

PEMBANGUNAN APLIKASI REKOMENDASI RUTE ANGKUTAN KOTA MENGGUNAKAN ALGORITMA A* BERBASIS LARAVEL

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan program Strata 1,
Di Program Studi Teknik Informatika, Universitas Pasundan

oleh :

Rika Febriyanti E. Wenda
Nrp. 21.304.0125



**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 : Rika Febriyanti E. Wenda

Nrp. : 21.304.0125

Dengan judul :

“Pembangunan Aplikasi Rekomendasi Rute Angkutan Kota Menggunakan Algoritma A* Berbasis Laravel”



(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, 14 Januari 2026

Yang membuat pernyataan,

A 1000 Rupiah postage stamp with a signature over it. The stamp features a blue star and the text '1000', 'REPUBLIK INDONESIA', and 'POSTAL TELEGRAM'. The signature is written in black ink over the stamp.

(Rika Febriyanti E. Wenda)

NRP.21.304.0125

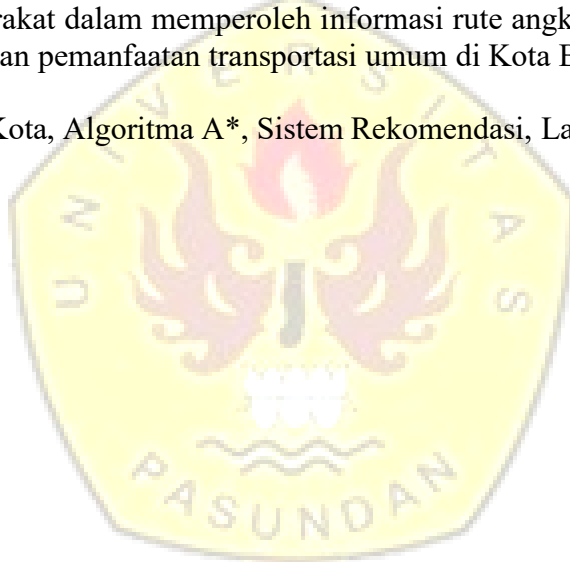
ABSTRAK

Angkutan kota (angkot) merupakan salah satu moda transportasi umum yang masih banyak digunakan oleh masyarakat di Kota Bandung. Namun, keterbatasan informasi yang jelas dan akurat mengenai rute angkot sering menimbulkan kebingungan bagi pengguna dalam menentukan jalur perjalanan yang harus ditempuh. Kondisi tersebut menyebabkan perjalanan menjadi kurang efisien dan menurunkan minat masyarakat dalam menggunakan angkutan kota.

Penelitian ini bertujuan untuk membangun aplikasi rekomendasi rute angkutan kota berbasis web yang dapat membantu pengguna dalam menentukan rute angkot yang optimal di wilayah Kota Bandung. Metode yang digunakan dalam penentuan rute adalah algoritma A* (A-Star), yang bekerja dengan memanfaatkan perhitungan biaya aktual dan estimasi jarak menuju tujuan untuk memperoleh jalur dengan bobot terendah.

Aplikasi dikembangkan menggunakan framework Laravel dengan basis data MySQL/MariaDB, serta memanfaatkan Leaflet.js untuk menampilkan visualisasi peta rute angkot dalam bentuk marker dan polyline. Sistem ini menyajikan informasi pendukung berupa nama rute, tarif, warna angkot, dan gambar angkot yang dapat diakses tanpa proses login. Hasil penelitian menunjukkan bahwa aplikasi mampu memberikan rekomendasi rute angkot secara informatif dan mudah digunakan. Dengan adanya aplikasi ini, diharapkan dapat membantu masyarakat dalam memperoleh informasi rute angkutan kota secara lebih efektif serta meningkatkan pemanfaatan transportasi umum di Kota Bandung.

Kata Kunci: Angkutan Kota, Algoritma A*, Sistem Rekomendasi, Laravel, Leaflet.



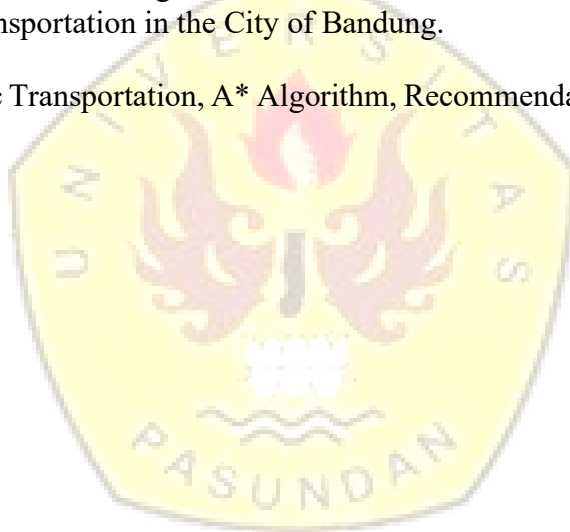
ABSTRACT

Urban public transportation (angkot) is one of the modes of transportation that is still widely used by the community in the City of Bandung. However, the lack of clear and accurate information regarding angkot routes often causes confusion for users in determining the travel paths they should take. This condition results in less efficient journeys and reduces public interest in using urban public transportation.

This study aims to develop a web-based angkot route recommendation application that can assist users in determining optimal angkot routes within the City of Bandung. The method used for route determination is the A* (A-Star) algorithm, which utilizes actual cost calculations and estimated distances to the destination to obtain routes with the lowest weight.

The application is developed using the Laravel framework with a MySQL/MariaDB database and utilizes Leaflet.js to display route map visualizations in the form of markers and polylines. The system provides supporting information such as route names, fares, angkot colors, and vehicle images, which can be accessed without requiring a login process. The results of this study indicate that the application is able to provide informative and user-friendly angkot route recommendations. With the implementation of this application, it is expected to help the public obtain angkot route information more effectively and encourage greater use of public transportation in the City of Bandung.

Keywords: Urban Public Transportation, A* Algorithm, Recommendation System, Laravel, Leaflet.



KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat, rahmat, dan penyertaan-Nya, penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Pembangunan Aplikasi Rekomendasi Rute Angkutan Kota Menggunakan Algoritma A* Berbasis Laravel”.

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

Penulis menyadari bahwa proses penyusunan laporan ini tidak terlepas dari berbagai hambatan dan tantangan. Namun, berkat bimbingan, dukungan, serta bantuan dari berbagai pihak, laporan ini dapat diselesaikan dengan baik. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Puji syukur ke hadirat Tuhan Yang Maha Esa, Sang Pemilik ilmu, atas segala limpahan rahmat, kekuatan, dan penyertaan-Nya yang tak terbatas. Tanpa izin dan kehendak-Nya, perjalanan panjang ini mustahil dapat penulis lalui hingga mencapai garis akhir.
2. Kedua orang tua dan keluarga tercinta, terima kasih atas segala tetesan keringat, pengorbanan yang tak terhitung, serta kasih sayang yang tak pernah menuntut balas. Skripsi ini adalah kado kecil untuk segala cinta dan dukungan moril maupun materil yang telah menjadi fondasi kekuatan saya selama ini.
3. Bapak Ade Sukendar, S.T., M.T., selaku Dosen Pembimbing, yang telah dengan sabar memberikan bimbingan, arahan, kritik, saran, serta masukan yang sangat berharga selama proses penyusunan Tugas Akhir ini.
4. Bapak Anggoro Ari Nurcahyo, S.T., M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Pasundan.
5. Bapak Wanda Gusdya Purnama, S.T., M.T., selaku Koordinator Tugas Akhir, terima kasih atas bantuan, bimbingan, dan koordinasi yang sangat membantu penulis dalam menempuh setiap tahapan proses administrasi dan akademik Tugas Akhir ini.
6. Seluruh dosen dan civitas akademika Program Studi Teknik Informatika Universitas Pasundan, yang telah memberikan ilmu pengetahuan, pengalaman akademik, serta bimbingan selama penulis menempuh pendidikan.
7. Putri, Adawiyah, Raisa dan Vina. Empat sosok yang bukan sekadar teman, melainkan 'rumah' bagi saya selama perjalanan ini. Terima kasih telah menjadi saksi jatuh bangunnya saya, yang tak pernah lelah meminjamkan bahu di saat penat, dan selalu hadir untuk merayakan setiap kemajuan kecil kita. Perjuangan ini terasa jauh

lebih ringan karena kita melaluinya dengan saling menggenggam tangan.

