rute angkot berupa gambar, warna, dan tarif dengan lebih mudah digunakan</p>	<pre> graph TD A{{2}} --> B[Communication] B --> C[Planning] C --> D[Modeling] D --> E[Construction] E --> F{{3}} </pre>	<p>Software Engineering A pratitioner's Approach " Roger s pressman</p>
<p>4.</p>	<p>Tahap 4 : Kesimpulan dan saran</p> <p>Hasil: Memberikan kesimpulan dan saran pada tugas akhir</p> <p>Kontribusi : Memberikan kesimpulan dan saran pada tugas akhir</p>	<pre> graph TD A{{3}} --> B[Kesimpulan] B --> C[Saran] </pre>	

3.2 Perumusan Masalah

Perumusan masalah adalah Menyusun masalah yang akan di teliti secara jelas dan terarah. Dengan tujuan agar dapat menemukan solusi yang tepat, efektif dan sesuai dengan permasalahan yang ada. Pada perumusan masalah terdapat beberapa bagian, yaitu ;

3.2.1 Analisis Sebab Akibat



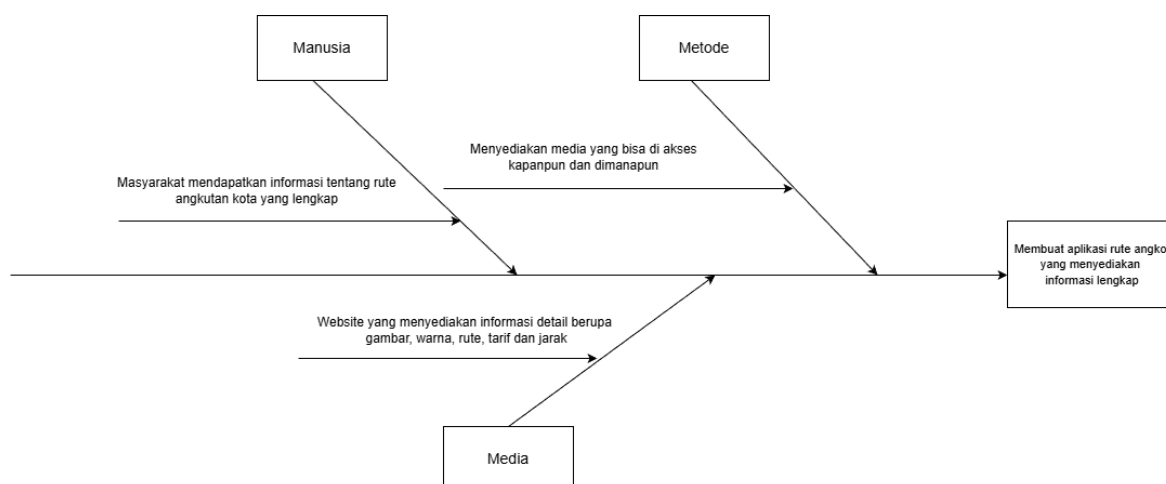
Gambar 3. 1 Analisis Sebab Akibat

Berdasarkan Gambar 3.2.1 di atas deskripsi analisis sebab akibat sebagai berikut ;

1. Manusia
Penyebab masalah ini pada manusia masih kurang pengetahuan atau informasi mengenai rute angkot dan informasi lainya seperti gambar per rute, tarifnya dan warna angkot.
2. Media
Masalah pada medianya adalah Sudah ada media seperti website dan aplikasi namun informasi yang diberikan masih kurang lengkap.
3. Metode
Metode agar kita mendapatkan informasi rute angkot masih manual yaitu dengan bertanya kepada orang lain atau supir angkot.

3.2.2 Analisis Solusi Masalah

Analisis sebab akibat adalah cara berpikir atau metode yang digunakan untuk mencari tahu kenapa suatu masalah bisa terjadi (sebab) dan apa dampaknya (akibat). Tujuan utamanya adalah untuk menemukan akar masalah, bukan cuma melihat gejala di permukaan. Berikut ini merupakan analisis sebab akibat pada penelitian tugas akhir :



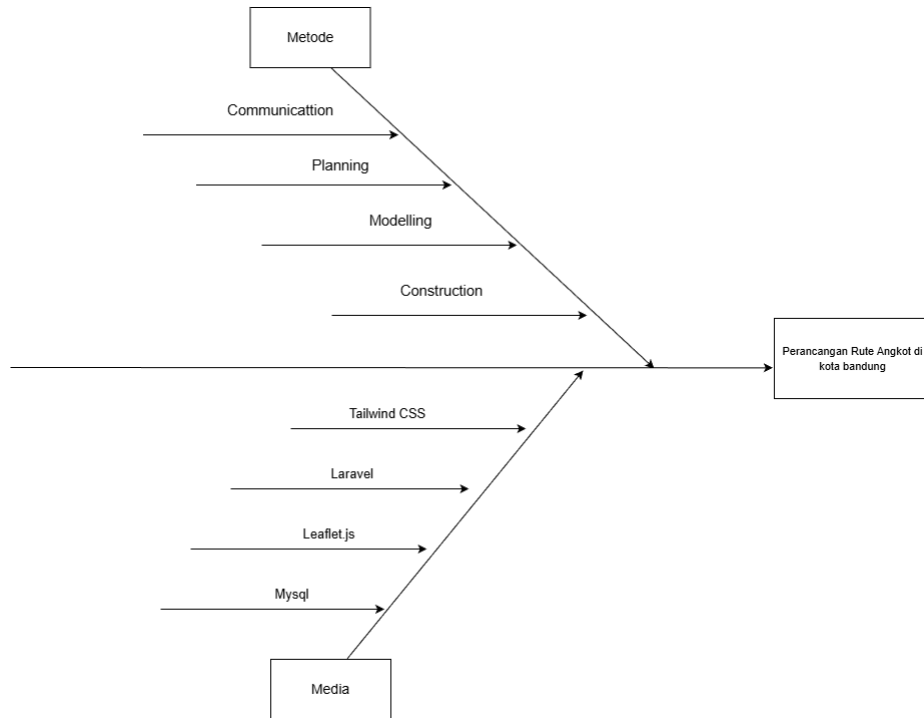
Gambar 3. 2 Analisis Solusi Masalah

Berdasarkan dari analisis solusi masalah pada gambar di atas, Berikut adalah deskripsi dari penyebab masalah yang terjadi pada tugas akhir.

1. Manusia
Masyarakat mendapatkan informasi tentang rute angkot yang akurat beserta detail informasi berupa gambar, warna dan tarif angkot.
2. Media
Membuat media berupa website yang lebih detail dalam menyediakan informasi.
3. Metode
Menyediakan media yang dapat diakses kapan pun dan di mana pun dalam bentuk website

3.2.3 Kerangka Berpikir Teoritis

Kerangka berpikir teoritis Adalah menggambarkan hubungan antara konsep-konsep yang digunakan dalam penelitian. Melalui kerangka teoritis, peneliti dapat menyusun dasar pemikiran yang sistematis sebagai landasan dalam menganalisis permasalahan. Adapun gambaran kerangka teoritis dapat dilihat pada penjelasan berikut :



Gambar 3. 3 Kerangka Berpikir Teoritis

Berdasarkan Gambar 3.3, berikut adalah penjabaran Kerangka Teoritis pada proyek akhir ini.

1. Metode

Pendekatan yang digunakan adalah waterfall.

2. Media

Menggunakan Tailwind CSS dan Laravel, leaflet.js (untuk memvisualisasikan peta), serta MySQL sebagai basis data.

BAB 4

ANALISIS DAN PERANCANGAN

4.1 Analisis Perancangan

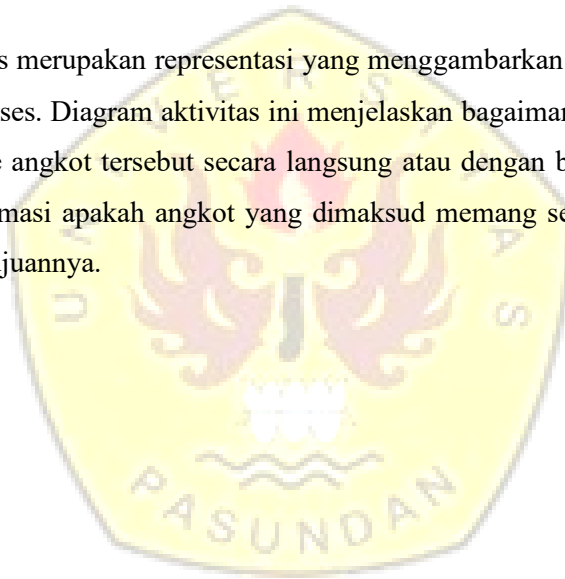
Analisis perangkat lunak merupakan langkah pertama dalam proses pengembangan sistem. Pada langkah ini, perhatian utama adalah untuk memahami dengan jelas masalah yang ingin diatasi oleh perangkat lunak, sembari mengenali secara rinci kebutuhan fungsional dan non-fungsional, serta berbagai batasan yang perlu diperhatikan.

4.1.1 Model Analisis

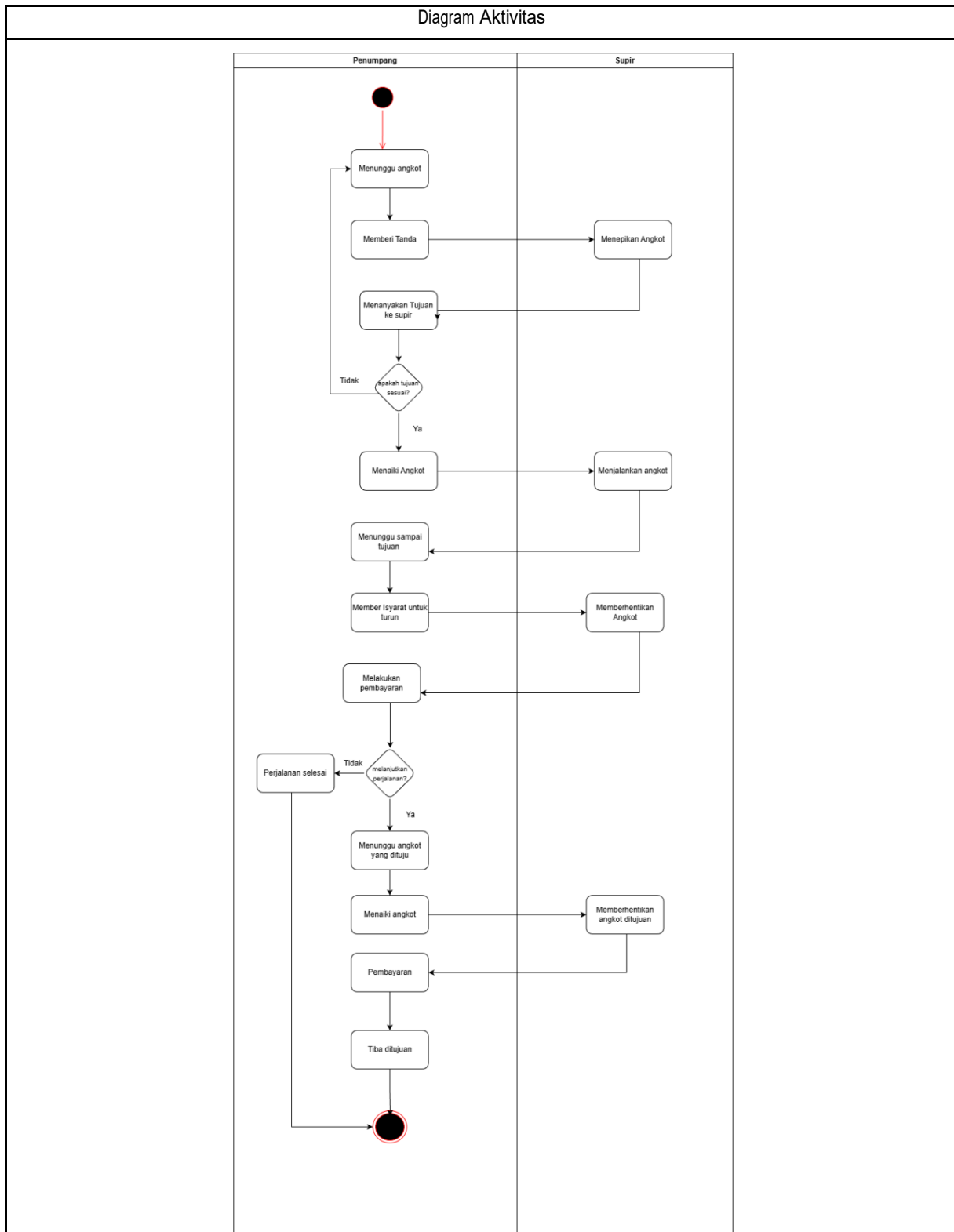
Model analisis merupakan gambaran yang tersusun dengan baik dan lengkap dari semua kebutuhan dan kendala yang harus dipenuhi oleh suatu aplikasi perangkat lunak. Di fase ini, sangat penting untuk menjamin bahwa perangkat lunak yang akan dibuat benar-benar sesuai dengan kebutuhan dan harapan, sebelum tahap pengembangan yang sesungguhnya dimulai.

4.1.1.1 Diagram Aktivitas

Diagram Aktivitas merupakan representasi yang menggambarkan langkah-langkah atau urutan peristiwa dalam suatu proses. Diagram aktivitas ini menjelaskan bagaimana seseorang naik angkot dan mencari tahu tentang rute angkot tersebut secara langsung atau dengan bertanya kepada sopir angkot untuk mendapatkan informasi apakah angkot yang dimaksud memang sesuai dengan rute yang ingin diambil demi mencapai tujuannya.



Tabel 4. 1 Diagram Aktivitas



Gambar 4. 1 Diagram Aktivitas

Diagram aktivitas tersebut menggambarkan alur penggunaan angkot yang melibatkan dua aktor, yaitu penumpang dan supir, dimulai dari penumpang menunggu angkot, memberikan tanda, serta menyampaikan tujuan kepada supir. Setelah tujuan dinyatakan sesuai, penumpang naik angkot dan supir menjalankan kendaraan hingga mencapai tujuan. Penumpang kemudian memberi isyarat untuk turun,

supir menghentikan angkot, dan penumpang melakukan pembayaran sehingga perjalanan selesai. Apabila tujuan belum tercapai atau memerlukan pergantian angkot, penumpang akan menunggu angkot lanjutan, naik kembali, dan melakukan pembayaran hingga akhirnya tiba di tujuan akhir.

4.1.1.2 Identifikasi User Requirement

User requirement (kebutuhan pengguna) tentang apa yang diinginkan atau dibutuhkan oleh pengguna dari sebuah sistem, dan bagaimana sistem tersebut harus berperilaku untuk membantu mereka mencapai tujuan tertentu.

Tabel 4. 2 Identifikasi User Requirement

Kode	User Requirement
UR - 01	User Requirement ingin dapat melakukan pencarian angkot berdasarkan titik awal dan titik tujuan
UR - 02	User Requirement ingin dapat melihat detail informasi angkot berupa rute, tarif, kode, warna, gambar dan lainnya berdasarkan titik awal dan titik tujuan
UR - 03	User Requirement Ingin dapat melihat daftar angkot di kota bandung

4.1.1.3 Identifikasi Software Requirement

Software requirement adalah deskripsi lengkap dan terperinci tentang apa yang harus bisa dilakukan sebuah sistem perangkat lunak. Ini mencakup semua fungsi, fitur, performa, dan batasan yang harus dipenuhi oleh perangkat lunak tersebut.

Tabel 4. 3 Identifikasi Software Requirement

Kode	User Requirement	Kode	Software Requirement
UR - 01	User Requirement ingin dapat melakukan pencarian angkot berdasarkan titik awal dan titik tujuan	SR - 01	Perangkat Lunak harus dapat mencari angkot berdasarkan titik awal dan titik tujuan
UR - 02	User Requirement ingin dapat melihat detail informasi angkot berupa rute, tarif, kode, warna, gambar dan lainnya berdasarkan titik awal dan titik tujuan	SR - 02	Perangkat Lunak harus dapat menampilkan detail informasi berupa rute, tarif, kode, warna, gambar dan lainnya berdasarkan titik awal dan titik tujuan
UR - 03	User Requirement Ingin dapat melihat daftar angkot di kota bandung	SR - 03	Perangkat Lunak harus dapat menampilkan daftar angkot di kota bandung

4.1.1.4 Function Requirement

Functional adalah daftar spesifik tentang apa saja yang harus bisa dilakukan oleh sebuah sistem perangkat lunak. Ini menjelaskan fungsi-fungsi inti dan perilaku khusus yang diharapkan dari sistem tersebut.