8. Teman-teman seperjuangan Teknik Informatika 2021, yang telah memberikan dukungan dan semangat selama proses penyusunan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki keterbatasan dan kekurangan, baik dari segi penyajian maupun isi. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi penyempurnaan di masa mendatang.

Akhir kata, penulis berharap semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi penulis khususnya, serta bagi pengembangan ilmu pengetahuan dan teknologi pada umumnya.

Bandung, 14 Januari 2026

Rika Febriyanti E. Wenda



DAFTAR ISI

ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR ISTILAH.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xv
DAFTAR SIMBOL.....	xvi
BAB 1 PENDAHULUAN.....	1-1
1.1. Latar Belakang.....	1-1
1.2. Identifikasi Masalah.....	1-2
1.3. Tujuan Tugas Akhir.....	1-3
1.4. Lingkup Tugas Akhir.....	1-3
1.5. Metodologi Tugas Akhir.....	1-5
1.6. Sistematika Penulisan Tugas Akhir.....	1-8
BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU.....	2-1
2.1. Teori yang digunakan.....	1-8
2.1.1. Pembangunan.....	1-8
2.1.2. Website.....	2-1
2.1.3. Rute.....	2-2
2.1.4. Angkutan Kota (Angkot).....	2-3
2.1.5. Waterfall.....	2-5
2.1.6. Algoritma A*.....	2-7
2.1.7. Framework.....	2-10
2.1.8. Laravel.....	2-11
2.1.9. Tailwind CSS.....	2-13
2.1.10. Leaflet.....	2-15
2.2. Penelitian Terdahulu.....	2-17
BAB 3 SKEMA PENELITIAN.....	3-1
3.1. Alur Penyelesaian Tugas Akhir.....	3-1
3.2. Perumusan Masalah.....	3-4

3.2.1.	Analisis Sebab Akibat.....	3-4
3.2.2.	Analisis Solusi Masalah.....	3-5
3.2.3.	Kerangka Berpikir Teoritis.....	3-6
BAB 4 ANALISIS DAN PERANCANGAN.....		4-1
4.1.	Analisis Perangkat Lunak	4-1
4.1.1.	Algoritma Pencarian Rute (Algoritma A*).....	4-1
4.1.1.1.	Simulasi Algoritma A*	4-1
4.1.2.	Model Analisis	4-10
4.1.2.1.	Diagram Aktivitas.....	4-10
4.1.2.2.	Identifikasi User Requirement.....	4-12
4.1.2.3.	Identifikasi Software Requirement	4-12
4.1.2.4.	Functional Requirement	4-12
4.1.3.	Model Use Case.....	4-13
4.1.3.1.	Diagram Use Case.....	4-13
4.1.3.2.	Definisi Aktor.....	4-14
4.2.	Perancangan	4-14
4.2.1.	Skenario Use Case	4-14
4.2.1.1.	UC – 01 Melakukan Pencarian Rute.....	4-15
4.2.1.2.	UC – 02 Melihat detail informasi	4-16
4.2.2.	Diagram Sequence.....	4-16
4.2.2.1.	UC – 01 Melakukan Pencarian Rute.....	4-17
4.2.2.2.	UC – 02 Melihat detail informasi	4-18
4.2.3.	Pemodelan Berbasis Kelas	4-19
4.2.3.1.	Mengidentifikasi Kelas.....	4-19
4.2.3.2.	Menentukan Atribut.....	4-21
4.2.3.3.	Menentukan Operasi	4-22
4.2.3.4.	Diagram Kelas	4-28
4.2.3.5.	Skema Database.....	4-29
4.2.3.6.	Perancangan Arsitektur	4-30
4.3.	Perancangan Antarmuka Pengguna.....	4-30
BAB 5 IMPLEMENTASI DAN PENGUJIAN		4-36
5.1	Kontruksi Perangkat Lunak.....	5-1
5.1.1.	Kebutuhan Kakas Perangkat Lunak.....	5-1
5.1.2.	Kebutuhan Kakas Perangkat Keras	5-2

5.2 Pengkodean Aplikasi	5-2
5.2.1. Struktur Aplikasi	5-2
5.2.2. Model.....	5-2
5.2.3. View.....	5-3
5.2.4. Controller	5-3
5.2.4.1. Services	5-4
5.3 Implementasi Antarmuka.....	5-4
BAB 6 KESIMPULAN DAN SARAN	6-1
6.1. Kesimpulan	6-1
6.2. Saran.....	6-1
DAFTAR PUSTAKA.....	i
LAMPIRAN.....	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 dimana 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.	Visited	Penanda yang digunakan dalam algoritma pencarian untuk menunjukkan bahwa suatu node telah dikunjungi sehingga tidak di proses kembali.
6.	Starting Point	Istilah untuk lokasi awal dari suatu objek.
7.	Open List	Tempat di mana data node yang sedang diproses dari starting point disimpan.
8.	Closed List	Tempat untuk menyimpan data node sebelum A yang merupakan bagian dari jalur terpendek yang telah ditemukan dengan sukses
9.	Informed Search Algorithm	algoritma pencarian dalam kecerdasan buatan (AI) yang menggunakan informasi tambahan (disebut <i>heuristic</i>) untuk memperkirakan seberapa jauh posisi saat ini dengan tujuan.
10.	Queue	Dalam struktur data, <i>Queue</i> adalah sekumpulan data yang mengikuti prinsip FIFO (First In, First Out) . Artinya, data yang pertama kali masuk adalah yang pertama kali keluar, persis seperti antrean orang saat membeli tiket di loket.
11.	Pseudocode	cara menuliskan logika pemrograman menggunakan bahasa manusia yang sederhana (seperti Bahasa Indonesia atau Inggris) namun tetap mengikuti struktur kode komputer.
12.	Heuristik	Fungsi estimasi dalam algoritma A* yang digunakan untuk memperkirakan jarak atau biaya dari suatu node menuju node tujuan.
13.	Algoritma A*	Algoritma pencarian rute terpendek yang mengkombinasikan biaya aktual perjalanan ($g(n)$) dan estimasi jarak ke tujuan ($h(n)$) untuk menentukan jalur dengan bobot terendah.
14.	Fishbone	Diagram sebab akibat yang digunakan untuk mengidentifikasi dan menganalisis penyebab suatu masalah
15.	Requirement	Kebutuhan atau spesifikasi yang harus dipenuhi oleh suatu sistem agar dapat berfungsi sesuai tujuan yang diharapkan.
16.	Use Case	Gambaran interaksi antara pengguna dan sistem yang menunjukkan fungsi-fungsi yang dapat dilakukan oleh sistem.
17.	Design	Proses perancangan sistem yang menggambarkan struktur, tampilan, dan alur kerja sebelum sistem dikembangkan.
18.	Database	Kumpulan data yang tersusun secara terstruktur dan saling berhubungan, yang disimpan secara sistematis agar dapat diakses, dikelola, dan diperbarui dengan mudah.
19.	Sequence	Diagram yang menggambarkan urutan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu.

No.	Istilah	Keterangan
20.	Framework	kerangka kerja berupa kumpulan fungsi, pustaka (library), dan aturan yang sudah jadi untuk membantu pengembang membangun aplikasi dengan lebih cepat dan terstruktur.
21.	MVC (Model View Controller)	Pola arsitektur perangkat lunak yang memisahkan logika data (Model), tampilan (View), dan pengendali alur aplikasi (Controller).
22.	Testing	Proses pengujian sistem untuk memastikan fungsi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan.



DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu.....	2-17
Tabel 3. 1 Alur Penyelesaian Tugas Akhir	3-1
Tabel 4. 1 Data Rute Angkot	4-3
Tabel 4. 2 Total Jarak Trayek 15.....	4-8
Tabel 4. 3 Diagram Aktivitas	4-11
Tabel 4. 4 User Requirement.....	4-12
Tabel 4. 5 Software Requirement.....	4-12
Tabel 4. 6 Functional Requirement.....	4-12
Tabel 4. 7 Deskripsi Use Case.....	4-13
Tabel 4. 8 Definisi Aktor	4-14
Tabel 4. 9 Skenario Use Case UC – 01 Melakukan Pencarian Rute	4-15
Tabel 4. 10 Skenario Use Case UC – 02 Melihat detail informasi.....	4-16
Tabel 4. 11 Diagram Sequence UC – 01 Melakukan Pencarian Rute	4-17
Tabel 4. 12 Diagram Sequence UC – 02 Melihat detail informasi.....	4-18
Tabel 4. 13 Identifikasi Kelas.....	4-19
Tabel 4. 14 Menentukan Atribut	4-21
Tabel 4. 15 Menentukan Operasi.....	4-22
Tabel 4. 16 Perancangan Antarmuka - Tampilan Halaman Home.....	4-31
Tabel 4. 17 Perancangan Antarmuka - Tampilan Halaman Pencarian	4-32
Tabel 4. 18 Perancangan Antarmuka - Tampilan Halaman Hasil Pencarian	4-33
Tabel 4. 19 Perancangan Antarmuka - Tampilan Halaman Detail.....	4-33
Tabel 4. 20 Perancangan Antarmuka - Tampilan Halaman Kategori.....	4-35
Tabel 4. 21 Perancangan Antarmuka - Tampilan Halaman Hubungi Kami	4-36
Tabel 5. 1 Kebutuhan Kakas Perangkat Lunak	5-1
Tabel 5. 2 Kebutuhan Kakas Perangkat.....	5-2
Tabel 5. 3 Model.....	5-2
Tabel 5. 4 View	5-3
Tabel 5. 5 Contreller	5-4
Tabel 5. 6 Services.....	5-4
Tabel 5. 7 Implementasi Antarmuka - Tampilan Halaman Home.....	5-5
Tabel 5. 8 Implementasi Antarmuka – Tampilan Halaman Pencarian	5-6
Tabel 5. 9 Implementasi Antarmuka - Tampilan Halaman Hasil Pencarian.....	5-7
Tabel 5. 10 Implementasi Antarmuka – Tampilan Halaman Detail Segment.....	5-8

Tabel 5. 11 Implementasi Antarmuka – Tampilan Halaman Kategori.....	5-9
Tabel 5. 12 Implementasi Antarmuka – Tampilan Halaman Detail.....	5-10
Tabel 5. 13 Implementasi Antarmuka – Tampilan Halaman Hubungi Kami	5-11
Tabel 5. 14 Pengujian Perangkat Lunak	5-12




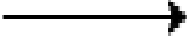
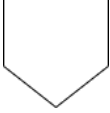
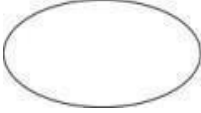



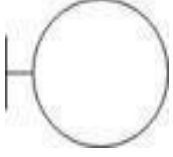
DAFTAR GAMBAR



Gambar 1. 1 Metodologi Penelitian.....	1-5
Gambar 2. 1 Waterfall Pressman.....	2-5
Gambar 2. 2 Laravel	2-11
Gambar 2. 3 MVC (Model, View, Controller).....	2-12
Gambar 2. 4 Tailwind CSS	2-13
Gambar 2. 5 Graph	2-15
Gambar 2. 6 Tampilan Peta Leaflet.....	2-16
Gambar 3. 1 Fishbone Diagram Sebab Akibat.....	3-4
Gambar 3. 2 Fishbone Diagram Analisis Solusi Masalah	3-5
Gambar 3. 3 Fishbone Diagram Kerangka Pemikiran Teoritis	3-6
Gambar 4. 1 Data Trayek Berdasarkan BPS Kota Bandung.....	4-2
Gambar 4. 2 Graf Pohon Simulasi A*	4-9
Gambar 4. 3 Diagram Aktivitas.....	4-11
Gambar 4. 4 Diagram Use Case	4-13
Gambar 4. 5 Diagram Sequence UC – 01 Melakukan Pencarian Rute	4-17
Gambar 4. 6 Diagram Sequence UC – 02 Melihat detail informasi.....	4-18
Gambar 4. 7 Diagram Kelas.....	4-28
Gambar 4. 8 Skema Database	4-29
Gambar 4. 9 Perancangan Arsitektur.....	4-30
Gambar 4. 10 Perancangan Antarmuka - Tampilan Halaman Home	4-31
Gambar 4. 11 Perancangan Antarmuka - Tampilan Halaman Pencarian	4-32
Gambar 4. 12 Perancangan Antarmuka - Tampilan Halaman Hasil Pencarian.....	4-33
Gambar 4. 13 Perancangan Antarmuka - Tampilan Halaman Detail	4-34
Gambar 4. 14 Perancangan Antarmuka - Tampilan Halaman Kategori	4-35
Gambar 4. 15 Perancangan Antarmuka - Tampilan Halaman Hubungi Kami.....	4-36
Gambar 5. 1 Implementasi Antarmuka - Tampilan Halaman Home	5-5
Gambar 5. 2 Implementasi Antarmuka – Tampilan Footer Halaman Home	5-5
Gambar 5. 3 Implementasi Antarmuka - Tampilan Halaman Pencarian	5-6
Gambar 5. 4 Implementasi Antarmuka - Tampilan Halaman Hasil Pencarian	5-7
Gambar 5. 5 Implementasi Antarmuka - Tampilan Halaman Detail Segment.....	5-8
Gambar 5. 6 Implementasi Antarmuka - Tampilan Halaman Kategori.....	5-9
Gambar 5. 7 Implementasi Antarmuka - Tampilan Halaman Detail Trayek	5-10
Gambar 5. 8 Implementasi Antarmuka - Tampilan Halaman Hubungi Kami	5-11

DAFTAR LAMPIRAN

Lampiran 1 Data Trayek Angkutan Kota Bandung.....	3
Lampiran 2 Kuesioner Penumpang Angkot.....	4
Lampiran 3 Kuesioner Penumpang Angkot.....	5
Lampiran 4 Kuesioner Penumpang Angkot.....	6
Lampiran 5 Persentase Jenis Kelamin Responden	7
Lampiran 6 Persentase Usia Responden	7
Lampiran 7 Persentase Domisili Responden.....	8
Lampiran 8 Persentase Penggunaan Angkot.....	8
Lampiran 9 Persentase Penggunaan Angkot.....	8
Lampiran 10 Persentase Pencarian Informasi	9
Lampiran 11 Persentase Informasi Warna Angkot.....	9
Lampiran 12 Persentase Mengenai Detail Informasi.....	9
Lampiran 13 Persentase Sumber Informasi Angkot.....	10
Lampiran 14 Persentase Keakuratan Informasi Angkot	10
Lampiran 15 Persentase Saran Responden.....	10
Lampiran 16 Saran Responden.....	11
Lampiran 17 Dokumentasi Wawancara.....	12

DAFTAR SIMBOL

Simbol	Nama Simbol	Keterangan
	Activity	Simbol ini digunakan untuk merepresentasikan aktivitas yang dilakukan oleh aktor yang terlibat.
	Transition	Simbol ini digunakan untuk merepresentasikan alur aktivitas.
	Off Page Connector	Simbol ini digunakan untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman yang lain.
	Input/Output	Simbol ini digunakan sebagai tanda untuk masukan atau keluaran dari aktivitas.
	Start State	Simbol ini digunakan untuk merepresentasikan awal dari sebuah aktivitas.
	End State	Simbol ini digunakan untuk merpresentasikan akhir dari sebuah aktivitas.
	Actor	Simbol ini digunakan untuk merepresentasikan orang yang berinteraksi dengan sistem atau aplikasi.
	View Object	Simbol ini digunakan untuk merepresentasikan antarmuka sistem yang berinteraksi dengan aktor pengguna.
Simbol	Nama Simbol	Keterangan

	<p>Controller Object</p>	<p>Simbol ini digunakan untuk merepresentasikan objek yang mengendalikan aliran jalannya sistem atau use case.</p>
	<p>Model Object</p>	<p>Simbol ini digunakan untuk merepresentasikan informasi yang tersimpan dan dipertahankan untuk jangka waktu yang lama, seperti data dalam database.</p>



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Transportasi kota (angkot) adalah salah satu jenis layanan transportasi umum yang masih menjadi bagian penting dari pergerakan masyarakat di Kota Bandung. Angkot memiliki keuntungan berupa cakupan rute yang luas dan biaya yang cukup terjangkau. Namun, di balik kelebihan tersebut, ada masalah utama yang sering dihadapi pengguna, yaitu kurangnya informasi yang jelas dan tepat mengenai rute angkot.