Tabel 4. 4 Function Requirement

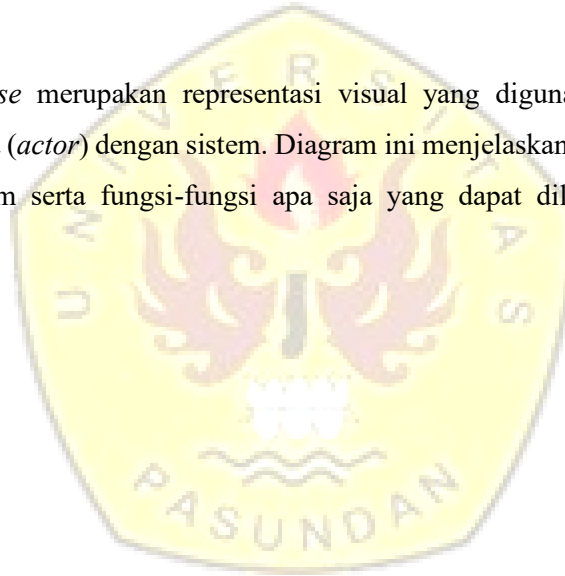
Kode	Kebutuhan Fungsional	Ketreneragan
FR - 01	Perangkat Lunak harus dapat mencari angkot berdasarkan titik awal dan titik tujuan	Perangkat Lunak Harus bisa mencari rute angkot berdasarkan titik awal dan titik tujuan yang diminta user atau pengguna
FR - 02	Perangkat Lunak harus dapat menampilkan detail informasi berupa rute, tarif, kode, warna, gambar dan lainnya berdasarkan titik awal dan titik tujuan	Perangkat Lunak Harus bisa menampilkan detail informasi berupa rute, tarif, kode, warna, gambar dan lainnya berdasarkan titik awal dan titik tujuan yang diminta user atau pengguna
FR - 03	Perangkat Lunak harus dapat menampilkan daftar angkot di kota bandung	Perangkat Lunak Harus bisa menampilkan daftar rute angkot di kota bandung

4.1.2 Model Use Case

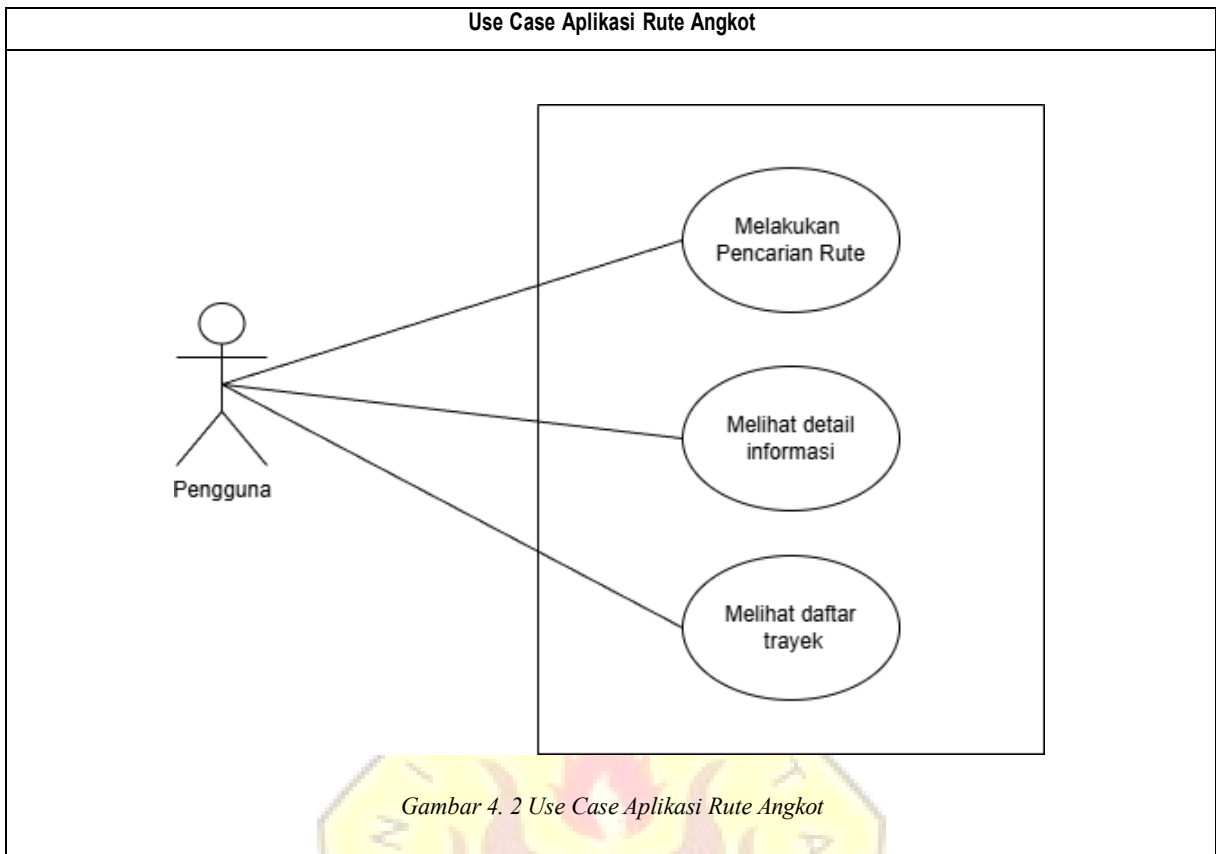
Model Use Case adalah metode visual untuk menampilkan fungsi penting suatu sistem dari sudut pandang pengguna. Ini berfungsi sebagai panduan yang menunjukkan siapa saja yang akan berhubungan dengan sistem, serta tujuan tertentu apa yang ingin mereka raih melalui interaksi tersebut.

4.1.2.1 Diaram Use Case

Diagram *Use Case* merupakan representasi visual yang digunakan untuk menggambarkan interaksi antara pengguna (*actor*) dengan sistem. Diagram ini menjelaskan siapa saja pihak yang terlibat dalam penggunaan sistem serta fungsi-fungsi apa saja yang dapat dilakukan oleh masing-masing pengguna.



Tabel 4. 5 Use Case Aplikasi Rute Angkot



Gambar 4. 2 Use Case Aplikasi Rute Angkot

Pada Gambar 4.2 Use Case Aplikasi Rute Angkot di atas Berikut adalah penjelasannya. Terdapat 1 Aktor yaitu Pengguna (Masyarakat yang ingin mencari informasi angkot berupa pengguna angkot atau bukan) dan terdapat 3 use case yaitu Melakukan pencarian angkot (mencari informasi rute angkot dengan menginput titik awal dan titik tujuan), Melihat detail informasi (Melihat detail hasil dari pencarian rute angkot berupa rute yang di lewati, kode, warna, tarif, gambar dan yang lainnya berdasarkan titik awal dan titik tujuan yang di input.

4.1.2.2 Definisi Aktor

Aktor itu adalah siapa pun yang berinteraksi dengan sistem perangkat lunak kita dari luar. Lebih mudahnya, Aktor adalah "si pengguna" atau "si pemakai" dari sebuah fungsi atau fitur di dalam sistem. Mereka adalah pihak yang punya tujuan saat menggunakan sistem tersebut.

Tabel 4. 6 Definisi Aktor

Aktor	Keterangan
Pengguna	Masyarakat yang ingin mencari informasi rute angkot berupa pengguna angkot atau yang tidak menggunakan angkot

4.2 Perancangan

Perancangan merupakan fase dalam proses pengembangan perangkat lunak yang berfokus pada cara pembangunan sistem. Ini adalah langkah di mana kebutuhan yang telah diidentifikasi pada tahap analisis (yakni, "apa yang harus dilaksanakan oleh sistem") diterjemahkan menjadi rencana rinci mengenai bagaimana kebutuhan tersebut akan diimplementasikan secara teknis.

4.2.1 Diagram Skenario

Skenario Use Case adalah langkah demi langkah yang merinci bagaimana seorang aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu yang digambarkan dalam sebuah Use.

UC – 01 Melakukan Pencarian

Tabel 4. 7 Diagram Skenario Melakukan Pencarian

Identifikasi	
Kode	UC - 01
Nama	Melakukan Pencarian Rute
Tujuan	Agar pengguna dapat melakukan pencarian rute angkot berdasarkan titik awal dan titik tujuan
Deskripsi	Use Case dapat melakukan pencarian rute angkot
Skenario Utama	
Kondisi Awal : Berada di halaman home	
Aksi Aktor	Reaksi Sistem
1. Klik Button cari di halaman home	2. Masuk kehalaman Pencarian Rute
3. Masukan input lokasi di form pada titik awal dan titik tujuan	
4. Klik button Cari Rute	
	5. Menampilkan card hasil berisi informasi singkat dan button lihat detail
Akhir : Berada di Halaman Pencarian dengan menampilkan card hasil sesuai titik awal dan titik tujuan	

UC – 02 Melihat Detail Informasi

Tabel 4. 8 Diagram Skenario Melihat Detail Informasi

Identifikasi	
Kode	UC - 02
Nama	Melihat Detail Informasi
Tujuan	Agar pengguna dapat melihat detail informasi dari angkot yang di cari
Deskripsi	Use Case dapat melihat detail informasi angkot
Skenario Utama	
Kondisi Awal : Berada di Halaman Pencarian dengan menampilkan card hasil sesuai titik awal dan titik tujuan	
Aksi Aktor	Reaksi Sistem
1. Klik Button lihat detail	
	2. Masuk kehalaman Detail
	3. Menampilkan Detail Informasi angkot berupa Map,Rute yang dilewati, Nama angkot, kode, warna, tarif dan lainnya
Akhir : Berada di Halaman detail yang menampilkan informasi	

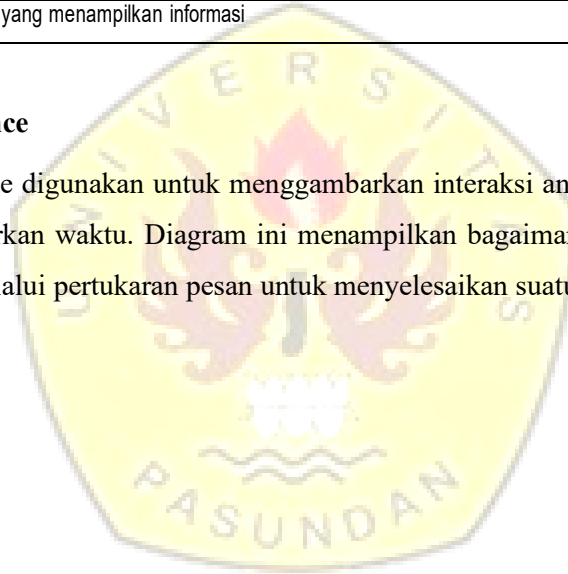
UC – 03 Melihat Daftar Trayek

Tabel 4. 9 Diagram Skenario Melihat Daftar Trayek

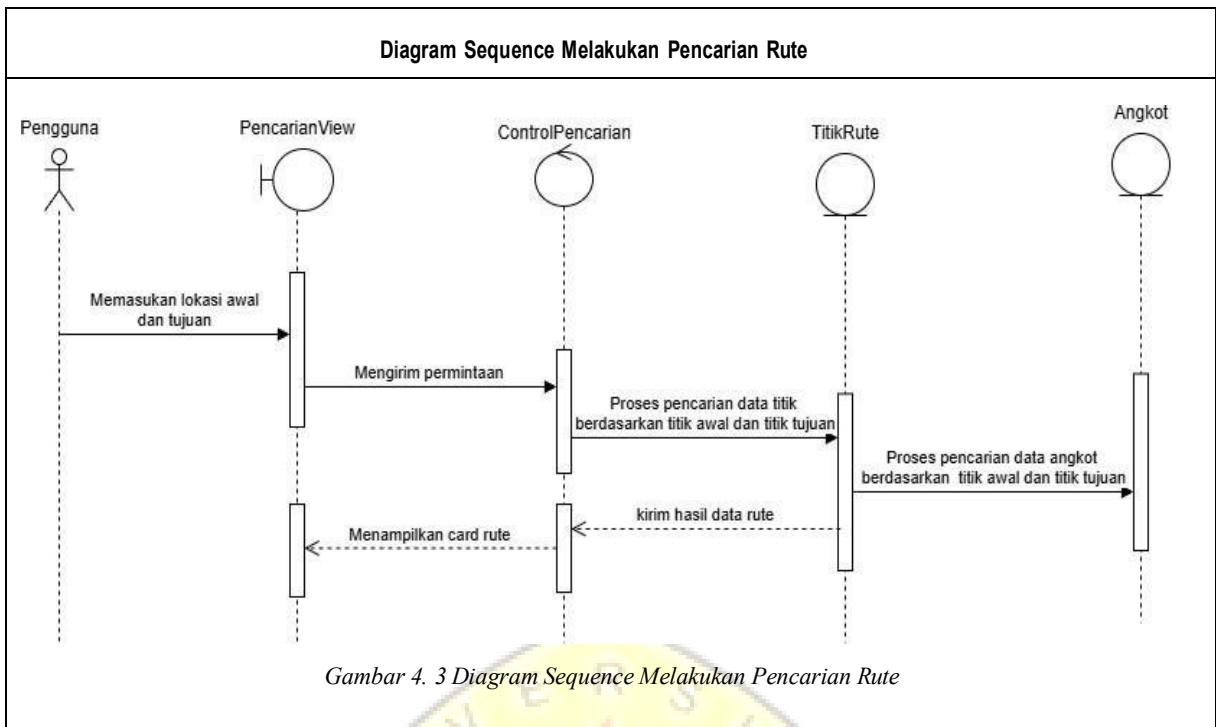
Identifikasi	
Kode	UC - 03
Nama	Melihat Daftar Trayek
Tujuan	Agar pengguna dapat melihat daftar taryek di kota bandung
Deskripsi	Use Case dapat melihat daftar taryek angkot
Skenario Utama	
Kondisi Awal : Berada di halaman home	
Aksi Aktor	Reaksi Sistem
1. Klik navbar trayek	
	2. Masuk kehalaman daftar trayek
3. Cari angkot dengan menginput nama angkot menggunakan Filter	
	4. Menampilkan Card trayek angkot yang di cari
Akhir : Berada di Halaman detail yang menampilkan informasi	

4.2.2 Diagram Sequence

Diagram sequence digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem secara berurutan berdasarkan waktu. Diagram ini menampilkan bagaimana objek-objek dalam sistem saling berkomunikasi melalui pertukaran pesan untuk menyelesaikan suatu proses atau fungsi tertentu