Isu ini menyebabkan banyak penumpang—khususnya orang-orang yang baru datang, mahasiswa, atau pelancong—kesulitan dalam menentukan angkot mana yang perlu dinaiki untuk mencapai lokasi tertentu. Informasi tentang rute dan jalur angkot selama ini hanya didapatkan melalui papan rute, keterangan dari sopir atau penumpang lain, juga dari pengalaman pribadi. Sumber informasi ini tidak selalu mudah diakses, kadang tidak konsisten, bahkan seringkali sudah tidak memenuhi kondisi terkini di lapangan.

Dampak dari kurangnya informasi ini sangat jelas. Penumpang sering kali merasa bingung, salah memilih angkot, atau harus turun di tengah perjalanan untuk bertanya lagi. Situasi ini tidak hanya mengurangi efektivitas perjalanan, tetapi juga menurunkan ketertarikan masyarakat untuk menggunakan angkot sebagai transportasi sehari-hari. Akibatnya, masyarakat lebih memilih menggunakan kendaraan pribadi atau layanan transportasi berbasis aplikasi, yang pada gilirannya memperburuk masalah kemacetan di Kota Bandung.

Oleh karena itu, diperlukan sebuah sistem informasi yang dapat memberikan rekomendasi rute angkot dengan cepat, akurat, dan mudah diakses. Pengembangan aplikasi transportasi umum berbasis situs web dapat membantu mempermudah akses informasi bagi masyarakat. Dengan fitur-fitur seperti rekomendasi rute tercepat, tarif kendaraan, gambaran kendaraan, dan juga warnanya. Sistem ini akan merekomendasikan angkot yang memiliki jalur paling efisien untuk mencapai tujuan, dengan mengedepankan rute terpendek atau biaya jalur terendah, sehingga pengguna dapat lebih cepat sampai tujuan. Hal ini tidak hanya meningkatkan pengalaman pengguna, tetapi juga mendorong masyarakat untuk beralih dari kendaraan pribadi ke transportasi umum, yang dapat mengurangi kepadatan lalu lintas dan dampak negatifnya pada lingkungan.

Untuk memastikan kualitas dan efisiensi pengembangan aplikasi transportasi umum ini, Metode *Waterfall* dipilih sebagai pendekatan utama dalam siklus pengembangan. Dengan mengikuti langkah-langkah berurutan dari analisis kebutuhan, desain sistem,

implementasi, pengujian, hingga penerapan, diharapkan aplikasi transportasi umum ini dapat dibangun secara sistematis dan terukur. Pada tahap desain sistem, algoritma pencarian A akan diintegrasikan sebagai pusat dari fitur rekomendasi angkot berdasarkan jarak terpendek atau biaya komputasional terendah. Pemilihan Algoritma A* didasarkan pada efektivitasnya dalam menentukan jalur optimal dengan mempertimbangkan biaya aktual antar simpul (*node*) dan perkiraan jarak menuju tujuan (*heuristic*).

Dengan penggunaan algoritma ini, sistem diharapkan dapat merekomendasikan angkot dengan rute paling sedikit untuk mencapai tujuan pengguna, sehingga perjalanan menjadi lebih cepat, praktis, dan efisien. Dengan demikian, aplikasi ini tidak hanya memberikan informasi angkot secara umum, tetapi juga menyarankan pilihan angkot yang memiliki rute paling optimal berdasarkan struktur jaringan rute yang ada.

Penerapan algoritma A* (A-Star) ini telah dibuktikan dalam studi sebelumnya oleh Rahmat Kurniawan R, Abdul Halim Hasugian, dan Putri Hanifah yang berjudul “Penentuan Rute dan Tarif perjalanan Angkutan Umum di Kota Medan Menggunakan Algoritma A*”. Penelitian tersebut secara langsung berfokus pada penentuan rute dan tarif dalam sistem angkutan kota dengan memanfaatkan A*. Dalam penelitian tersebut, pemodelan jaringan angkot dilakukan dalam bentuk graf yang mencakup *node* (titik), *edge* (jalur), dan bobot (jarak atau biaya), yang selanjutnya diproses dengan algoritma A* untuk menemukan rute yang optimal. Temuan dari studi ini menunjukkan bahwa algoritma A* dapat memberikan pencarian rute dan tarif yang cepat, tepat, dan efisien. Hasil ini menjadi dasar untuk pengembangan sistem dalam penelitian ini, karena menunjukkan bahwa A* dapat diterapkan dalam konteks angkot dan perjalanan untuk merekomendasikan jalur dengan efisiensi yang terbaik.

Dengan memperhatikan kondisi nyata serta dukungan dari penelitian sebelumnya, aplikasi transportasi umum berbasis situs web ini diharapkan dapat menjadi solusi digital yang praktis dan efisien, serta mendukung usaha Pemerintah Kota Bandung dalam menangani masalah kemacetan dan meningkatkan layanan transportasi publik.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka permasalahan yang dimunculkan pada tugas akhir pembangunan aplikasi rekomendasi rute angkutan kota menggunakan algoritma A* berbasis laravel adalah sebagai berikut.

1. Bagaimana merancang sistem rekomendesi angkot yang mudah digunakan pengguna?
2. Bagaimana algoritma A* diterapkan untuk mencari rute angkot dengan jarak atau

bobot jalur terendah?

3. Bagaimana sistem dapat menyajikan informasi tarif, warna dan gambar angkot dalam website?

1.3. Tujuan Tugas Akhir

Tujuan tugas akhir dari pembangunan aplikasi rekomendasi rute angkutan kota menggunakan algoritma A* berbasis laravel adalah sebagai berikut.

1. Merancang dan membangun sistem rekomendasi angkot berbasis web yang mudah digunakan oleh pengguna.
2. Mengimplementasikan algoritma pencarian A* untuk menentukan rute angkot dengan jarak terpendek atau bobot jalur paling rendah menuju tujuan pengguna.
3. Menyajikan informasi pendukung yang relevan bagi pengguna seperti tarif, warna, dan gambar angkot dalam tampilan website secara informatif dan menarik.

1.4. Lingkup Tugas Akhir

Penyelesaian tugas akhir dari pembangunan aplikasi rekomendasi rute angkutan kota menggunakan algoritma A* berbasis laravel ini di batasi sebagai berikut.

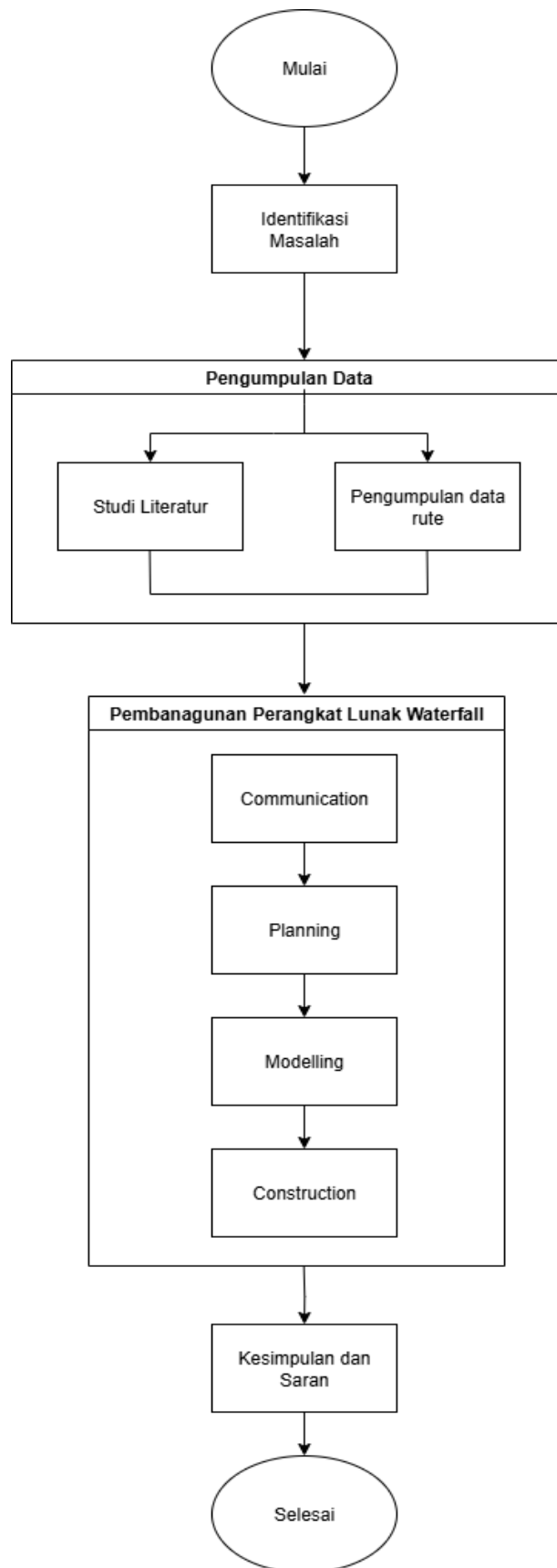
1. Sistem dibatasi pada trayek angkutan kota yang beroperasi di dalam wilayah administratif Kota Bandung. Trayek angkot yang melintas ke luar batas Kota Bandung (Kabupaten Bandung, Kota Cimahi, atau Kabupaten Bandung Barat) tidak termasuk dalam cakupan penelitian.
2. Algoritma yang digunakan dalam penentuan jalur adalah algoritma A*.
3. Sistem menampilkan informasi pendukung berupa: nama rute, tarif, warna angkot, dan gambar.
4. Data titik (*node*), jalur (*edge*), dan informasi mengenai angkot dikelola melalui antarmuka admin.
5. Sistem tidak mencakup pelacakan real-time GPS dan kondisi lalu lintas(kemacetan).
6. Rute yang digunakan adalah data statis yang telah dimasukkan ke dalam basis data secara manual.
7. Sistem ditujukan untuk pengguna umum, tanpa perlu melakukan login.
8. Penelitian ini hanya dibatasi sampai tahap Construction (implementasi aplikasi), sehingga tahap Deployment (pengujian langsung oleh pengguna/implementasi penuh di lapangan) tidak dilakukan.
9. Sistem diuji pada lingkungan pengembangan yang mendukung PHP 8.4,

database MySQL/MariaDB, dan dijalankan pada perangkat PC/Laptop sesuai dengan spesifikasi perangkat keras yang telah ditentukan.

10. Kerangka kerja utama pengembangan website menggunakan framework laravel.
11. Sistem menggunakan Leaflet.js sebagai library pemetaan untuk menampilkan peta pada halaman detail informasi rute, termasuk visualisasi jalur angkot dalam bentuk polyline, penanda lokasi (marker), dan elemen peta pendukung lainnya.



1.5. Metodologi Tugas Akhir



Gambar 1. 1 Metodologi Penelitian

Pada gambar 1.1 tersebut berisi mengenai tahapan penyelesaian tugas akhir.

Berikut adalah penjelasan pada metodologi di atas, sebagai berikut :

1. Identifikasi Masalah

Identifikasi masalah adalah tahap yang sangat penting dalam proses penelitian. Saat peneliti mengamati fenomena yang dapat dijadikan objek penelitian, langkah berikut yang harus dilakukan adalah mengidentifikasi masalah yang tampak dari fenomena tersebut. Dalam penelitian sosial, proses identificação masalah dapat dilakukan dengan menemukan masalah sosial yang terdeteksi. Dari sini peneliti melanjutkan untuk menggali lebih dalam, bisa melalui observasi, mempelajari literatur, atau melakukan survei awal. Identifikasi masalah adalah tindakan yang dilakukan peneliti di tahap awal riset. Peneliti melakukan identifikasi masalah dengan menguraikan apa yang diidentifikasi dan cara mengukur serta menghubungkan masalah tersebut dengan prosedur penelitian[RAM21].

2. Studi Literatur

Pada dasarnya, studi literatur berperan sebagai acuan untuk memberikan informasi yang relevan mengenai topik yang diteliti. Melalui studi literatur, peneliti mampu mengidentifikasi ciri-ciri, variabel, atau elemen penting dalam area yang sedang diteliti. Untuk memperoleh informasi langsung dari sumber aslinya, bisa berasal dari berbagai tempat, seperti buku, artikel jurnal, laporan penelitian, dan sumber akademis lainnya. Selanjutnya, data yang telah dikumpulkan diatur dan disusun secara sistematis agar dapat digunakan sebagai dasar untuk menghasilkan teori dan hipotesis penelitian[HAN25].

3. Pengumpulan Data

Pengumpulan data adalah salah satu langkah penting dalam mendukung keberhasilan penelitian. Pengumpulan data merupakan proses pengambilan data guna mendapatkan informasi yang relevan sesuai topik yang diteliti. Untuk mengumpulkan data secara akurat, peneliti harus menggunakan metode atau teknik pengumpulan data. Tujuan umum pengumpulan data adalah untuk mengumpulkan informasi yang dapat mendukung analisis dan interpretasi hasil penelitian.

4. Pembangunan aplikasi

Pembangunan aplikasi adalah kegiatan menyeluruh yang mencakup perancangan, pengembangan, pengujian, dan pemeliharaan perangkat lunak

agar bisa beroperasi di banyak *platform*. Tujuan dari pembuatan aplikasi adalah untuk menawarkan solusi digital yang sesuai dengan kebutuhan khusus pengguna atau perusahaan, mulai dari pekerjaan yang mudah hingga sistem yang lebih rumit.

Dalam pembuatan situs web untuk transportasi umum (angkot) ini diterapkan metode *waterfall*, untuk memastikan pembangunan aplikasi berjalan sistematis. Namun, pada penelitian ini, tahapan dibatasi hanya sampai pada tahap *Construction*. berikut adalah langkah-langkahnya:

- *Communication*

Tahap ini merupakan langkah awal untuk memahami kebutuhan dari sistem yang akan dibangun. Penulis melakukan diskusi dan pengumpulan data untuk mengidentifikasi masalah dalam transportasi angkot.

- Analisis Kebutuhan: Mengumpulkan informasi mengenai rute angkot, titik pemberhentian, tarif, serta kendala yang dihadapi penumpang dalam mencari informasi angkot.
- Identifikasi Pengguna: Menentukan siapa saja yang akan menggunakan web ini (masyarakat umum sebagai penumpang).

- *Planning*

Setelah data terkumpul, dilakukan perencanaan teknis agar proses pembangunan web terukur, dan terarah.

- Estimasi Sumber Daya: Menentukan perangkat lunak (seperti bahasa pemrograman, *framework*, dan *database*) serta perangkat keras yang dibutuhkan.
- Penjadwalan (*Scheduling*): Menyusun lini masa pengerjaan proyek mulai dari pengolahan rute hingga situs web siap dikodekan.
- Analisis Risiko: Memetakan kemungkinan kendala teknis, seperti akurasi data rute atau tampilan peta yang akan digunakan.

- *Modelling*

Tahap ini adalah proses menerjemahkan hasil analisis ke dalam bentuk rancangan sebelum masuk ke tahap pengkodean.

- Model Sistem: Menjelaskan alur kerja sistem menggunakan diagram (seperti *Use Case Diagram* atau *Flowchart*).
- Desain Arsitektur Data: Merancang struktur basis data (*Entity*

Relationship Diagram) untuk menyimpan data trayek, sopir, dan rute angkot.

- Desain Antarmuka (UI/UX): Membuat *wireframe* atau *mockup* tampilan situs web agar ramah pengguna (user-friendly), sehingga memudahkan penumpang melihat rute di layar ponsel atau komputer.

- *Construction*

Tahapan dimana rancangan diubah menjadi produk nyata.