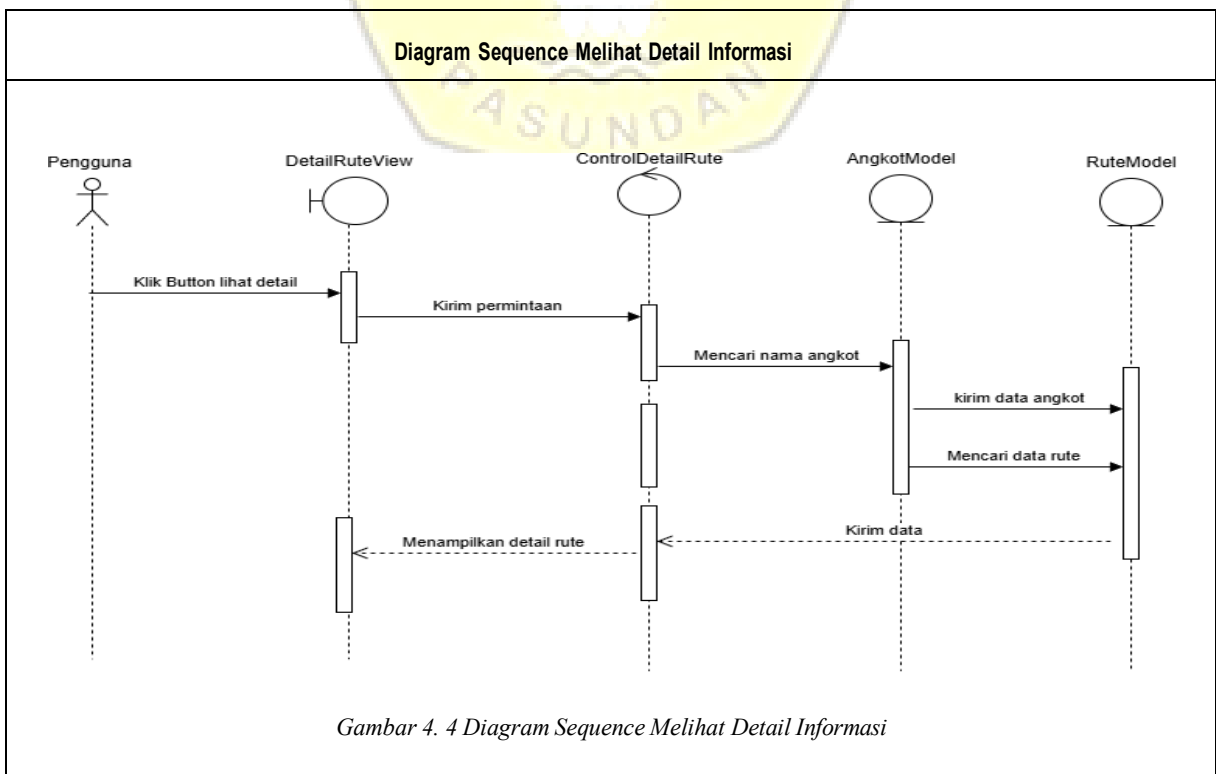


Tabel 4. 10 Diagram Sequence Melakukan Pencarian Rute



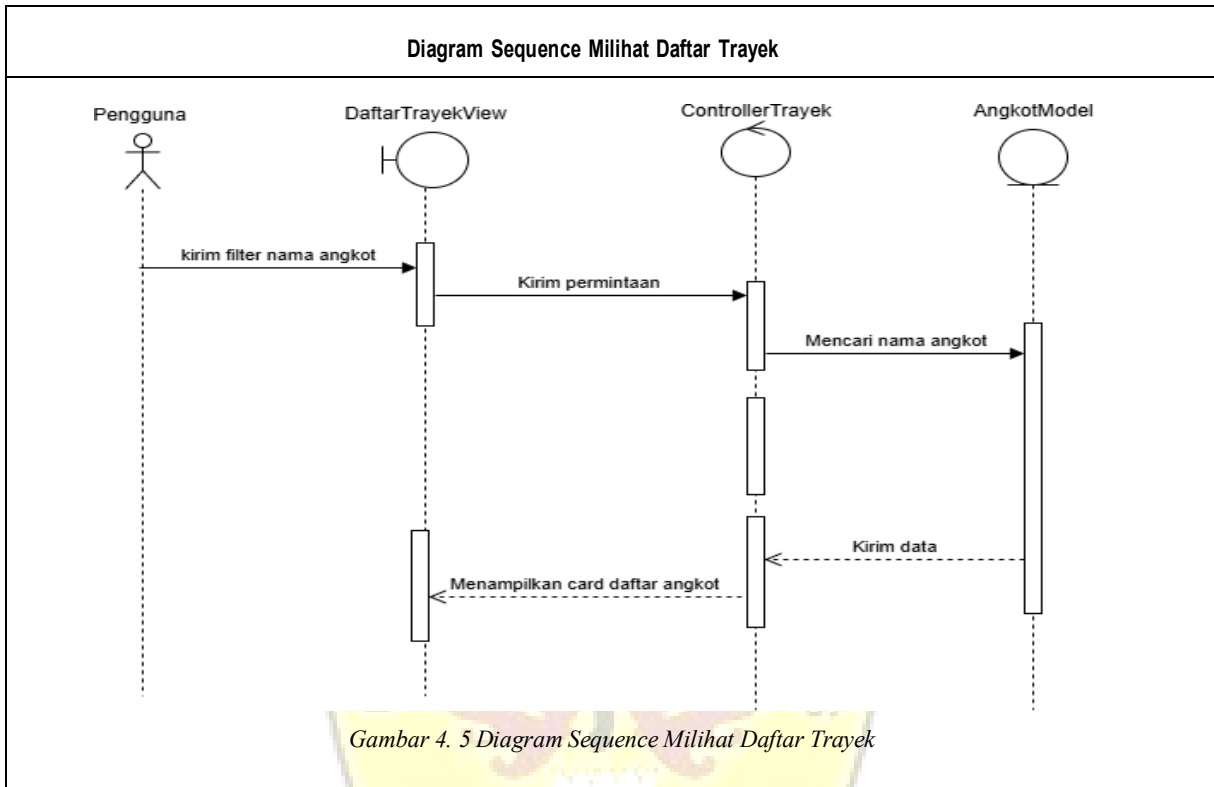
Pada Gambar 4.3 Diagram Sequence Melakukan Pencarian Rute, Berikut adalah penejelasanannya. Pada Gambar di atas Terdapat 5 komponen yaitu aktornya adalah Pengguna, View adalah PencarianView, Controller adalah ControllerPencarian dan untuk Model terdapat 2 yaitu TitikRute dan Angkot.

Tabel 4. 11 Diagram Sequence Melihat Detail Informasi



Pada Gambar 4.4 Diagram Sequence Melihat Detail Informasi diatas, Berikut adalah penjelasannya. Pada gambar di atas terdapat 5 komponen yaitu Aktor adalah pengguna, View adalah DetailView, Controller adalah ControllerDetailRute dan untuk Model terdapat 2 yaitu Angkot dan Rute.

Tabel 4. 12 Diagram Sequence Melihat Daftar Trayek



Pada Gambar 4.4 Diagram Sequence Melihat Daftar Trayek, Berikut adalah penjelasannya. Pada gambar di atas terdapat 4 komponen yaitu Aktor adalah pengguna, View adalah DaftarTrayekView, Controller adalah ControllerTrayek dan untuk Model adalah Angkot

4.2.3 Pemodelan Kelas Analisis

Diagram kelas analisis digunakan untuk menggambarkan struktur statis dari sistem yang sedang dianalisis. Diagram ini memodelkan kelas-kelas yang ada dalam sistem beserta atribut, metode, dan hubungan antar kelas tersebut, tanpa memperlihatkan detail implementasi.

4.2.3.1 Mengidentifikasi Kelas – Kelas Analisis

Proses dalam tahap analisis sistem yang bertujuan untuk menemukan dan menentukan objek-objek utama yang terdapat pada sistem berdasarkan kebutuhan fungsional. Kelas-kelas ini merepresentasikan entitas penting yang terlibat dalam sistem, baik yang berasal dari data, proses, maupun aktor yang berinteraksi dengan sistem.

Tabel 4. 13 Mengidentifikasi Kelas – Kelas Analisis

No	Nama Kelas	Jenis	Keterangan
View			
1.	PencarianView	View	Kelas ini digunakan untuk menampilkan halaman pencarian dengan detail form pencarian titik awal, titik tujuan dan button CariRute
2.	DetailView	View	Kelas ini digunakan untuk menampilkan halaman detail informasi dengan detail sebagai berikut map, nama angkot, tarif, kode, warna, dan rute yang di lewati
3.	TrayekView	View	Kelas ini digunakan untuk menampilkan halaman daftar trayek angkot dan filter
4.	HomeView	View	Kelas ini digunakan untuk menampilkan halaman Home page atau
Controller			
1.	PencarianController	Controller	Kelas ini digunakan untuk menjalankan perintah pencarian rute yang di hubungkan dengan Graph dan Bfs
2.	DetailController	Controller	Kelas ini digunakan untuk menjalankan Detail informasi yang di tampilkan sesuai dengan titik awal dan titik tujuan
3.	TrayekController	Controller	Kelas ini digunakan untuk menjalankan daftar trayek sesuai dengan database
4.	HomeController	Controller	Kelas ini digunakan untuk menjalankan halaman Home
5.	AngkotController	Controller	Kelas ini digunakan untuk Menangani proses pengambilan data nama titik rute dari database yang digunakn untuk menampilkan pilihan di input pencarian
Model			
1.	AngkotModel	Model	Kelas ini digunakan untuk menghubungkan data angkot dengan model lain
2.	RuteModel	Model	Kelas ini digunakan untuk menghubungkan data rute dengan model lain
3.	TitikRuteModel	Model	Kelas ini digunakan untuk menghubungkan data TitikRute dengan model lain

4.2.3.2 Menentukan Atribut

Proses dalam tahap analisis dan perancangan sistem yang bertujuan untuk menetapkan karakteristik atau data yang dimiliki oleh setiap kelas yang telah diidentifikasi. Atribut merepresentasikan informasi penting yang menggambarkan keadaan atau sifat suatu objek dalam sistem.

Tabel 4. 14 Menentukan Atribut

No	Nama Kelas	Jenis Kelas	Atribut	Keterangan
1.	AngkotModel	Model	Id_ angkot	Atribut ini digunakan untuk id angkot
			Nama_angkot	Atribut ini digunakan untuk nama angkot “ ledeng – margahayu “
			Kode	Atribut ini digunakan untuk kode angkot
			Warna	Atribut ini digunakan untuk warna angkot
			Tarif	Atribut ini digunakan untuk tarif angkot
			Gambar	Atribut ini digunakan untuk gambar
2.	RuteModel	Model	Id_rute	Atribut ini digunakan untuk id rute angkot
			Nama_rute	Atribut ini digunakan untuk nama rute (nama nama jalan)
			jarak	Atribut ini digunakan untuk jarak rute
3.	TitikRuteModel	Model	Id_titik_rute	Atribut ini digunakan untuk id titik rute

			Urutan	Atribut ini digunakan untuk mendeskripsikan urutan nama titik (1 – jalan)
			Nama_titik	Atribut ini digunakan untuk nama titik (nama jalan)
			Longitude	Atribut ini digunakan untuk longitude
			Latitude	Atribut ini digunakan untuk latitude

4.2.3.3 Menentukan Operasi

Proses dalam tahap analisis dan perancangan sistem yang bertujuan untuk menetapkan fungsi atau tindakan yang dapat dilakukan oleh setiap kelas yang telah diidentifikasi. Operasi menggambarkan perilaku suatu objek dalam sistem, yaitu bagaimana objek tersebut memproses data, merespons permintaan, serta berinteraksi dengan objek lain.

Tabel 4. 15 Menentukan Operasi

No	Nama Kelas	Jenis Kelas	Nama Operasi	Kandidat Perilaku
Controller				
1.	PencarianController	Controller	Validate()	Melakukan Validasi pada setiap input
			graphBuilder()	Mengubah data titik dari database menjadi struktur data graf
			BfsService()	Mencari jalur terpendek atau jalur yang menghubungkan titik awal ke titik tujuan
			Input()	Menerima data dari pengguna melalui
			IF()	Mengambil keputusan untuk mengembalikan
			Foreach()	Melakukan perulangan pada setiap data
			For()	Melakukan perulangan untuk memproses pemotongan rute dan tarif
			Where (Query)	Mencari data koordinat berdasarkan nama titik
			WhereIn()	Mengambil data rute berdasarkan ID
			Slice()	Memotong daftar titik rute agar yang ditampilkan hanya bagian yang dilewati
			haversine()	Menghitung jarak koordinat
2.	DetailController	Controller	Input()	Menerima data dari pengguna melalui
			findOrFail()	mencari data rute berdasarkan ID

			BfsService()	Mencari jalur terpendek atau jalur yang menghubungkan titik awal ke titik tujuan
			haversine()	Menghitung jarak berdasarkan koordinat
			IF()	Mengambil keputusan untuk mengembalikan
			Foreach()	Melakukan perulangan pada setiap data
			For()	Melakukan perulangan untuk memproses pemotongan rute dan tarif
3.	TrayekController	Controller	Index()	Mengatur seluruh jalannya logika pada halaman trayek dengan data tertentu
			Select()	Mengambil data nama rute dari database untuk ditampilkan pada filter
			Get()	Mengeksekusi query dan mengambil data trayek dari database
			Where (Query)	Operasi penyaringan data di database berdasarkan filter
4.	AngkotController	Controller	Input()	Menangkap data yang diketik pengguna melalui kunci pencarian
			IF()	Menghentikan proses jika input kosong
			Where()	Mencari nama titik rute yang mengandung kata kunci tertentu
			Distinct()	Memastikan nama lokasi yang muncul tidak duplikat
			Pluck()	mengambil nilai kolom tertentu dan mengubahnya menjadi format array sederhana.
5.	HomeController	Controller	Index()	Menampilkan halaman Homepage aplikasi.
Model				
1.	AngkotModel	Model	firstOrCreate()	Mencari data terlebih dahulu di database, apabila tidak ada akan dibuatkan lalu disimpan otomatis
			Create()	Menambahkan data baru ke dalam basis data sesuai dengan atribut yang diberikan.

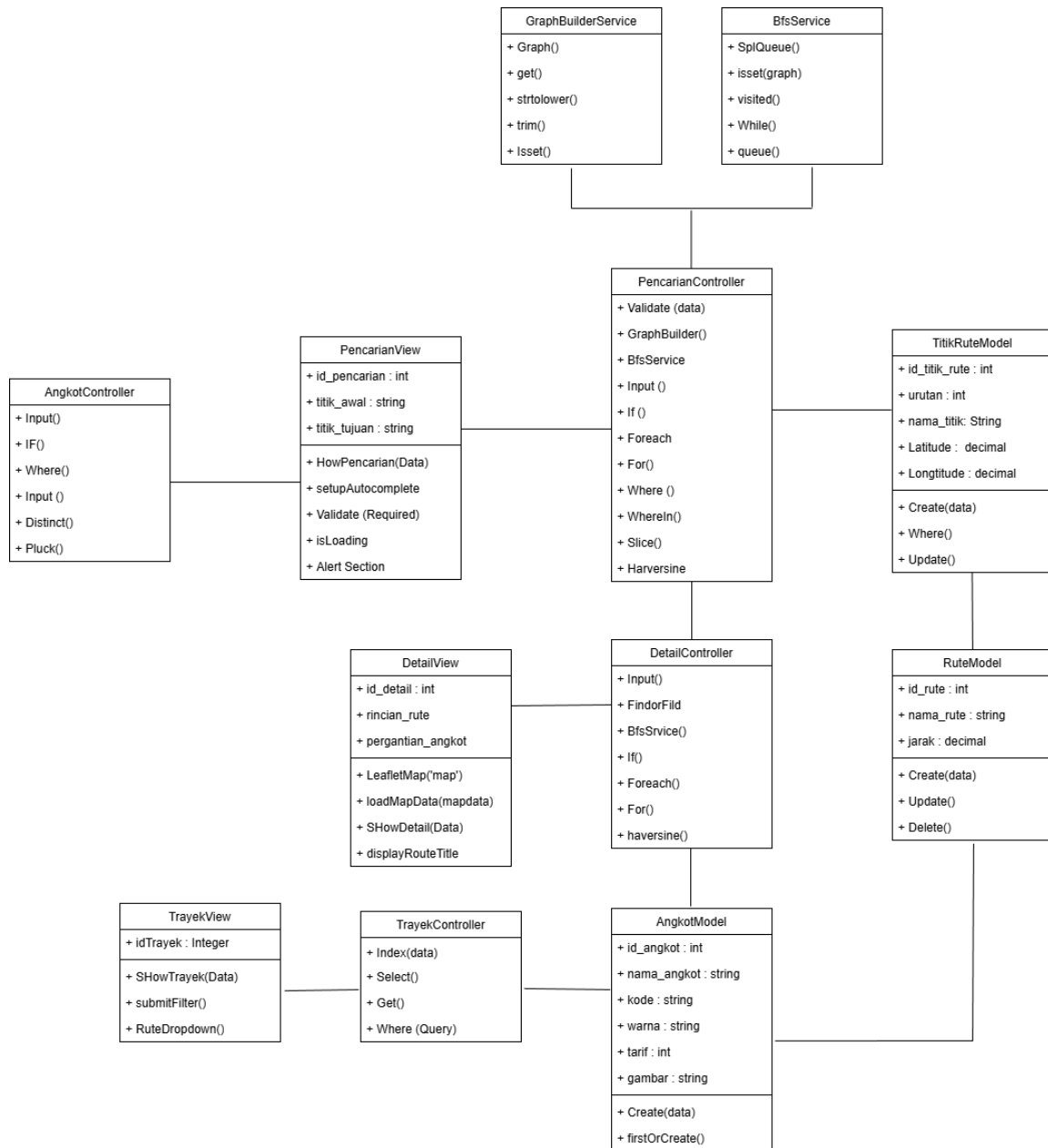
2.	RuteModel	Model	Create()	Menambahkan data baru ke dalam basis data sesuai dengan atribut yang diberikan.
			Update()	Memperbarui atau mengubah data yang sudah ada di dalam basis
			Delete()	Menghapus data dari basis data
3.	TitikRuteModel	Model	Create()	Menambahkan data baru ke dalam basis data sesuai dengan atribut yang diberikan.
			Where()	Memfilter data berdasarkan kondisi tertentu
			Update()	Memperbarui atau mengubah data yang sudah ada di dalam basis
View				
1	HomeView	View	ShowHome(data)	Menampilkan seluruh informasi pada halaman Home, meliputi navigasi utama, konten hero, daftar trayek angkot, deskripsi aplikasi, serta footer
2.	PencarianView	View	HowPencarian(Data)	Menampilkan seluruh informasi pada halaman Pencarian, meliputi navigasi utama, konten hero, daftar trayek angkot, deskripsi aplikasi, serta footer
			setupAutocomplete (asal dan tujuan)	Menampilkan fitur daftar saran tempat atau titik lokasi saat pengguna mengetik
			Validasi Input (required)	Mencegah pengiriman formulir jika kolom titik awal atau titik tujuan masih kosong
			isLoading	Saat tombol cari ditekan, teks tombol berubah menjadi "Sedang Memproses...".
			Alert Section (@if(session('error')))	Menampilkan pesan kesalahan (seperti rute tidak ditemukan atau validasi gagal)
			fetchLokasiSuggestion('/get-lokasi?term)	Mengambil data saran lokasi dari database berdasarkan input pengguna.
3.	DetailView	View	initializeLeafletMap('map')	Menginisialisasi peta menggunakan library Leaflet dengan titik awal rute.

			loadMapData(mapdata)	Mengambil data koordinat rute dari database untuk ditampilkan pada peta.
			HowDetail(Data)	Menampilkan seluruh informasi pada halamanDetail, meliputi navigasi utama, konten hero, daftar trayek angkot, deskripsi aplikasi, serta footer
			displayRouteTitle({{ \$route->nama_rute }})	Menampilkan data rute angkot yang sedang dilihat pada halaman detail.
4.	TrayekView	View	HowTrayek(Data)	Menampilkan seluruh informasi pada halaman Trayek, meliputi navigasi utama, konten hero, daftar trayek angkot, deskripsi aplikasi, serta footer
			RuteDropdown(@foreach(\$trayeks as \$option))	Mengisi pilihan dropdown rute trayek dari data trayek yang tersedia.
			submitFilterOnChange(this.form.submit)	Mengirimkan form filter secara otomatis saat pengguna memilih rute.
Service				
1.	BfsService	Service	SplQueue()	Membuat struktur data antrian (queue) sebagai penunjang algoritma Breadth First Search.
			isset(graph)	Memvalidasi keberadaan node awal di dalam graph sebelum proses BFS dijalankan.
			queue()	Untuk menyimpan node terdepan dari antrian untuk diproses terlebih dahulu
			visited()	Menyimpan daftar node yang telah dikunjungi untuk mencegah pengulangan pencarian.
			While()	Melakukan proses pencarian selama masih terdapat node di dalam antrian.
			foreach (graph)	Menelusuri seluruh node tetangga yang terhubung dengan node saat ini.
			isset(visited)	Mengecek apakah node tetangga belum pernah dikunjungi sebelumnya.

			return \$path	Mengembalikan jalur rute ketika node tujuan berhasil ditemukan.
			return null	Mengembalikan nilai kosong apabila tidak ditemukan jalur dari titik awal ke tujuan.
2.	GraphBuilderService	Service	Graph()	Menginisialisasi array graph sebagai struktur data untuk menyimpan node dan koneksi antar node.
			get()	Mengambil seluruh data rute beserta titik-titiknya dari basis data.
			strtolower()	Mengubah seluruh huruf dalam sebuah string menjadi huruf kecil
			trim()	Menghapus spasi atau karakter kosong di awal dan di akhir sebuah string
			Count(titik)	Menghitung jumlah titik pada satu rute.
			Isset()	Mengecek apakah node asal sudah terdaftar di dalam graph.
			Return (graph)	Mengembalikan struktur graph yang telah terbentuk untuk digunakan dalam proses BFS.

4.2.3.4 Diagram Kelas

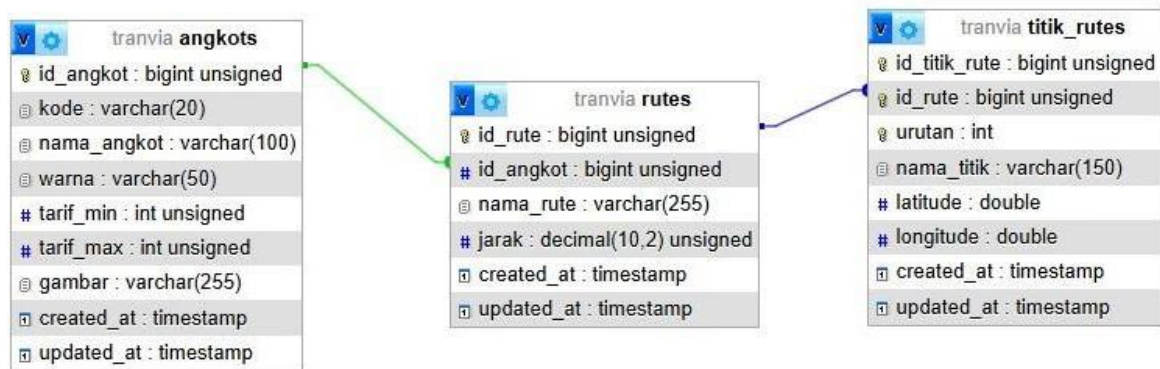
Diagram kelas analisis adalah diagram yang digunakan pada tahap analisis sistem untuk menggambarkan struktur logis dari sistem, terutama untuk mengidentifikasi objek-objek utama (kelas-kelas) dan hubungan antar objek tersebut berdasarkan kebutuhan pengguna.



Gambar 4. 6 Diagram Kelas

4.2.3.5 Skema Basis Data

Perancangan skema basis data adalah proses menyusun struktur data yang akan digunakan oleh sistem informasi. Tujuan dari perancangan ini adalah untuk memastikan bahwa semua data yang dibutuhkan oleh sistem dapat disimpan, diakses, dan dikelola dengan baik.



Gambar 4. 7 Skema Basis Data

4.2.3.6 Contoh Implementasi Algoritma Breadth First Search

Berikut adalah salah satu contoh pencarian rute angkot “ Terminal Ledeng – Jl. Cihampelas”

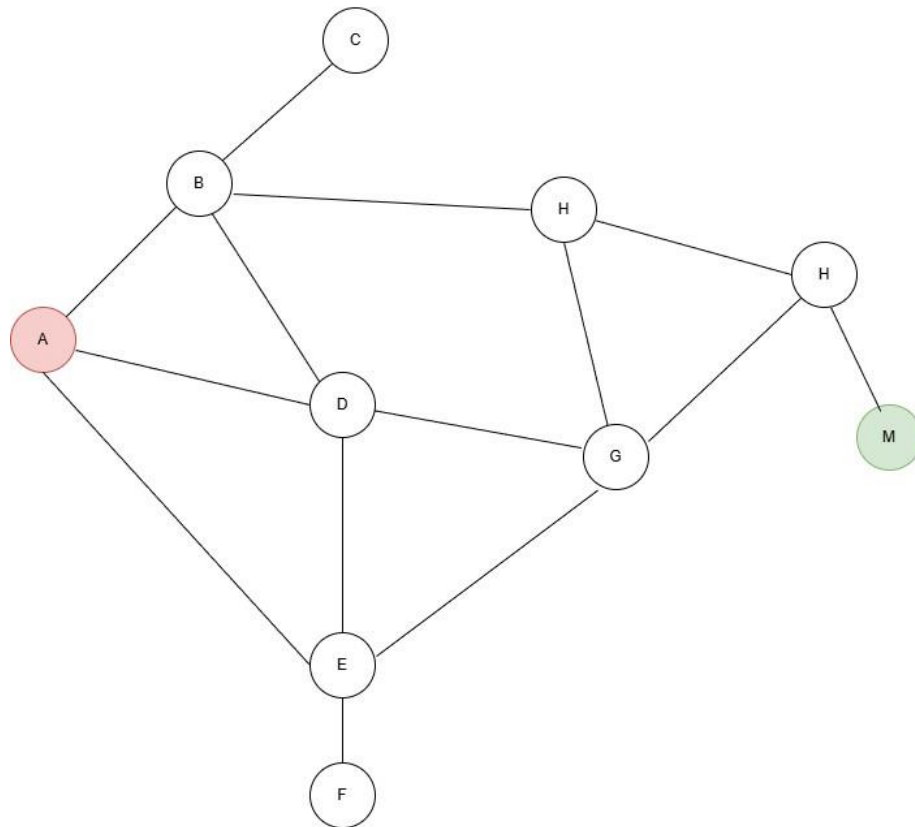
Keterangan :

- A -> Terminal Ledeng
- B -> Jl. Setiabudi
- C -> Jl. Karang Setra
- D -> Jl. Karang Sari
- E -> Jl. Sukajadi
- F -> Jl. Dago
- G -> Jl. Jurang
- H -> Jl Sederhana
- I -> Jl. Eyckman
- J -> Jl. Cihampelas



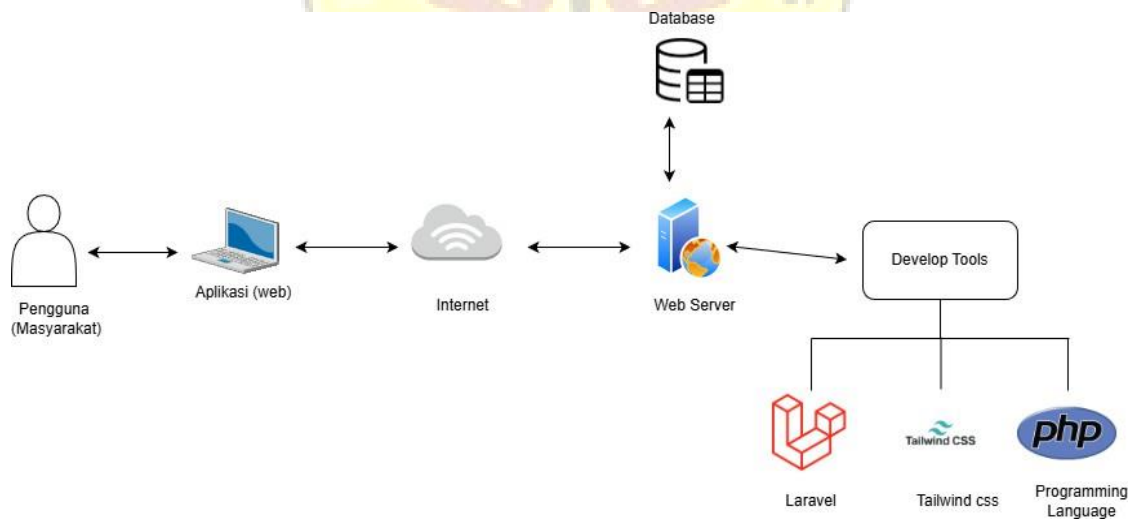
Penjelasan

Proses pencarian rute dari titik awal Terminal Ledeng (A) menuju Terminal Margahayu (M) dilakukan dengan memasukkan simpul A ke dalam antrian (*queue*), lalu secara berulang mengambil simpul deidepannya atau tetangga untuk divalidasi apakah simpul tersebut merupakan titik tujuan yang dicari. Jika simpul yang diperiksa bukan merupakan solusi, algoritma akan memasukkan seluruh simpul tetangga yang bersentuhan langsung ke dalam antrian, menandai simpul yang telah diperiksa sebagai sudah dikunjungi (*visited*), dan melanjutkan penelusuran secara melebar lapis demi lapis sesuai urutan masuknya. Langkah-langkah ini diulangi terus-menerus hingga simpul target M berhasil ditemukan dan dikembalikan sebagai rute akhir, atau pencarian dihentikan dengan hasil solusi tidak ditemukan apabila seluruh antrian telah kosong dan semua simpul telah diperiksa.



Gambar 4. 8 Contoh Graph BFS

4.2.3.7 Perancangan Arsitektur



Gambar 4. 9 Perancangan Arsitektur

Perancangan arsitektur sistem Pada Gambar di atas ini menggunakan model berbasis web yang memberi kesempatan kepada pengguna untuk mengakses sistem melalui internet. Model ini diciptakan agar bisa diakses dengan mudah, mudah untuk dikembangkan lebih lanjut, dan memiliki pembagian tugas yang jelas antara elemen-elemen yang ada. Pengguna dapat mengunjungi aplikasi lewat antarmuka web yang berfungsi sebagai medium untuk berinteraksi, termasuk memasukkan lokasi awal dan akhir

dalam pencarian rute angkot. Permintaan yang diajukan oleh pengguna dikirim melalui internet menuju server web yang bertanggung jawab untuk memproses logika pada aplikasi. Di sisi server, aplikasi dibangun menggunakan Laravel sebagai framework dengan PHP berfungsi sebagai backend. Laravel digunakan untuk mengatur alur sistem, logika bisnis, serta menerapkan algoritma Breadth First Search (BFS) dalam menemukan rute. Desain antarmuka aplikasi dibuat menggunakan Tailwind CSS untuk menghasilkan tampilan yang responsif dan seragam.

4.2.3.8 Perancangan Antarmuka

Perancangan antar muka (User Interface Design atau UI Design) adalah proses merancang tampilan dan interaksi visual antara pengguna (user) dan sistem atau aplikasi. Tujuan utamanya adalah untuk menciptakan antarmuka yang mudah digunakan, menarik secara visual, dan efektif dalam digunakan oleh pengguna.

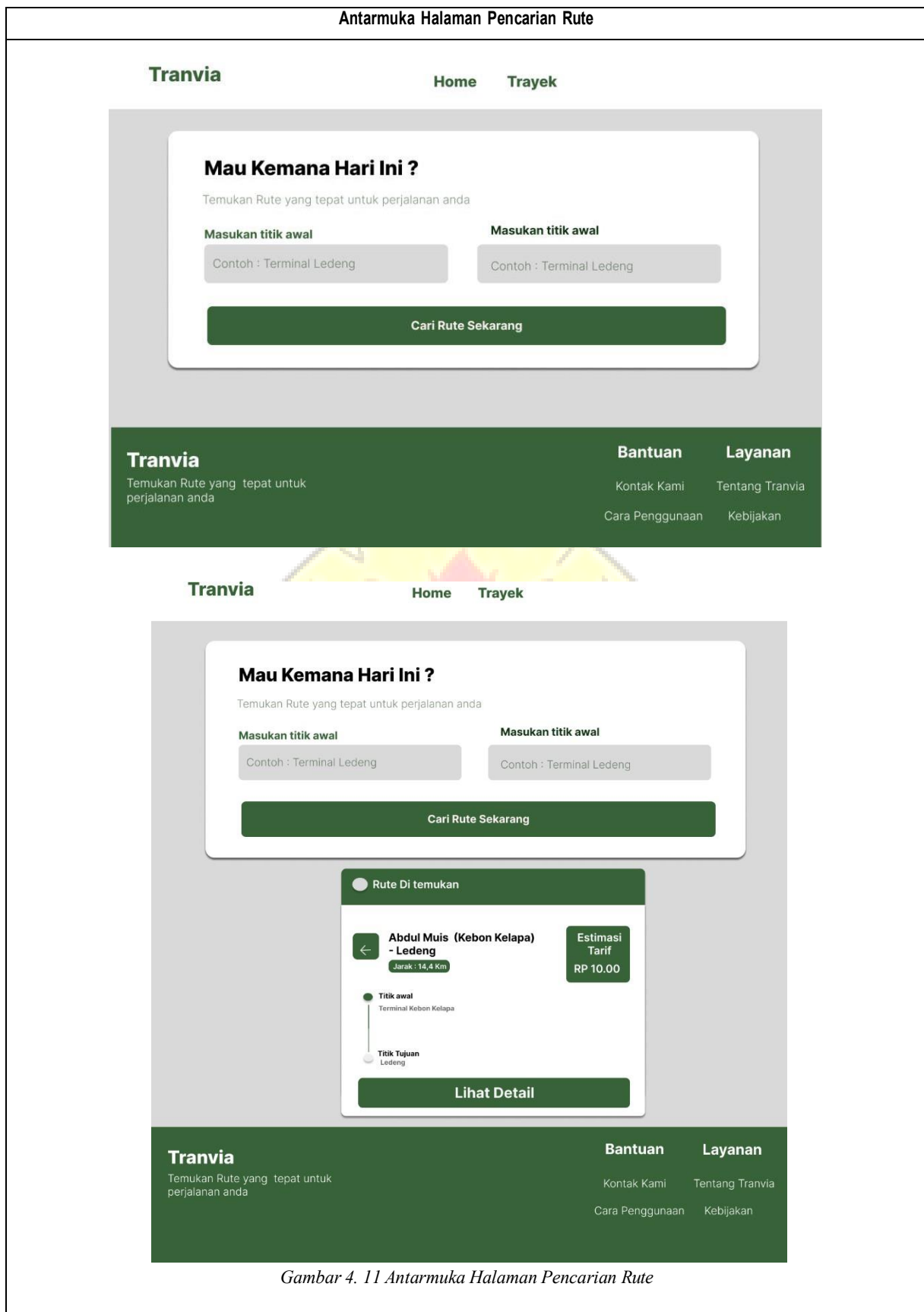
Tabel 4. 16 Antarmuka Halaman Home



Gambar 4. 10 Antarmuka Halaman Home

Deskripsi Pada Antarmuka UI untuk tampilan Home, detailnya sebagai berikut. Yang Pertama ada Konten 1 yaitu gambar angkot dengan teks singkat dan button cari untuk menuju halaman pencarian. Konten kedua adalah Daftar card trayek angkot.konten terakhir adalah penjelasan singkat dari pengertian tranvia.