- *Coding* (Pengkodean): Mengimplementasikan desain ke dalam baris kode menggunakan teknologi web (seperti HTML, CSS, JavaScript, dan bahasa *backend* lainnya).
- Implementasi Fitur: Membangun fitur utama seperti pencarian rute, visualisasi peta angkot, dan pengolahan data admin.
- Pengujian Internal: Melakukan pengecekan pada kode untuk memastikan tidak ada kesalahan logika (*bug*) dasar sebelum sistem dianggap selesai dalam batas ruang lingkup penelitian ini.

5. Kesimpulan dan saran

Pada bagian ini terdapat ringkasan singkat dan penjelasan yang jelas mengenai pembahasan yang telah disampaikan dalam keseluruhan penelitian serta penilaian terhadap hasil penelitian dan perkembangan aplikasi. Disamping itu, disertakan saran atau rekomendasi untuk langkah-langkah yang bisa diambil ke depan agar aplikasi dapat terus diperbaiki.

1.6. Teori yang digunakan

Berikut adalah beberapa teori yang digunakan dalam membuat penelitian tugas akhir pembangunan website rekomendasi angkutan kota (angkot) menggunakan algoritma A* dengan pendekatan metode waterfall berbasis laravel.

1.7. Sistematika Penulisan Tugas Akhir

Penulisan tugas akhir ini ditulis dengan sistematika sebagai berikut :

1. BAB 1 : Pendahuluan

Tujuan dari bagian pendahuluan adalah untuk memperkenalkan pembaca pada permasalahan penelitian; pentingnya isu ini yang didukung oleh data dari jurnal dan referensi yang relevan; urutan waktu atau alasan munculnya masalah; serta gagasan solusi yang diajukan oleh peneliti. Langkah-langkah di bagian pendahuluan mencakup ringkasan mengenai tinjauan pustaka, struktur konseptual, pernyataan mengenai masalah, serta hipotesis dan beberapa asumsi

yang menjadi dasar penelitian serta analisis yang logis dalam mengeksplorasi permasalahan penelitian. Pada pendahuluan, juga perlu diulas tentang variabel dan definisi operasional secara singkat. Walaupun penjelasan detail terdapat di bagian metodologi, ringkasan di tahap pendahuluan mengenai konsep atau definisi penting akan membantu pembaca memahami istilah-istilah sejak awal serta memahami apa yang akan dilaksanakan dalam studi tersebut[NUR08].

2. BAB 2 : Landasan Teori dan penelitian terdahulu

Landasan teori merupakan sebuah konsep yang tersusun dengan jelas dan teratur, mengandung variabel-variabel dalam penelitian karena menjadi basis yang kokoh untuk penelitian yang akan dilakukan. Definisi lain dari landasan teori adalah kumpulan definisi, konsep, dan proposisi yang telah disusun secara sistematis mengenai variabel-variabel dalam penelitian. Landasan teori ini akan berfungsi sebagai fondasi yang kuat untuk penelitian yang dilakukan. Oleh sebab itu, menciptakan landasan teori yang baik dalam penelitian menjadi sangat penting, karena landasan teori berperan sebagai fondasi dalam penelitian itu sendiri[RAM21].

3. BAB 3 : Skema Penelitian

Bab skema penelitian mencakup langkah-langkah atau alur dalam menyelesaikan tugas akhir, analisis masalah serta manfaat dari tugas akhir, kerangka pemikiran secara teoritis, dan profil lokasi penelitian.

4. BAB 4 : Analisis dan Perancangan

Bab analisis dan perancangan adalah bagian utama dari proses pengembangan solusi dalam proyek akhir. Bagian ini menjelaskan analisis tentang kebutuhan aplikasi dan perancangan aplikasi transportasi umum dengan menggunakan metode waterfall berbasis web sesuai dengan kebutuhan.

5. BAB 5 : Implementasi

Bab implementasi menguraikan secara teknis langkah-langkah dalam pembangunan sistem, yang mencakup teknologi yang diterapkan, langkah-langkah instalasi, pengaturan direktori, serta tampilan antarmuka pengguna. Tujuan utamanya adalah untuk memperlihatkan bahwa sistem telah dibangun dan beroperasi sesuai dengan spesifikasi yang telah dirancang.

6. BAB 6 : Penutup

Penutup dalam sebuah laporan penelitian merupakan bagian terakhir sebelum menyusun daftar pustaka. Penutup memiliki peranan yang sangat signifikan

dalam laporan penelitian, karena mencakup 2 (dua) elemen, yaitu kesimpulan dan rekomendasi (saran)[DRK21].

Penutup tidak sekadar menjadi pengakhir tulisan, melainkan berfungsi sebagai sintesis dari seluruh proses penelitian yang telah dilakukan. Bagian ini memberikan jawaban tegas atas permasalahan yang diangkat pada bagian awal laporan. Melalui penutup, pembaca dapat memahami esensi dari temuan penelitian tanpa harus membaca seluruh isi dokumen kembali.



BAB 2

LANDASAN TEORI DAN PENELITIAN TERDAHULU

2.1. Landasan Teori

2.1.1. Pembangunan

Pembangunan merupakan suatu proses yang terus-menerus dan dinamis, dengan tujuan untuk mengubah sebuah konsep yang tidak nyata atau keperluan tertentu menjadi sebuah perangkat lunak yang dapat digunakan, interaktif, dan memberikan manfaat bagi penggunanya. Proses pembuatan aplikasi memerlukan perencanaan yang baik, desain yang teliti, pelaksanaan yang mahir, pengujian yang seksama, peluncuran yang berhasil, serta pemeliharaan yang sigap.

Hasil akhir dari pembangunan perangkat lunak adalah sebuah produk digital yang siap untuk divalidasi. Keberhasilan dalam tahap pembangunan diukur dari sejauh mana perangkat lunak tersebut mampu merepresentasikan solusi atas kebutuhan pengguna yang telah diidentifikasi sebelumnya, serta stabilitasnya saat dioperasikan dalam berbagai kondisi.

2.1.2. Website

Situs web ialah halaman yang memberikan informasi dan dapat diakses melalui internet, sehingga bisa diunjungi dari mana saja selama terhubung dengan jaringan internet. Proses pembuatan sistem informasi atau pengembangan sistem (*system development*). Pengembangan sistem informasi dapat diartikan sebagai aktivitas yang bertujuan untuk menciptakan sistem informasi berbasis komputer guna menyelesaikan masalah di organisasi atau memanfaatkan peluang (*opportunities*) yang ada[ELV23].

Website adalah kumpulan halaman-halaman web yang ada dalam satu domain yang mengandung informasi. Umumnya, website terdiri dari banyak halaman web yang saling terhubung. Keterkaitan antar halaman web ini dikenal dengan istilah *hyperlink*, sementara kata yang digunakan sebagai penghubung disebut *hypertext*.

Domain adalah nama unik yang dimiliki oleh suatu institusi agar dapat diakses di internet, seperti *lintau.com*, *yahoo.com*, *google.com*, *ephi.web.id*, dan lain-lain. Untuk mendapatkan sebuah domain, kita harus melakukan pendaftaran di registrar yang ditunjuk. Istilah lain yang sering kita dengar terkait dengan website adalah *homepage*. *Homepage* adalah halaman utama dari suatu domain. Contohnya, ketika kita membuka website *www.lintau.com*, halaman pertama yang muncul adalah *homepage*. Jika Anda mengklik menu dan berpindah ke halaman lain, itu disebut *web page*, sedangkan total isi atau konten dari domain disebut sebagai website.

Seiring dengan pesatnya kemajuan teknologi informasi, website juga berkembang

dengan cepat. Dalam pengelompokan tipe website, biasanya dibagi berdasarkan fungsi, sifat, dan bahasa pemrograman yang digunakan. Tipe-tipe website berdasarkan sifatnya meliputi:

1. Website dinamis, yaitu website yang menyajikan konten yang selalu berubah. Contohnya adalah website berita.
2. Website statis, yaitu website dengan konten yang jarang sekali diubah. Misalnya, website profil organisasi, dan yang serupa.

Berdasarkan tujuannya, website dikelompokkan menjadi:

1. Website pribadi, yaitu website yang berisi informasi mengenai individu.
2. Website korporat, yaitu website yang dimiliki oleh sebuah perusahaan.
3. Website portal, yaitu website yang menyediakan berbagai layanan, mulai dari berita, email hingga layanan lainnya.
4. Website forum, yaitu website yang digunakan sebagai tempat diskusi. Disamping itu, ada juga website *e-Government*, *e-Banking*, *e-Payment*, *e-Procurement*, dan sebagainya.

Dari segi bahasa pemrograman yang digunakan, website dibagi menjadi:

1. *Server Side*, yaitu website yang memanfaatkan bahasa pemrograman yang bergantung pada adanya server. Contohnya, PHP, ASP, dan lain-lain. Jika server tidak tersedia, website dengan bahasa pemrograman tersebut tidak dapat berfungsi dengan baik.
2. *Client Side*, adalah website yang tidak memerlukan server untuk berfungsi, cukup diakses melalui browser. Contohnya adalah HTML. [YUH09].

2.1.3. Rute

Rute merujuk pada jarak atau jalur yang perlu diikuti secara berurutan. Rute yang paling cepat adalah jarak yang harus ditempuh dari lokasi awal menuju lokasi akhir dalam waktu tertentu. Secara lebih spesifik, rute tercepat adalah jalur terpendek yang dibutuhkan untuk sampai dari satu lokasi ke lokasi yang lain[HAR21].

Rute merujuk pada perjalanan atau jalur yang dilalui dari satu lokasi ke lokasi lainnya dalam jangka waktu tertentu. Rute transportasi umumnya ditentukan dan berada di area yang diprediksi memiliki calon penumpang yang dapat dilayani. Secara umum, jalur transportasi yang ada untuk memenuhi kebutuhan masyarakat di sebuah daerah jumlahnya lebih dari satu. Dalam sistem jaringan rute tersebut, dua atau lebih rute akan bertemu. Di titik-titik ini, memungkinkan terjadinya perubahan rute, karena kenyataannya seorang penumpang tidak hanya menggunakan satu jalur perjalanan dari asal menuju

tujuan[CHA21].

Dalam manajemen transportasi perkotaan, rute tidak hanya dipandang sebagai garis statis, melainkan sebagai sebuah sistem dinamis yang melibatkan variabel waktu, kepadatan, dan aksesibilitas. Karakteristik rute yang efisien ditentukan oleh kemampuannya dalam menghubungkan titik asal (*origin*) dan titik tujuan (*destination*) dengan hambatan seminimal mungkin. Hal ini menjadi sangat krusial pada moda transportasi umum seperti angkot, di mana rute yang ditetapkan harus mencakup area pemukiman, pusat perbelanjaan, dan fasilitas publik lainnya guna menjamin keberlanjutan layanan.

Dari perspektif teknologi informasi, pengelolaan rute berkaitan erat dengan representasi data spasial. Rute diubah menjadi sekumpulan titik koordinat (*nodes*) dan garis penghubung (*edges*) yang membentuk suatu graf jaringan. Dalam implementasi sistem informasi berbasis web, penyajian informasi rute harus bersifat interaktif dan informatif, mencakup detail mengenai arah lintasan, titik transit, serta identitas armada. Keakuratan representasi visual rute pada peta digital sangat memengaruhi tingkat kepercayaan pengguna terhadap sistem, karena pengguna cenderung memilih rute yang memberikan kepastian informasi.

Oleh sebab itu, rute merupakan konsep mendasar yang memperlihatkan urutan, jalan, dan arah pergerakan atau perkembangan, baik itu untuk objek fisik, informasi, maupun perjalanan hidup, yang bertujuan mencapai lokasi atau hasil yang diinginkan.

2.1.4. Angkutan Kota (Angkot)

Angkutan kota, yang sering disingkat angkot, adalah jenis kendaraan yang berbentuk bus kecil dan biasanya memiliki cat warna-warni serta dilengkapi dengan rak di atap untuk menyimpan barang-barang penumpang. Angkot mulai muncul sejak tahun 1960-an, ketika pemerintah berusaha menyediakan transportasi umum untuk masyarakat yang tinggal di kota dengan jumlah yang terus bertambah. Pada awalnya, layanan angkutan kota ini hanya ada di Jakarta, tetapi dengan cepat menyebar ke kota-kota besar lainnya di Pulau Jawa seperti Bandung, Surabaya, dan Yogyakarta. Sampai sekarang, angkutan kota telah ada di semua kota yang terdapat di Indonesia[ALT24].

Angkutan kota merupakan salah satu cara untuk berkomunikasi di dalam dan antar kota di Indonesia, yang banyak dimanfaatkan, dengan menggunakan mobil jenis minibus atau van yang dikemudikan oleh seorang pengemudi dan kadang dibantu oleh seorang kondektur. Setiap rute dibedakan berdasarkan warna kendaraan atau melalui kode angka. [LAB21].

Dalam perkembangannya, angkutan kota memainkan peran krusial sebagai

transportasi *feeder* atau pengumpan yang menghubungkan kawasan pemukiman padat penduduk dengan jalur transportasi utama. Karakteristik utama angkot adalah sifatnya yang fleksibel dalam hal menaikkan dan menurunkan penumpang, meskipun tetap terikat pada lintasan trayek resmi. Hal ini menjadikan angkot sebagai tumpuan mobilitas karena tarifnya yang terjangkau dan jangkauannya yang luas hingga ke jalan-jalan sekunder.

2.1.5. Trayek

Trayek merupakan manifestasi dari pola pergerakan spasial yang disusun secara sistematis untuk menghubungkan ruang-ruang aktivitas manusia. Dalam konteks yang lebih luas, trayek tidak hanya dipandang sebagai jalur perjalanan, tetapi sebagai sebuah kesatuan teknis yang memiliki struktur dan hirarki. Keberadaan trayek berfungsi sebagai instrumen pengatur lalu lintas agar pergerakan dari titik asal ke titik tujuan menjadi lebih teratur dan efisien.

Secara lebih terperinci, pemahaman mengenai trayek mencakup beberapa dimensi teori sebagai berikut:

1. Komponen Fisik dan Geometris Trayek

Setiap trayek memiliki identitas fisik yang menentukan kualitas pergerakan di dalamnya. Karakteristik ini meliputi:

- Lintasan (Alignment): Garis kontur jalan yang diikuti oleh trayek, baik secara horizontal (tikungan) maupun vertikal (tanjakan/turunan).
- Lebar dan Kapasitas Jalan: Ruang fisik jalan yang tersedia yang mempengaruhi volume kendaraan yang dapat melintasi trayek tersebut dalam satu satuan waktu.
- Interseksi (Persimpangan): Titik-titik pertemuan antar trayek yang menjadi lokasi kritis dalam manajemen rute, karena di titik inilah terjadi potensi konflik arus atau pilihan perubahan arah perjalanan.

2. Struktur Hierarki Trayek

Dalam sistem tata kota, trayek diatur berdasarkan fungsi jalan yang dilaluinya:

- Trayek Arteri: Melayani angkutan utama dengan ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk yang dibatasi secara efisien.
- Trayek Kolektor: Melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk yang terbatas.
- Trayek Lokal: Melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk yang tidak dibatasi.

3. Parameter Teknis Operasional

Untuk mendefinisikan sebuah trayek dalam sistem informasi atau basis data, diperlukan parameter kuantitatif yang jelas:

- Jarak Tempuh Kumulatif: Perhitungan total panjang lintasan yang diperoleh dari penjumlahan setiap segmen jalan yang membentuk trayek tersebut.
- Headway (Waktu Antara): Interval waktu antara dua kendaraan yang berurutan pada titik yang sama dalam satu trayek (khusus pada transportasi umum).
- Variabel Hambatan: Faktor-faktor luar yang mempengaruhi efektivitas trayek, seperti keberadaan lampu lalu lintas, persimpangan sebidang (kereta api), atau penyempitan jalan (*bottleneck*).