Tabel 4. 17 Antarmuka Halaman Pencarian Rute



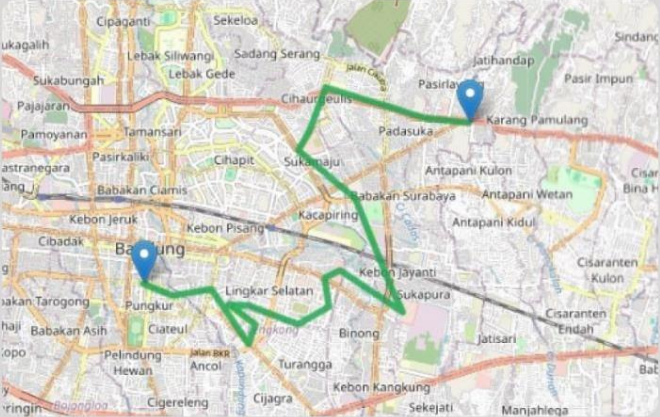
Deskripsi Pada AntarMuka UI untuk tampilan pencarian, detailnya sebagai berikut. Yang pertama ada form card saat pertama kali Anda masuk ke halaman pencarian menampilkan form dengan detail titik awal, titik tujuan dan button carirute, lalu selanjutnya menampilkan card hasil sesuai dengan input lokasi titik awal dan titik tujuan dengan detail kode, nama angkot, tarif, jarak dan button lihat detail

Tabel 4. 18 AntarMuka Halaman Detail Informasi

AntarMuka Halaman Detail Informasi

Tranvia
Home Trayek

Kembali



Rute Yang di Lewati

1 Terminal Kebon kelapa

2 Jl. Dewi sartika

3 Jl. Kautaman Istri

4 Jl. Pungkur

5 Terminal Ledeng


Terminal Kebon Kelapa - Ledeng

Wrana : Hijaun - Kuning

Jarak Tempuh
14,4 KM

Estimasi Tarif
RP. 10.000

Informasi Angkot



Kode 01A

Tranvia

Temukan Rute yang tepat untuk perjalanan anda

Bantuan Layanan

Kontak Kami

Tentang Tranvia

Cara Penggunaan

Kebijakan

Gambar 4. 12 AntarMuka Halaman Detail Informasi

Deskripsi Pada AntarMuka UI untuk tampilan Detail sebagai Brikut dengan isi Navbar, footer , Map untuk menampilkan titik rute, ada informasi angkot dan ada rute yang di lewati


Tabel 4. 19 Antarmuka Halaman Daftar Trayek

Antarmuka Halaman Daftar Trayek

Tranvia

[Home](#)
[Trayek](#)
[Profile](#)

Filter



Ledeng - Margahayu 17 Km

Terminal Ledeng - Terminal Margahayu

Rp. 15000

Tranvia

Temukan Rute yang tepat untuk perjalanan anda

Bantuan

Kontak Kami

Cara Penggunaan

Layanan

Tentang Tranvia

Kebijakan

Gambar 4. 13 Antarmuka Halaman Daftar Trayek

Deskripsi Pada AntarMuka UI untuk tampilan Daftar Trayek sebagai Brikut dengan isi Navbar, footer card yang di ambil dari database atau disesuaikan dengan database Angkot dan ada search filter untuk mencari nama angkot sesuai yang kit mau

BAB 5

IMPLEMENTASI DAN PENGUJIAN

5.1 Kontruksi Perangkat

Konstruksi perangkat lunak adalah tahapan yang mencakup identifikasi kebutuhan perangkat lunak dan perangkat keras, perancangan struktur aplikasi, pengkodean, serta pengujian. Tahapan konstruksi ini dilakukan berdasarkan hasil analisis dan rancangan yang telah dibuat sebelumnya.

5.1.1 Kontruksi Kakas Perangkat Lunak

Dalam pembangunan perangkat lunak diperlukan beberapa kakas dan kebutuhan perangkat lunak. Berikut adalah kakas dan kebutuhan perangkat lunak yang digunakan dalam penelitian ini :

Tabel 5. 1 Kakas Perangkat Lunak

No	Nama Perangkat Lunak	Sumber	Deskripsi
1.	Visual Studio Code	https://code.visualstudio.com/	Perangkat lunak ini merupakan editor kode yang digunakan untuk menuliskan kode program aplikasi. Digunakan bersama Laragon, Laravel, Tailwind CSS, dan Leaflet.
2.	Laragon	https://laragon.org/	Perangkat lunak ini digunakan sebagai lingkungan pengembangan lokal untuk menjalankan server web, PHP, dan database MySQL. Memudahkan pembangunan aplikasi menggunakan Laravel.
3.	Mysql	https://www.mysql.com/	Perangkat lunak ini digunakan sebagai sistem manajemen basis data untuk menyimpan data rute, titikrute, dan angkot.
4.	Laravel	https://laravel.com/	Perangkat lunak ini merupakan framework PHP yang digunakan untuk membangun aplikasi web. Memudahkan pengelolaan rute, model, view, dan kontroler dalam proyek.
5.	Tailwind css	https://tailwindcss.com/	Perangkat lunak ini digunakan untuk membuat tampilan antarmuka aplikasi web dengan cepat dan responsif. Digunakan bersama Laravel dan editor kode.
6.	Leaflet.js	https://leafletjs.com/	Perangkat lunak ini merupakan library JavaScript untuk membuat peta interaktif pada aplikasi web, digunakan untuk menampilkan rute dan titik angkot.

5.1.2 Kontruksi Kakas Perangkat Keras

Dalam pembangunan aplikasi diperlukan beberapa perangkat keras pendukung. Berikut adalah perangkat keras yang digunakan dalam penelitian ini:

Tabel 5. 2 Kakas Perangkat Keras

No		Komponen	Rekomendasi Sistem
1.	Personal PC / Laptop	Processor (CPU)	Minimal: Intel Core i3 Gen 10 / AMD Ryzen 3. Rekomendasi: Intel Core i5/i7 Gen 10+ / Ryzen 5/7.
		RAM	Minimal: 8 GB. Rekomendasi: 16 GB untuk multitasking (VS Code, browser, DB).
		Storage	Minimal: SSD 256 GB. Rekomendasi: SSD 512 GB NVMe untuk kecepatan optimal.
		Sistem Operasi	Windows 10/11 64-bit / Linux (Ubuntu 20.04+) / macOS 11+.

5.2 Pengkodean

Tahap ini meliputi penerjemahan rancangan sistem ke kode program. Aplikasi menggunakan PHP dengan Laravel untuk back-end, Tailwind CSS dan Flowbite untuk front-end, serta MySQL sebagai sistem manajemen basis data.

5.2.1 Struktur Aplikasi

Pembangunan aplikasi ini menerapkan pendekatan arsitektur MVC (Model-View-Controller), yang membagi peran menjadi tiga bagian: Model untuk pengelolaan data, View untuk tampilan antarmuka pengguna, dan Controller sebagai penghubung serta pengatur alur logika antara keduanya

5.2.1.1 View

View dalam arsitektur MVC berfungsi menampilkan antarmuka pengguna serta menampilkan data dari controller dalam format yang mudah dipahami. View juga menerima input dari pengguna yang kemudian diteruskan ke controller untuk diproses.

Tabel 5. 3 View

No	Nama View	Deskripsi
1.	Pencarian.blade.php	Digunakan untuk menampilkan halaman Pencarian Rute
2.	Detail.blade.php	Digunakan untuk menampilkan halaman Detail Informasi angkot
3.	Trayek.blade.php	Digunakan untuk menampilkan halaman Daftar Trayek angkot
4.	Home.blade.php	Digunakan untuk menampilkan halaman Home page

5.2.1.2 Controller

Controller dalam arsitektur MVC berperan sebagai penghubung antara Model dan View. Controller menerima permintaan pengguna, memrosesnya sesuai logika aplikasi, mengambil data dari Model jika diperlukan, dan mengirim hasilnya ke View untuk ditampilkan.

Tabel 5. 4 Controlller

No	Nama Controller	Deskripsi
1.	PencarianController	Digunakan untuk Menerima perintah menampilkan halaman Pencarian, Menjalankan perintah pencarian rute dan menghubungkan dan menjalankan Graph dan Bfs
2.	DetailController	Digunakan untuk Menerima Perintah untuk mnjalankan tampilan dan Mengambil data lengkap dari database lalu mengirimkannya ke View untuk menampilkan informasi detail.
3.	TrayekController	Digunakan untuk mengambil data lengkap dari database lalu di mengirimkannya ke view trayek.blade untuk menampilkan daftar trayek sesuai dengan database
4.	HomeController	Digunakan untuk menerima perintah menampilkan halama Home page.
5.	AngkotController	Digunakan untuk mengambil dan menyediakan data titik awal dan titik tujuan dari basis data untuk kebutuhan AJAX pada proses pencarian rute

5.2.1.3 Model

Model dalam arsitektur MVC berfungsi untuk mengelola data dengan berkomunikasi langsung dengan database, serta memproses data yang diterima dari Controller, seperti menyimpan, mengambil, dan menghapus data. Laravel menyediakan fitur Eloquent yang memudahkan pengembang dalam pengelolaan database.

Tabel 5. 5 Model

No	Nama Model	Deskripsi
1.	AngkotModel	Digunakan untuk merepresentasikan tabel angkot di dalam database. berfungsi untuk menyimpan, memperbarui, dan mengambil data angkot yang ada di AngkotModel.
2.	RuteModel	Digunakan untuk merepresentasikan tabel rute di dalam database. Model ini berfungsi untuk menyimpan, memperbarui, dan mengambil data rute angkot,
3.	TitikRuteModel	Digunakan untuk merepresentasikan tabel titik_rute di dalam database. Model ini berfungsi untuk menyimpan, memperbarui, dan mengambil data di database. Model TitikRute menjadi komponen penting dalam proses pencarian rute menggunakan algoritma Breadth First Search (BFS), karena digunakan sebagai node yang merepresentasikan jalur perjalanan angkot dari titik awal menuju titik tujuan.

5.2.1.4 Services

Dalam arsitektur MVC, Service digunakan sebagai tempat penyimpanan logika algoritma Breadth First Search (BFS) yang berfungsi untuk mengolah data dan dihubungkan dengan Controller serta Model.

Tabel 5. 6 Services

No	Nama Model	Deskripsi
1.	BfsService	Digunakan untuk menyimpan dan menjalankan logika algoritma Breadth First Search (BFS) yang digunakan dalam proses pencarian rute angkot.
2.	GraphBuilderService	Digunakan untuk untuk membangun struktur graf dengan mengubah nama titik atau lokasi menjadi node dan edge yang siap digunakan dalam proses pencarian rute.

5.3 Implementasi Antarmuka

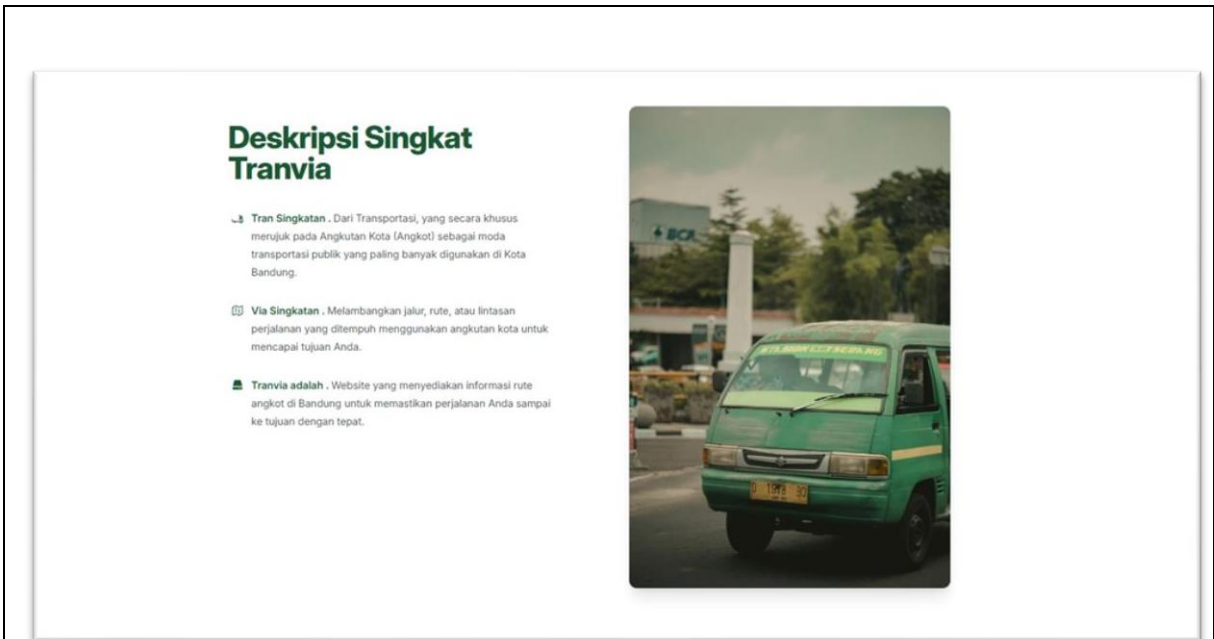
Implementasi antarmuka pengguna ini didasarkan pada rancangan yang dibuat pada tahap analisis dan perancangan. Berikut merupakan hasil implementasi antarmuka aplikasi web :

Tabel 5. 7 Implementasi Antarmuka Halaman Home

Implementasi Antarmuka Halaman Home

The screenshot shows a web browser window with the URL 127.0.0.1:8000. The page title is 'Tranvia' and the navigation menu includes 'Home' and 'Trayek'. The main content area has a large heading 'Selamat Datang Di Tranvia' and a sub-heading 'Informasi rute angkutan kota (Angkot) di Kota Bandung. Temukan jalur terbaik untuk perjalanan Anda.' A 'Cari Rute' button is visible below the text. Below this is a section titled 'Daftar Trayek Angkot' with the sub-heading 'Temukan informasi rute dan tarif angkutan kota favorit Anda.' This section contains three route cards, each with a bus image, route name, description, estimated fare, and trayek code.

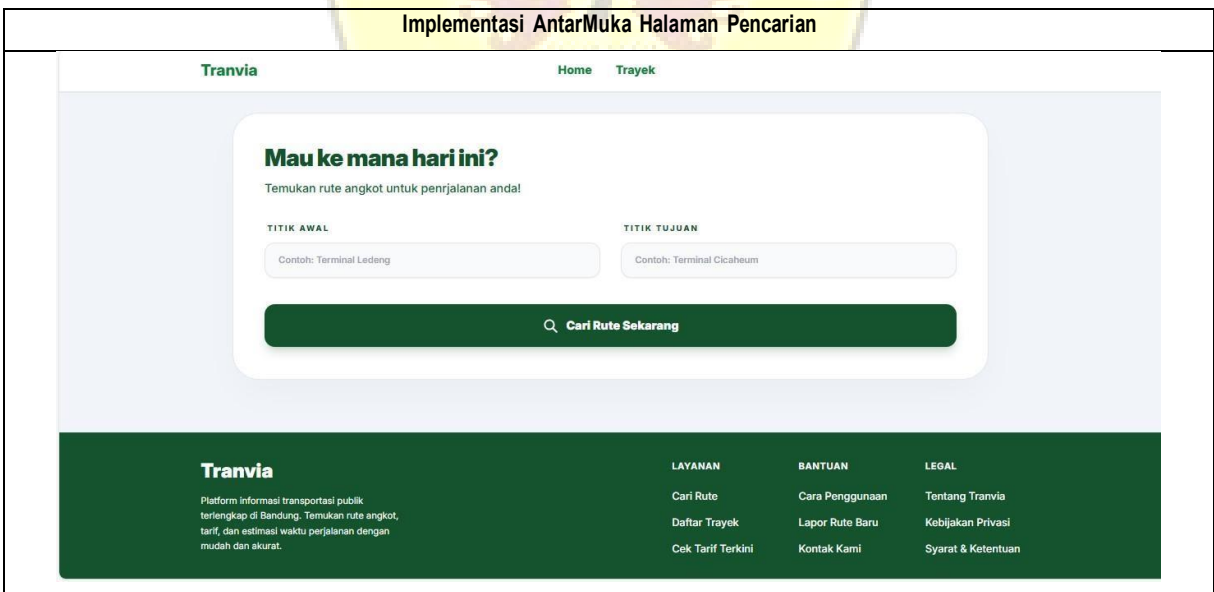
Route	Description	Tarif Estimasi	Kode Trayek
LEDENG - MARGAHAYU	Rute dari Terminal Ledeng ke Margahayu Raya.	Rp 15.000	01A
CICAHEUM - CIROYOM	Menghubungkan Cicaheum dan Ciroyom melewati Jalan Ahmad Yani.	Rp 10.000	05
ST. HALL - SARIJADI	Rute cepat menuju area perumahan utara Bandung.	Rp 8.000	13

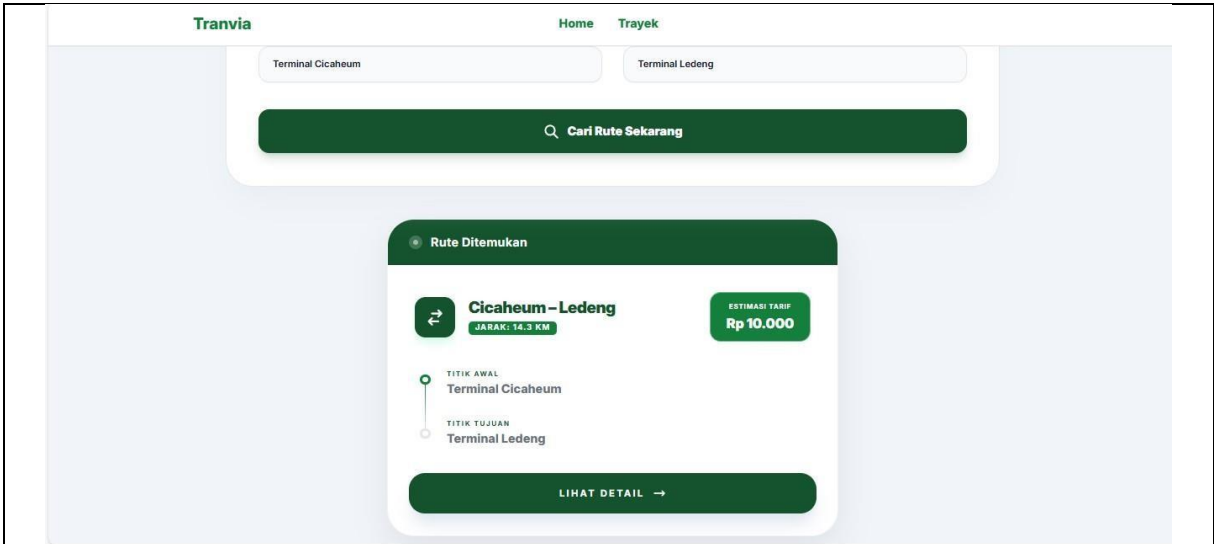


Gambar 5. 1 Implementasi AntarMuka Halaman Home

Deskripsi Pada Implementasi Tampilan menggunakan laravel 11 dan tailwind css. Tampilan pertama adalah Home.blade.php. Ada Navbar dengan konten pertama yaitu gambar dengan teks pembukaan beserta button Cari Rute. Konten kedua yaitu daftar tabel . lalu konten ketiga deskripsi singkat Tranvia.

Tabel 5. 8 Implementasi AntarMuka Halaman Pencarian



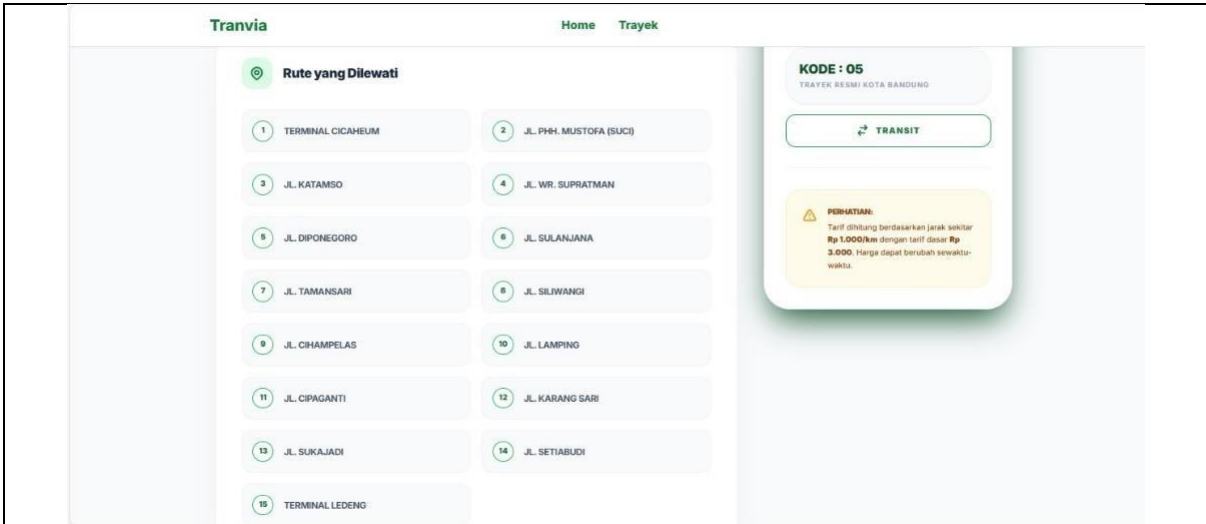


Gambar 5. 2 Implementasi AntarMuka Halaman Pencarian

Deskripsi Pada Implementasi Tampilan menggunakan laravel 11 dan tailwind css. Tampilan pertama ada di Pencarian.blade.php dengan isi navbar, form pencarian, hasil card. User mengisi titik awal dan tujuan dengan "Terminal atau jalan" lalu klik button cari rute, lalu card muncul dengan hasil pencariannya.

Tabel 5. 9 Implementasi AntarMuka Halaman Detail





Gambar 5. 3 Implementasi AntarMuka Halaman Detail

Deskripsi Pada Implementasi Tampilan menggunakan laravel 11 dan tailwind css. Tampilan pertama ada di Detail.blade.php dengan isi Map yang menampilkan titik rute, nama angkot, kode, jarak, tarif, gambar dan rute yang dilewati

Tabel 5. 10 Implementasi AntarMuka Halaman Daftar Trayek



Gambar 5. 4 Implementasi AntarMuka Halaman Daftar Trayek

Deskripsi Pada Implementasi Tampilan menggunakan laravel 11 dan tailwind css. Tampilan pertama ada di Pencarian.blade.php dengan isi navbar, form pencarian, hasil. User isi titik awal dan tujuan dengan "Terminal atau jalan" lalu klik button cari rute, lalu card muncul dengan hasil pencariannya.

5.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak memiliki tujuan untuk mengevaluasi kualitas serta memastikan bahwa perangkat lunak beroperasi sesuai dengan kebutuhan dan harapan pengguna. Proses ini dilakukan untuk menemukan kesalahan atau bug yang mungkin muncul, sehingga dapat segera diperbaiki sebelum perangkat lunak digunakan oleh banyak orang. Salah satu teknik yang biasa digunakan dalam proses pengujian adalah black box testing. Teknik ini berfokus pada pengujian fungsi perangkat lunak tanpa perlu memahami kode di dalamnya. Dengan kata lain, pengujian hanya menilai apakah fitur-fitur yang ada bekerja seperti yang diinginkan dan sesuai dengan spesifikasi yang telah ditetapkan, tanpa memikirkan struktur atau logika di balik program tersebut.

Tabel 5. 11 Pengujian Perangkat Lunak

Kode Test Skenario	Test Skenario	Kode Test Case	Test Case	Test Data	Expected Result	Actual Result	Pass/Fail
TS - 01	Pengujian Fungsional Pencarian	TC01 - 1	Pengguna menginputkan titik awal dan titik tujuan berdasarkan "Terminal" dengan via	Titik awal : Terminal Kebon kelapa Titik tujuan : Terminal cicaheum	Sistem menampilkan dua Card hasil dengan rute yang sama tetapi berbeda via.	Sistem menampilkan dua Card hasil dengan rute yang sama tetapi berbeda via.	Pass
		TC01 - 2	Pengguna menginputkan titik awal dan tujuan berdasarkan "Terminal".	Titik awal : Terminal Ledeng Titik tujuan: Terminal margahayu	Sistem Menampilkan Card hasil sesuai dengan titik yang di input.	Sistem Menampilkan Card hasil sesuai dengan titik yang di input.	Pass
		TC01 - 3	Pengguna menginputkan titik awal dan tujuan berdasarkan "Terminal" tetapi dengan arah sebaliknya.	Titik awal : Terminal Margahayu Titik tujuan: Terminal ledeng	Sistem menampilkan card hasil sesuai dengan rute sebenarnya.	Sistem menampilkan card hasil sesuai dengan rute sebenarnya.	Pass
		TC01 - 4	Pengguna menginputkan titik awal dan titik tujuan berdasarkan "Terminal – Jalan".	Titik awal : Terminal Ledeng Titik tujuan : Jl. Cihampelas	Sistem menampilkan card hasil dengan informasi angkot yang sesuai titik awal dan titik tujuan	Sistem menampilkan card hasil dengan informasi angkot yang sesuai titik awal dan titik tujuan	Pass
		TC01 –5	Pengguna menginputkan titik	Titik awal : Jl. Setiabudi	Sistem menampilkan	Sistem menampilkan	Pass

			awal dan titik tujuan berdasarkan nama "jalan".	Titik Tujuan; Jl. Ahmad Yani	card hasil dengan informasi angkot yang sesuai dengan nama jalan.	card hasil dengan informasi angkot yang sesuai dengan nama jalan.	
		TC01 - 6	Pengguna Menginputkan Titik awal dan titik tujuan dengan rute berbeda atau transit	Titik awal : Terminal cicaheum Titik Tujuan : Terminal margahayu	Sistem menampilkan 2 card hasil dengan informasi rute yang berbeda atau sesuai dengan yang di inputkan pada titik.	Sistem tidak menampilkan 2 card hasil, tetapi tidak error	Fail
		TC01 - 7	Pengguna tidak mengisi titik awal dan titik tujuan	Titik Awal : Titik Tujuan :	Muncul pesan bawaan	Muncul pesan bawaan	Pass
		TC01 - 8	Pengguna memasukan titik awal dan titik tujuan dengan simbol	Titik awal ; !@#\$% Titik tujuan : \$&*%	Muncul pesan eror "Pastikan Ejaan titik benar".	Muncul pesan eror "Pastikan Ejaan titik benar".	Pass
		TC01 - 9	Pengguna memasukan titik awal dan titik tujuan dengan nama jalan asal	Titik awal ; usyfgr Titik tujuan : jhdfgh	Muncul pesan eror "Titik asal dan tujuan tidak valid. Pastikan Ejaan titik benar".	Tidak Muncul Pesan error	Pass
TS - 2	Pengujian fungsi lihat detail	TC02 - 1	Pengguna klik button lihat detail	Klik button lihat detail	Masuk kehalaman detail dengan isi Map, nama angkot, kode ,warna, tarif dan rute yang di lewati sesuai titik awal dan titik tujuan	Masuk kehalaman detail dengan isi Map, nama angkot, kode ,warna, tarif dan rute yang di lewati sesuai titik awal dan titik tujuan	Pass
		TC01 - 2	Pengguna Klik button Transit pada halaman detail	Klik button Transit	Masuk ke halaman detail transit untuk data angkot 2	Masuk ke halaman detail transit untuk data angkot 2	Fail

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penyelidikan, penerapan serta pengujian situs web rute pencarian angkot dengan menggunakan metode Breadth First Search, sejumlah Kesimpulan berikut bisa ditarik ;

1. Sistem pencarian rute angkot berbasis web berhasil dirancang dan dibangun untuk membantu masyarakat memperoleh informasi rute angkot di Kota Bandung Lebih lengkap.
2. Penerapan algoritma *Breadth First Search (BFS)* mampu menentukan jalur angkot dengan jumlah lintasan paling sedikit, sehingga rute yang dihasilkan bersifat optimal berdasarkan titik yang dilalui.
3. Aplikasi mampu menampilkan informasi pendukung secara lengkap, seperti nama rute, kode, tarif, warna angkot, serta visualisasi peta menggunakan Leaflet.js.
4. Proses pencarian rute dapat dilakukan dengan cepat hanya dengan memasukkan titik awal dan titik tujuan dengan menampilkan hasil cardnya.
5. Penerapan algoritma *Breadth First Search* pada aplikasi ini berhasil dirancang dengan menghasilkan satu rute perjalanan sebagai keluaran..

6.2 Saran

Berdasarkan hasil penelitian tugas akhir yang berkaitan dengan perancangan dan pengembangan sistem pencarian rute angkot menggunakan algoritma Breadth First Search (BFS), berikut adalah rekomendasi untuk pengembangan penelitian mendatang ;

1. Pengembangan selanjutnya dapat menambahkan fitur estimasi waktu tempuh, jarak perjalanan, Jam Operasional dan kondisi saat itu agar informasi yang diberikan lebih akurat dan informatif.
2. Diperlukan penambahan fitur alternatif rute yang mampu menampilkan lebih dari satu rute perjalanan, sehingga pengguna dapat memilih opsi rute yang paling sesuai dengan kebutuhan dan kondisi perjalanan.
3. Aplikasi dapat dikembangkan dalam versi mobile agar lebih mudah diakses oleh pengguna di lapangan.
4. Antarmuka pengguna dapat terus disempurnakan agar lebih intuitif dan ramah bagi semua kalangan.
5. Data rute angkot sebaiknya diperbarui secara berkala bekerja sama dengan instansi terkait agar informasi tetap valid dan relevan.
6. Sistem dapat dikembangkan dengan fitur pencarian berbasis lokasi pengguna (GPS) agar titik awal dapat terdeteksi secara otomatis.

DAFTAR PUSTAKA

- [ADI17] Aditiya Ramdani, *“Kajian Integrasi Rute Angkutan Umum di Kota Bandung”*, Volume 4, Nomor 3, November 2017.
- [ARY18] Arya Nugroho, Akhmadali, Sumiyattinah, *“Penentuan Trayek Jaringan Angkutan Umum di Pontianak Metropolitan Area Berbasis BRT (Bus Rapid Transit)”*, Volume 5, Nomor 2, Tahun 2018.
- [DEL16] Delima Zai, Haeni Budiati, Sunneng Sandino Berutu, *“Simulasi Rute Terpendek Lokasi Pariwisata di Nias dengan Metode Breadth First Search dan Tabu Search”*, Volume 1, Nomor 2, Mei 2016.
- [DIN24] Dini Nurul Azizah, Luthfi Dika Chandra, Muhammad Galuh Gumelar, Wien Kuntari, *“Implementasi Framework Laravel dalam Pembuatan Website Segitiga Motor dengan Metode Waterfall”*, Volume 2, Nomor 6, Desember 2024.
- [FED10] Feddy Setio Pribadi & Angraini Mulwinda, *“Pencarian Rute Terpendek dengan Menggunakan Algoritma Depth First, Breadth First, dan Hill Climbing (Studi Komparatif)”*, Volume 2, Nomor 1, November 2010.
- [JAE24] Jaemsyien Devgan Oktawijaya, Soffiana Agustin, *“Optimalisasi Sistem Digitalisasi Penerimaan dan Pengeluaran Barang dengan Laravel 11 pada PT Swadaya Graha Gresik”*, Volume 6, Nomor 3, Desember 2024.
- [MAN22] Manuel Kristo & Yusup Jauhari Shandi, *“Perancangan Aplikasi Pencarian Rute Perjalanan Angkutan Kota di Kota Bandung Berbasis Web Menggunakan Algoritma A*”*, Tahun 2022.
- [MBU24] M. Budi Hartanto, Yodhi Yuniarthe, Teuku Muhammad Fawa’ati, Ahmad Ikhwan, *“Pemanfaatan Leaflet.js dalam Implementasi Sistem Informasi Geografis (SIG) untuk Optimalisasi Pengelolaan Objek Pajak Bumi dan Bangunan di Dispenda Lampung Tengah”*, Volume 5, Nomor 1, Mei 2024.
- [MUH17] Muhammad Syaiful Buamona, James Timboeleng, Hendriek Karongkong, *“Analisis Pelayanan Transportasi Angkutan Kota di Kota Ternate”*, Volume 4, Nomor 3, Tahun 2017.
- [SEL15] Selvia Lorena Br. Ginting, Yogie Rinaldy Ginting, Tulus Prabudi, *“Penerapan Algoritma Best Path Planning untuk Aplikasi Pencarian Rute Transportasi Publik Berbasis Android”*, Volume 5, Nomor 2, September 2015.

- [SIN22] Sinung Suakanto, Randy Ferdiawan, Faishal Mufied Al Anshary, *“Perancangan Aplikasi Transportasi Angkot Berbasis Mobile untuk Penumpang Menggunakan Metode User Centered Design”*, Volume 8, Nomor 2, November 2022.
- [VIO19] Viona V. Wakari, Octavianus H. A. Rogi, Vicky H. Makarau, *“Daya Angkut Layanan Angkot Berdasarkan Jarak Jangkauan Masyarakat terhadap Jalur Trayek di Kota Manado”*, Volume 6, Nomor 3, Oktober 2019.
- [YUL24] Yuliana, Noviyanti, Mudawil Qulub, *“Implementasi Algoritma Depth First Search dan Breadth First Search pada Dokumen Akreditasi”*, Volume 7, Nomor 1, Februari 2024.



LAMPIRAN

Lampiran 1 Berita Acara

Dengan ini menerangkan bahwa mahasiswa yang bertanda tangan di bawah ini :

Nama : Raisa Isna Ainun

NRP : 21340112

Fakultas / Prodi : Teknik Teknik Informatika

Universitas : Universitas Pasundan

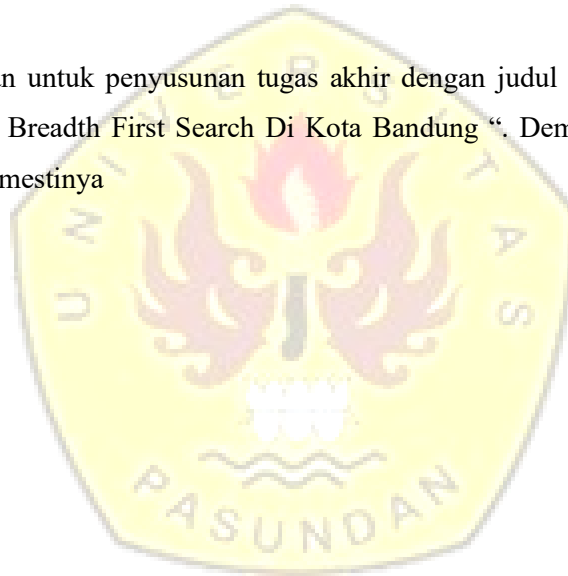
Telah melakukan wawancara dengan

Nama : Anton

Usia : 27 Tahun

Profesi : Supir Angkot

Guna keperluan penelitian untuk penyusunan tugas akhir dengan judul “ Perancangan Rute Angkot Menggunakan Algoritma Breadth First Search Di Kota Bandung “. Demikian berita acara ini dibuat untuk digunakan sebagai mestinya



Bandung, 16 juli 2025

Narasumber

Pewawancara

(Bapak Anton)

(Raisa Isna Ainun)

Lampiran 2 Laporan Wawancara

Nama : Anton
Umur : 27 Tahun
Waktu Pelaksanaan : Sekitar Jam 13.00 WIB
Tempat : Terminal Margahayu

Hasil Wawancara

- 1 Sudah Berapa Lama Menjadi Supir Angkot ?
Jawaban : 5 Tahun
- 2 Untuk Tarif Rute Ledeng – Margahayu Berapa ?
Jawaban : Dari ujung ke ujung sebesar Rp.15.000. Untuk Tarif jarak pendek Rp.3.000
- 3 Apakah Penumpang sering bertanya perihal rute atau lainnya?
Jawaban : Untuk Rute Jarang, Tetapi untuk tarif banyak di tanyakan.

Nama : Aif
Umur : 23 Tahun
Waktu Pelaksanaan : Sekitar Jam 14.00 WIB
Tempat : Jalan / Angkot

Hasil Wawancara

- 1 Sudah Berapa Lama Menjadi Supir Angkot ?
Jawaban : 2 Tahun Kurang
- 2 Untuk Tarif Rute Kalapa – Sukajadi Berapa ?
Jawaban : Dari cemara sampai alun alun sekitar Rp.7.000
- 3 Apakah Penumpang sering bertanya perihal rute atau lainnya?
Jawaban : Ya suka bertanya Tarif angkot.
- 4 Jam Operasional ?
Jawaban : Kalau dulu 24 jam perhari, kalau sekarang sampai jam 7.00 malam

Lampiran 3 Kuesioner

Kuesioner Penumpang Angkot

Pertanyaan Jawaban 21 Setelan

Jenis Kclamin
Teks jawaban singkat

Usia
Teks jawaban singkat

Domisili
Teks jawaban singkat

Apakah anda menggunakan angkota kota(angkot)
Teks jawaban singkat

Kuesioner Penumpang Angkot

Pertanyaan Jawaban 21 Setelan

Bagian 2 dari 3

Pertanyaan angkot
Pada bagian ini anda akan mengisi kuesioner seputar angkot, apakah anda memiliki kendala saat akan menggunakan transportasi umum salah satunya angkutan kota(angkot)

Apakah anda sering anda menggunakan angkot sebagai transportasi umum ?

1 2 3 4 5

Seberapa sering anda mencari informasi angkot (rute, warna angkot atau tarif)?

1 2 3 4 5

Kuesioner Penumpang Angkot

Pertanyaan Jawaban 21 Setelan

Apakah anda mengetahui bahwa setiap angkot memiliki warna yang berbeda tergantung rutenya ?

Ya
Tidak

Menurut anda, apakah informasi detail tentang rute angkot mudah ditemukan secara online ?

Ya
Tidak

Darimana biasanya anda mendapatkan informasi tentang angkot
Teks jawaban singkat

Apakah informasi angkot saat ini sudah cukup lengkap dan mudah di akses

Ya
Tidak

Kuesioner Penumpang Angkot

Pertanyaan Jawaban 21 Setelan

menurut anda informasi angkot apakah yang perlu ada di sebuah website atau aplikasi tentang angkot

- jadwal operasional
- jalur lengkap
- warna kendaraan
- Gambar
- Tarif
- Peta visual

Berikan saran anda terkait penyediaan informasi angkot agar lebih mudah di akses masyarakat

Teks jawaban singkat: _____

Kuesioner Penumpang Angkot

Pertanyaan Jawaban 21 Setelan

21 jawaban [Link ke Spreadsheet](#)

Ringkasan Pertanyaan Individual

Jenis Kelamin

Jenis Kelamin	Jumlah	Persentase
Laki-laki	2	9.5%
Perempuan	13	61.9%
Perempuan	1	4.8%
Perempuan	1	4.8%
Perempuan	1	4.8%
Perempuan	2	9.5%

Usia

Usia	Jumlah	Persentase
18 tahun	1	4.8%
19	1	4.8%
20	2	9.5%
21	3	14.3%
22	6	28.6%
22 tahun	1	4.8%
24	3	14.3%
25	2	9.5%
26	2	9.5%

Kuesioner Penumpang Angkot

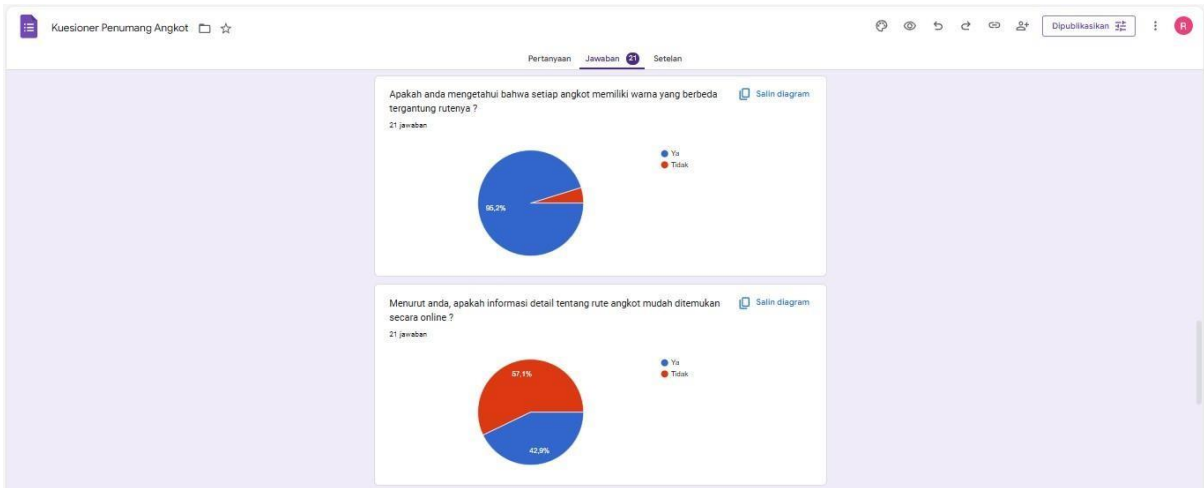
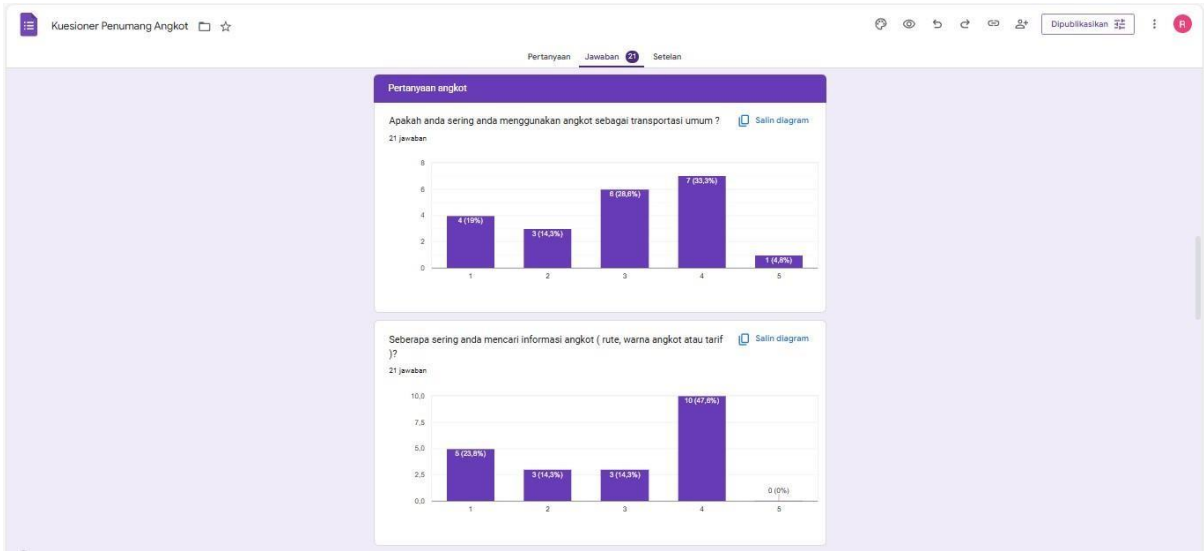
Pertanyaan Jawaban 21 Setelan

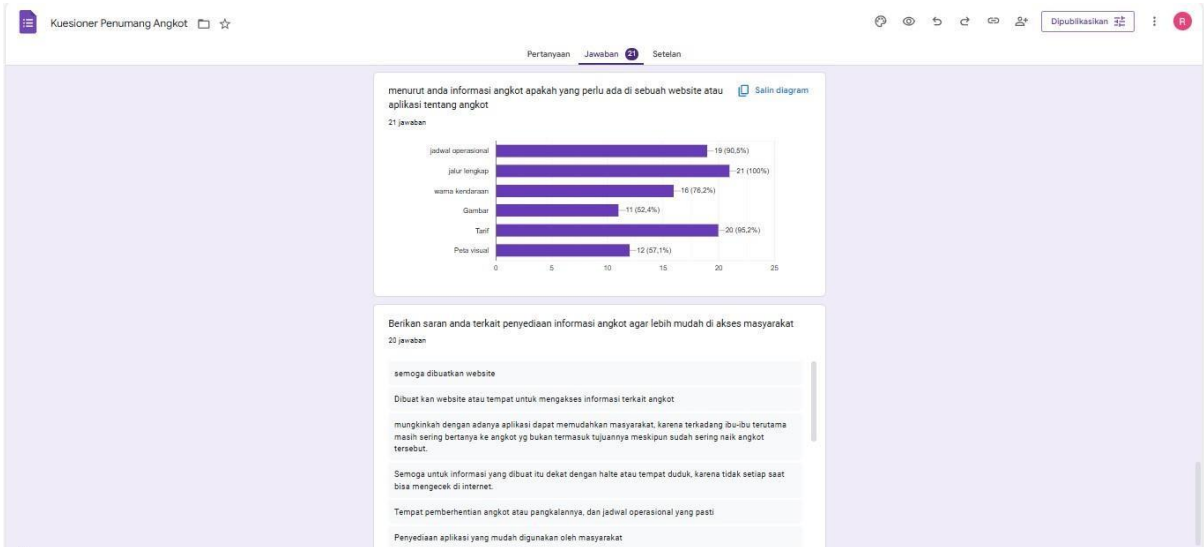
Domisili

Domisili	Jumlah	Persentase
Bandung	8	38.1%
Bandung barat	1	4.8%
Cimahi	1	4.8%
Cibincai Lembang	1	4.8%
Cimahi	1	4.8%
Karawang	1	4.8%
Kopo	1	4.8%
Kota Bandung	1	4.8%
Lembang	5	23.8%
Bandung	1	4.8%

Apakah anda menggunakan angkota kota(angkot)

Apakah anda menggunakan angkota kota(angkot)	Jumlah	Persentase
Iya	3	14.3%
Iya	1	4.8%
Iya	1	4.8%
Jarang Sekali	1	4.8%
Kadang kadang	1	4.8%
Kadang kadang	1	4.8%
Tidak	1	4.8%
Ya	7	33.3%
Ya	2	9.5%
Jarang	1	4.8%
Ya	2	9.5%





Lampiran 4 Dokumentasi

Jurusan kelapa - sukajadi

- Nama : Aif
- Umur : 23 tahun
- Sudah berapa lama menjadi supir Angkot : 2 tahun kurang
- Apakah penumpang sering menanyakan ^{Menanyakan} angkot, seberapa apabila ada ? 10 orang lebih
- Apakah penumpang Angkot sering menanyakan tarif ? Ya, suka menanyakan
- Berapa tarif angkot jurusan kelapa - sukajadi ? Rp. 7000 dari cemara sampai alun-alun
- Jam operasional ? kalau dulu 24 jam sehari, kalau sekarang semenjak ada apk online paling telat sampai jam 7.00 malam

Jurusan ledeng - Margahayu raya

- Nama : Anto
- umur : 27 tahun
- Sudah berapa lama menjadi supir angkot ? 5 tahun
- Tarif : dari ujung ke ujung harganya Rp. 15 ribu, untuk tarif jarak pendek Rp. 3000
- Apakah penumpang sering menanyakan angkot, seberapa banyak ? untuk rute jarang tetapi untuk tarif banyak ditanyakan.
- Jurusan kelapa - ledeng
- Nama : udan hidayat
- umur : 53 tahun
- sudah berapa lama menjadi supir angkot ? 30 tahun
- Tarif : 10 ribu
- jam operasional : jam 05.00 - 07.00 malam





Lampiran 1 Gambar dokumentasi



Lampiran 5 Kodingan

BfsServices

```
1  <?php
2
3  namespace App\Services;
4
5  use SplQueue;
6
7  class BfsService
8  {
9      public function bfs(array $graph, string $startInput, string $goalInput)
10     {
11         // Normalisasi input agar cocok dengan key di Graph (lowercase)
12         $startNode = strtolower(trim($startInput));
13         $endNode = strtolower(trim($goalInput));
14
15         // Validasi: Cek apakah titik awal ada di dalam peta/graph
16         if (!isset($graph[$startNode])) {
17             return null; // Titik awal tidak dikenal di sistem rute
18         }
19
20         // --- Mulai Algoritma BFS ---
21
22         // Queue menyimpan: [NamaNodeSaatIni, ArrayJalurYangSudahDilewati]
23         $queue = new SplQueue();
24         $queue->enqueue([$startNode, []]);
25
26         // node yang sudah dikunjungi
27         $visited = [$startNode => true];
28
29         while (!$queue->isEmpty()) {
30             // Ambil antrian paling depan (FIFO)
31             [$currentNode, $path] = $queue->dequeue();
32
33             // Jika belum sampai, cek tetangga dari node saat ini
34             if (isset($graph[$currentNode])) {
35                 foreach ($graph[$currentNode] as $edge) {
36                     $neighbor = $edge['node'];
37
38                     // Jika tetangga belum pernah dikunjungi, masukkan ke antrian
39                     if (!isset($visited[$neighbor])) {
40                         $visited[$neighbor] = true;
41
42                         // Salin jalur sebelumnya, lalu tambahkan langkah baru ini
43                         $newPath = $path;
44                         $newPath[] = [
45                             'from' => $currentNode,
46                             'to' => $neighbor,
47                             'id_rute' => $edge['id_rute']
48                         ];
49
50                         // Masukkan ke antrian untuk diperiksa selanjutnya
51                         $queue->enqueue([$neighbor, $newPath]);
52                     }
53                 }
54             }
55         }
56
57         return null;
58     }
59 }
60
61
62
63
64 }
```

Lampiran 2 Gambar BfsServices

Gambar di atas menunjukkan BfsService yang berfungsi menangani penerapan algoritma Breadth First Search (BFS) untuk mencari rute angkot. BfsService berperan sebagai logika utama yang digunakan oleh PencarianController dalam menjalankan proses pencarian sehingga menghasilkan rute angkot yang sesuai dengan input pengguna.

GraphBuilderServices

```
1 <?php
2
3 namespace App\Services;
4
5 use App\Models\Route;
6
7 class GraphBuilderService
8 {
9     public function buildGraph()
10     {
11         $graph = [];
12
13         // Ambil rute beserta titik-titiknya yang sudah diurutkan berdasarkan 'urutan'
14         $routes = Route::with(['titikRoutes' => function ($query) {
15             $query->orderBy('urutan', 'asc');
16         }])->get();
17
18         foreach ($routes as $route) {
19             $titiks = $route->titikRoutes;
20             $jumlahTitik = count($titiks);
21
22             // Loop setiap titik untuk membuat koneksi (Edge) antar titik
23             for ($i = 0; $i < $jumlahTitik - 1; $i++) {
24                 $currentInfo = $titiks[$i];
25                 $nextInfo = $titiks[$i + 1];
26
27                 // 1. Normalisasi nama (kecilkan huruf & hapus spasi berlebih agar pencarian akurat)
28                 $nodeAsal = strtolower(trim($currentInfo->nama_titik));
29                 $nodeTujuan = strtolower(trim($nextInfo->nama_titik));
30
31                 // Inisialisasi array node jika belum ada
32                 if (!isset($graph[$nodeAsal])) $graph[$nodeAsal] = [];
33                 if (!isset($graph[$nodeTujuan])) $graph[$nodeTujuan] = [];
34
35                 // 2. Tambahkan Jalur MAJU (A -> B)
36                 $graph[$nodeAsal][] = [
37                     'node' => $nodeTujuan,
38                     'id_rute' => $route->id_rute,
39                     'original_name' => $currentInfo->nama_titik // Simpan nama asli
40                 ];
41
42                 // Fitur "Dua Arah": Mengasumsikan angkot bisa pulang-pergi di jalur yang sama
43                 $graph[$nodeTujuan][] = [
44                     'node' => $nodeAsal,
45                     'id_rute' => $route->id_rute,
46                     'original_name' => $nextInfo->nama_titik
47                 ];
48             }
49         }
50
51         return $graph;
52     }
53 }
```

Lampiran 3 Gambar GraphBuilderServices

Gambar di atas menunjukkan GraphBuilderService yang berfungsi untuk menangani proses pencarian rute angkot dengan cara mengonversi data rute angkot ke dalam bentuk graf (graph). Graf tersebut kemudian digunakan untuk menentukan titik-titik rute yang akan dilalui dalam proses pencarian jalur.

Lampiran 6 Daftar Rute

Kode	Nama Angkot	Rute
01A	Abdul Muis(Kebon Kelapa – Cicaheum via binong)	Terminal Kebon Kelapa – Jl. Pungkur – Jl. Karapitan – Jl. Buah Batu – Jl. Banteng – Jl. Sancang – Jl. Lodaya – Jl. Martanegara – Jl. Turangga – Jl. Gatot Subroto – BSM – Binong – Jl. Kiara Condong – Jl. Jakarta – Jl. WR. Supratman – Jl. Katamso – Jl. Pahlawan – Jl. Cikutra – Jl. PHH. Mustofa (Suci) – Terminal Cicaheum
01B	Abdul Muis(Kebon Kelapa) – Cicaheum via aceh	Terminal Kebon Kelapa – Jl. Dewi Sartika – Jl. Kautamaan Istri – Jl. Balong Gede – Jl. Pungkur – Jl. Karapitan – Jl. Sunda – Jl. Lombok – Jl. Aceh – Jl. Taman Pramuka – Jl. Cendana – Taman WR. Supratman – Jl. Katamso – Jl. Pahlawan – Jl. Cikutra – Jl. PHH. Mustofa (Suci) – Terminal Cicaheum
02	Abdul Muis(Kebon Kelapa) – Dago	Terminal Kebon Kelapa – Jl. Dewi Sartika – Jl. Kautamaan Istri – Jl. Balong Gede – Jl. Pungkur – Jl. Karapitan – Jl. Sunda – Jl. Sumbawa – Jl. Belitung – Jl. Sumatera – Jl. Aceh – Jl. Sulawesi – Jl. Seram – Jl. RE Martadinata (Riau) – Jl. Ir. H. Juanda (Dago) – RS. Boromeus (Dago) – ITB (Jl. Ganesha, Dago) – Simpang Dago – Terminal Dago
03	Abdul Muis(Kebon Kelapa) – ledeng	Terminal Kebon Kelapa – Jl. Dewi Sartika – Jl. Kautamaan Istri – Jl. Balong Gede – Jl. Pungkur – Jl. Karapitan – Jl. Sunda – Jl. Sumbawa – Jl. Lombok – Jl. Banda – Jl. RE Martadinata (Riau) – BIP (Dago) – Jl. Merdeka – Jl. Aceh – Jl. Wastu Kencana – Jl. Rivai – Jl. Cipaganti – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Setiabudi – Terminal Ledeng
04	Abdul Muis(Kebon Kelapa) – elang	Terminal Kebon Kelapa – Jl. Pungkur – Jl. Otto Iskandardinata (Otista) – Jl. Ciateul – Jl. Astana Anyar – Jl. Panjunan – Jl. Kopo – Jl. Pasir Koja – Jl. Astana Anyar – Jl. Pagarsih – Jl. Nawawi – Jl. Aksan – Jl. Suryani – Jl. Holis – Jl. Bojong Raya – Jl. Cijerah – Jl. Sudirman – Jl. Rajawali Barat – Jl. Elang – Terminal Elang
05	Cicaheum – Ledeng	Terminal Cicaheum – Jl. PHH. Mustofa (Suci) – Jl. Katamso – Jl. WR. Supratman – Jl. Diponegoro – Jl. Sulanjana – Jl. Tamansari – Jl. Siliwangi – Jl. Cihampelas – Jl. Lamping – Jl. Cipaganti – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Setiabudi – Terminal Ledeng
06	Cicaheum - Ciroyom	Terminal Cicaheum – Jl. PHH. Mustofa (Suci) – Jl. Surapati (Suci) – Lapangan Gasibu (Surapati) – Jl. Panatayuda – Jl. Dipati Ukur – Simpang Dago – Jl. Sumur Bandung – Jl. Tamansari – Jl. Siliwangi – Jl. Cihampelas – Jl. Eyckman – RS. Hasan Sadikin – Jl. Pasir Kaliki – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Ciroyom – Terminal Ciroyom
07	Cicaheum - Ciwastra	Terminal Cicaheum – *Jl. PHH. Mustofa (Suci) – Jl. Surapati (Suci) – Jl. Sentot Alibasyah – Jl. Diponegoro – Jl. WR. Supratman – Jl. Ahmad Yani – Jl. Jakarta – Jl. Kiara Condong – Jl. Terusan Kiara Condong – Jl. Margacinta – Jl. Ciwastra – Terminal Ciwastra
08	Cicaheum - Cibaduyut	Terminal Cicaheum – *Jl. PHH. Mustofa (Suci) – *Jl. Katamso – *Jl. WR. Supratman – *Jl. Ahmad Yani – Jl. Jakarta – Jl. Kiara Condong – Jl. Sukarno-Hatta – Terminal Leuwi Panjang (Sukarno-Hatta) – Jl. Kopo – Jl. Cibaduyut – Terminal Cibaduyut
09	Stasiun - Dago	Terminal Dago – Jl. Ir. H. Juanda – Simpang Dago – ITB (Jl. Ganesha, Dago) – RS Boromeus (Dago) – BIP (Merdeka) – Jl. Merdeka – Jl. Perintis Kemerdekaan – Jl. Braga – Jl. Suniaraja – Jl. Stasiun Barat (Stasiun Bandung)
10	Stasiun – Sadang Serang	Terminal Stasiun – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Perintis Kemerdekaan – Braga – Jl. Lembong – Jl. Veteran – Jl. Sunda – Jl. Sumbawa – Jl. Lombok – Jl. Citarum –

		Jl. WR. Supratman – Jl. Katamso – Jl. Pahlawan – Jl. Cikutra Barat – Terminal Sadang Serang
11A	Stasiun – Ciumbuleuit Via Eyckman	Terminal Stasiun – Jl. Suniaraja – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Perintis Kemerdekaan – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cihampelas – Jl. Rivai – Jl. Cipaganti – Jl. Eyckman – Jl. Sederhana – Jl. Sempuna – Jl. Cipaganti – Jl. Setiabudi – Jl. Ciumbuleuit – UNPAR (Ciumbuleuit) – Terminal Ciumbuleuit
11B	Stasiun – Ciumbuleuit Via Cihampelas	Terminal Ciumbuleuit – Jl. Ciumbuleuit – UNPAR (Ciumbuleuit) – Jl. Cihampelas – Jl. Baba Husen – Jl. Sederhana – Jl. Pasir Kaliki – RS. Hasan Sadikin – Jl. Pasteur – Jl. Cihampelas – Jl. Rivai – Jl. Cipto – Jl. Pajajaran – Jl. Cicendo – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Jl. Kebon Jati – Terminal Stasiun
12	Stasiun – Gede Bage	Terminal Stasiun – Jl. Suniaraja – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Perintis Kemerdekaan – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cihampelas – Jl. Rivai – Jl. Cipaganti – Jl. Eyckman – Jl. Sederhana – Jl. Sempuna – Jl. Cipaganti – Jl. Setiabudi – Jl. Ciumbuleuit – UNPAR (Ciumbuleuit) – Terminal Ciumbuleuit
13	Stasiun - Sarijadi	Terminal Stasiun – Jl. Suniaraja – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Kebon Jukut – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Istana Plaza (Pasir Kaliki) – Jl. DR. Junjuran (Terusan Pasteur) – BTC (Pasteur) – Jl. Surya Sumantri – Universitas Maranatha (Surya Sumantri) – Jl. Lemah Nendeut (Sarijadi) – Jl. Sari Rasa (Sarijadi) – Jl. Sari Wangi (Sarijadi) – Jl. Sari Manah (Sarijadi) – Jl. Sari Asih (Sarijadi) – Terminal Sarijadi
14	Stasiun – Gunung - Batu	Terminal Stasiun – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Perintis Kemerdekaan – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cihampelas – Jl. Rivai – Jl. Rum – Jl. Gunawan – Jl. Otten – Jl. Pasteur – BEC (Pasteur) – Jl. Westhoff – Jl. DR. Junjuran (Terusan Pasteur) – Jl. Gunung Batu – Terminal Gunung Batu
15	Ledeng - Margahayu	Terminal Ledeng – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Cemara – Jl. Jurang – Jl. Sederhana – Jl. Eyckman – Jl. Cihampelas – Jl. Wastu Kencana – Jl. RE. Martadinata – Jl. Ahmad Yani – Cicadas (Ahmad Yani) – Jl. Kiara Condong – Jl. Sukarno-Hatta – Jl. Ranca Bolang (Margahayu Raya) – Terminal Margahayu
16	Dago – Riung Bandung	Terminal Dago – Jl. Ir. H. Juanda (Dago) – Simpang Dago – Jl. Dipati Ukur – Jl. Panatayuda – Jl. Surapati (Suci) – Jl. Sentot Alibasyah – Jl. Diponegoro – Jl. Citarum – Jl. RE. Martadinata – Jl. Laswi – Jl. Sukabumi – Jl. Ahmad Yani – Jl. Kiara Condong – Jl. Sukarno-Hatta – Margahayu Raya (Sukarno-Hatta) – Metro (Sukarno-Hatta) – Jl. Cipamolokan (Riung Bandung) – Jl. Riung Bandung – Terminal Riung Bandung
17	Dago – Pasar Induk Caringin	Terminal Dago – Jl. Cigadung Raya – Jl. Cikutra Barat – Jl. Pahlawan – Jl. Surapati (Suci) – Jl. Cikapayang – Jl. Tamansari – Jl. Sawunggaling – Jl. Rangga Gading – UNISBA & UNPAS (Tamansari) – Jl. Tamansari – Jl. Wastu Kencana – Jl. Purnawarman – Jl. Pajajaran – Jl. Cicendo – Jl. Rivai – Jl. Pasir Kaliki – Jl. Pajajaran – Jl. Arjuna – Jl. Supadio – Jl. Ciroyom – Jl. Rajawali Timur – Jl. Kebon Jati – Jl. Waringin – Jl. Sudirman – Jl. Jamika – Jl. Terusan Jamika – Jl. Sukamulya – Jl. Sukarno-Hatta – Jl. Babakan Ciparay – Pasar Induk Caringin (Sukarno-Hatta)
18	Dago – Panghegar Permai	Terminal Dipati Ukur – Jl. Hasanudin – Jl. Ir. H. Juanda – RS. Boromeus – Jl. Ganesha – ITB – Jl. Tamansari – Jl. Sawunggaling – UNISBA & UNPAS (Tamansari) – Jl. Wastu Kencana – Jl. RE. Martadinata – Jl. Banda – Jl. Belitung – Jl. Sumatera – Jl. Veteran – Jl. Ahmad Yani – Pasar Kosambi (Ahmad Yani) – Cicadas (Ahmad Yani) – Terminal

		Cicaheum – Jl. AH. Nasution – Sindanglaya (Nasution) – Ujung Berung (Nasution) – Jl. Cicukang – Jl. Cisaranten – Terminal Panghegar
19	Ciroyom – Sarijadi Via Sukajadi	Via Sukajadi Terminal Ciroyom – Jl. Arjuna – Jl. Pajajaran – Jl. Baladewa – Jl. Dursasana – Jl. Pasir Kaliki – Jl. Sukajadi – Jl. Sindang Sirna – Jl. Sindang Sirna – Jl. Geger Kalong Hilir – Jl. Sari Endah – Jl. Sari Jadi – Jl. Sari Manah (Sarijadi) – Jl. Sari Wangi (Sarijadi) – Terminal Sarijadi
20	Ciroyom – Bumi Asri	Terminal Ciroyom – Jl. Ciroyom – Jl. Garuda – Jl. Sudirman – Jl. Sukarno-Hatta – Jl. Holis – Jl. Bojong Raya – Jl. Cijerah – Bumi Asri
21	Ciroyom – Cikudapateuh	Terminal Ciroyom – Jl. Arjuna – Jl. Kebon Jati – Jl. Gardu Jati – Jl. Astana Anyar – Jl. Kopo – Jl. Peta – Jl. BKR – Jl. Moh. Ramdan – Jl. Banteng – Jl. Ahmad Yani – Jl. Cikudapateuh
22	Sederhana – Cijerah	Jl. Sederhana – Jl. Pasir Kaliki – Istana Plaza (Pasir Kaliki) – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Sudirman – Jl. Cijerah – Jl. Melong Asih
23	Sederhana – Cimindi	Jl. Sederhana – Jl. Sukajadi – Jl. Sukagalih – Jl. Cipedes Kidul – Jl. DR. Junjuran (Terusan Pasteur) – Jl. Gunung Batu – Cimindi
24	Ciwastra – Ujung Berung	Terminal Ciwastra – Riung Bandung – Jl. Sukarno-Hatta – Guruminda – Jl. Cisaranten – Jl. Cicukang – Jl. AH. Nasution (Raya Ujung Berung) – Pasar Ujung Berung
25	Cisitu – Tegallega	Terminal Cisitu – Jl. Cisitu Lama VIII – Jl. Cisitu – Jl. Sangkuriang – Jl. Siliwangi – Jl. Sumur Bandung – Jl. Tamansari – Jl. Siliwangi – Jl. Cihampelas – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cicendo – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Jl. Kebon Jati – Jl. Suniaraja – Terminal Stasiun – Jl. Dulatip – Pasar Baru – Jl. Sudirman – Jl. Astana Anyar – Jl. Kalipah Apo – Jl. Otto Iskandardinata (Otista) – Terminal Tegallega
26	Cijerah – Ciwastra	Jl. Cijerah – Jl. Bojong Raya – Jl. Holis – Jl. Caringin – Jl. Kopo – Jl. Peta – Jl. BKR – Jl. Moh. Ramdan – Jl. Sadakeling – Jl. Talaga Bodas – Jl. Pelajar Pejuang – Jl. Martanegara – Jl. Reog – Jl. Karawitan – Jl. Kliningan – Jl. Buah Batu – Jl. Sukarno-Hatta – Jl. Terusan Kiara Condong – Jl. Margacinta – Jl. Ciwastra – Terminal Ciwastra
27	Cicadas – Elang	Cicadas – Jl. Kiara Condong – Jl. Gatot Subroto – Jl. Burangrang – Jl. Sadakeling – Jl. Buah Batu – Jl. Gurame – Jl. Karapitan – Jl. Lengkong Kecil – Jl. Lengkong Besar – Jl. Pungkur – Terminal Kebon Kelapa – Jl. Dewi Sartika – Alun-Alun – Jl. Banceuy – Jl. Suniaraja – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Kebon Jukut – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Dadali – Jl. Kasuari – Jl. Rajawali Barat – Elang
28	Antapani – Ciroyom	Terminal Antapani – Jl. Cibatuan (Antapani) – Jl. Purwakarta (Antapani) – Jl. Jakarta – Jl. Sukabumi – Jl. Laswi – Stadion Persib (Ahmad Yani) – Jl. Ahmad Yani – Jl. Gudang Utara – Jl. Bangka – Jl. Belitung – Jl. Sumatera – Jl. Aceh – Jl. Merdeka – Jl. Perintis Kemerdekaan – Wastu Kencana – Jl. Pajajaran – Jl. Cicendo – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Istana Plaza (Pajajaran) – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Ciroyom – Terminal Ciroyom
29	Cicadas – Cibiru	Cicadas (Ahmad Yani) – Jl. Kiara Condong – Jl. Sukarno-Hatta – Margahayu Raya (Sukarno-Hatta) – Metro (Sukarno-Hatta) – Riung Bandung (Sukarno-Hatta) – Pasar Induk Gede Bage (Sukarno-Hatta) – Terminal Cibiru
30	Bumi Panyileukan – Sekemirung	Pasar Induk Gede Bage (Sukarno-Hatta) – Jl. Rumah Sakit – Jl. AH. Nasution – Ujung Berung (Nasution) – Sindanglaya (Nasution) – Terminal Cicaheum – Jl. PHH. Mustofa (Suci) – Jl. Cikutra – Jl. Pahlawan – Jl. Surapati (Suci) – Lapangan Gasibu (Surapati) – Jl. Panatayuda – Jl. Dipati Ukur – Jl. Ki Gede Utama – Jl. Hasanudin – RS. Boromeus

		(Juanda) – Jl. Ir. H. Juanda – Simpang Dago – *Jl. TB. Ismail – *Jl. Sadang Serang – Jl. Cikutra Barat – *Jl. Cikondang – Jl. Cibeunying Kolot – *Jl. Cigadung Raya – Terminal Awiligar
31	Sadang Serang – Caringin	Terminal Sadang Serang – Jl. Serang Serang – Jl. Tubagus Ismail – Simpang Dago – Jl. Tamansari – Jl. Sawunggaling – Jl. Tamansari – Jl. Wastu Kencana – Jl. RE. Martadinata – Jl. Purnawarman – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cicendo – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawung) – Jl. Pasir Kaliki – Jl. Kresna – Jl. Bima – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Sudirman – Jl. Cijerah – Jl. Bojong Raya – Jl. Holis – Jl. Caringin – Terminal Caringin
32	Cibaduyut – Karang Setra	Jl. Cibaduyut – Jl. Bojongloa – Jl. Peta – Jl. BKR – Jl. Moh. Toha – Jl. Pungkur – Terminal Kebon Kalapa – Jl. Pasir Koja – Jl. Pajagalan – Jl. Gardu Jati – Jl. Pasir Kaliki – Jl. Sukajadi – Karang Setra
33	Cibogo – Elang	Cibogo – Jl. Hercules – Jl. Pinggir Tol – Jl. Sukawarna – Jl. Suparmin – Jl. Sebelah IPTN – Jl. Pajajaran – Jl. Abdul Rahman Saleh – Jl. Garuda – Jl. Sudirman – Elang