4. Tipologi Alur Trayek

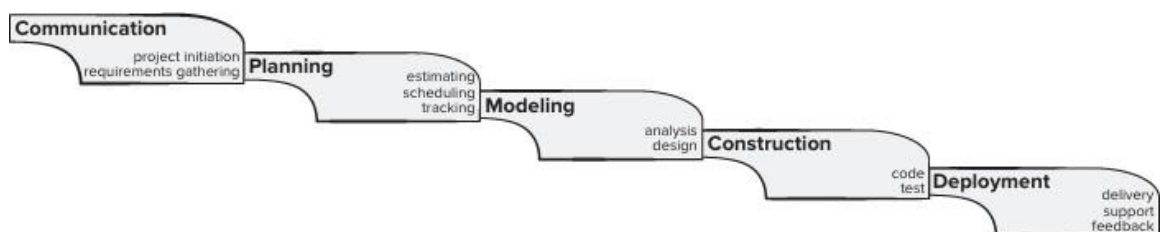
Berdasarkan bentuk lintasannya, trayek dapat dikategorikan menjadi beberapa pola:

- Pola Radial: Trayek yang memusat atau menyebar dari satu titik pusat (misal: pusat kota) menuju pinggiran kota.
- Pola Grid (Catur): Trayek yang membentuk jalur tegak lurus, memberikan banyak pilihan titik persimpangan bagi pengguna.
- Pola Linier: Trayek yang mengikuti jalur utama yang memanjang, biasanya mengikuti garis pantai atau jalan negara.

2.1.6. Waterfall

Berdasarkan berbagai sumber yang mengacu pada Pressman (2020), model waterfall ini telah mengalami beberapa kemajuan dalam proses pengembangannya. Tahapan metode waterfall terdapat dua jenis yaitu pertama: *requirement analysis, design, implementation, testing, dan maintenance*. Kedua yaitu *communication, planning, modeling, construction, dan deployment*[EKK21].

Berikut ilustrasi dari waterfall pada gambar 2.1 ini :



Gambar 2. 1 Waterfall Pressman

Berikut adalah proses untuk setiap tahapan pada model waterfall :

1. Communication

Communication, kegiatan, dan tugas yang dilakukan dalam fase *Communication* adalah

analisis kebutuhan awal sesuai dengan inisiasi proyek. Pada fase ini, terdapat aktivitas pengumpulan informasi yang akan dipakai dalam proses pengembangan[EKK21].

2. Planning

Langkah selanjutnya adalah fase perencanaan yang menjelaskan mengenai pengukuran tugas-tugas teknis yang akan dilaksanakan, kemungkinan risiko yang bisa muncul, sumber daya yang dibutuhkan untuk membangun sistem, produk yang ingin dihasilkan, jadwal pekerjaan yang akan dilakukan, dan pemantauan proses pembangunan sistem[ADH24].

3. Modeling

Langkah ketiga adalah *modeling* yang mencakup analisis dan desain. Dalam tahap ini, dilakukan perencanaan serta pemodelan dari segi struktur dan arsitektur perangkat lunak. Di tahap ini, tampilan perangkat lunak mulai nampak. Tujuan dari proses ini adalah supaya pengguna bisa memahami apa yang akan dilakukan ketika menggunakan perangkat lunak ini.

4. Construction

Langkah keempat ini adalah tahap untuk mengubah desain atau model tersebut menjadi kode yang dapat dipami oleh komputer. Setelah menyelesaikan langkah ini, langkah berikutnya adalah melakukan pengujian untuk menemukan potensi masalah yang mungkin muncul. Secara umum, langkah ini terbagi menjadi dua bagian, yaitu bagian mengkode dan bagian menguji.

5. Deployment (*Delivery, Support, Feedback*)

Langkah terakhir adalah tahap implementasi yang mencakup proses *delivery, support*, dan *feedback*. Di tahap ini, dilakukan pengujian penggunaan perangkat lunak kepada pelanggan, pemeliharaan, yang berkelanjutan, serta penilaian mengenai efektivitas perangkat lunak tersebut. Tujuannya adalah untuk memperoleh tanggapan (umpan balik) terkait pengembangan perangkat lunak ini di masa yang akan datang.

Sebagai model pengembangan perangkat lunak yang paling tua dan klasik, Waterfall memiliki karakteristik yang harus dipertimbangkan secara matang. Adapun kelebihan dan kekurangan dari Metode Waterfall, diantaranya :

- Kelebihan Metode Waterfall
 - Struktur yang Disiplin: Karena setiap tahap harus selesai sebelum pindah ke tahap berikutnya, model ini sangat terorganisir dan mudah dipantau kemajuannya.
 - Dokumentasi yang Kuat: Setiap fase menghasilkan dokumentasi yang lengkap, sehingga memudahkan proses pemeliharaan atau jika terjadi

- pergantian tim pengembang di tengah jalan.
- Meminimalisir Ketidakpastian: Karena kebutuhan didefinisikan secara kaku di awal (tahap Communication), potensi perubahan ruang lingkup (scope creep) di tengah proyek dapat ditekan.
 - Sangat Cocok untuk Proyek Terdefinisi: Efektif digunakan untuk proyek yang kebutuhan atau teknologinya sudah sangat jelas dan tidak akan berubah dalam waktu singkat.
 - Kekurangan Metode Waterfall
 - Kurang Fleksibel: Sangat sulit untuk kembali ke tahap sebelumnya jika terjadi perubahan kebutuhan di tengah jalan tanpa mengulang proses dari awal.
 - Risiko Tinggi di Akhir: Masalah besar atau ketidaksesuaian desain seringkali baru terlihat pada tahap Construction atau Testing, di mana biaya untuk memperbaikinya sudah sangat mahal.
 - Waktu Tunggu yang Lama: Pengguna tidak bisa melihat hasil kerja (prototipe) sampai tahapan akhir selesai, sehingga ada risiko produk akhir ternyata tidak sesuai dengan ekspektasi pengguna.
 - Untuk proyek yang bersifat eksploratif atau kebutuhan penggunanya belum pasti, model ini cenderung terlalu kaku.

2.1.7. Algoritma A*

Algoritma A* (A – Star) merupakan metode pencarian rute terpendek yang termasuk dalam klasifikasi algoritma pencarian dengan informasi (*informed search algorithm*) atau pencarian terbaik pertama (*best-first-search*). Metode ini secara efektif menentukan jalur dengan biaya terendah dari titik awal menuju titik akhir di dalam sebuah graf, menggunakan fungsi heuristik untuk memperkirakan biaya yang tersisa menuju tujuan.

A* menilai setiap node dalam graf dengan mengombinasikan dua nilai:

1. $g(n)$ adalah biaya nyata dari titik awal sampai *node* saat ini (n).
2. $h(n)$ adalah estimasi biaya heuristik dari *node* saat ini (n) menuju tujuan.
3. $f(n)$ adalah total estimasi biaya terendah untuk mencapai tujuan melalui simpul (n).

Total perkiraan biaya untuk jalur melalui *node* n dinyatakan sebagai

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

Algoritma A* memilih *node* dengan nilai $f(n)$ paling rendah untuk dieksplorasi selanjutnya. Kinerja A* sangat bergantung pada fungsi heuristik. Heuristik yang efektif akan

mempercepat proses pencarian, sedangkan heuristik yang kurang baik dapat menyebabkan algoritma berfungsi serupa dengan pencarian buta.

Dalam praktiknya, Algoritma A* memiliki beberapa istilah dasar, antara lain *starting point*, simpul (*nodes*), A, *open list*, *closed list*, harga(*cost*), dan halangan(*unwalkable*).

1. *Starting point* adalah istilah untuk lokasi awal dari suatu objek.
2. A adalah *node* yang sedang diproses dalam algoritma pencarian jalur terpendek.
3. Simpul adalah kotak-kotak kecil yang mewakili area pencarian jalur.
4. Harga adalah nilai yang diperoleh dari penjumlahan semua nilai setiap simpul dalam jalur terpendek dari *starting point* menuju A, ditambah estimasi nilai dari suatu simpul ke simpul tujuan.
5. Simpul tujuan adalah simpul yang ingin dicapai.
6. Halangan adalah atribut yang menunjukkan bahwa sebuah simpul tidak bisa dilalui oleh A*[DAL18].

Proses pencarian A* bekerja dengan mengelola dua struktur data utama:

1. *Open List*: Antrean prioritas (*priority queue*) yang berisi simpul-simpul yang telah ditemukan tetapi belum dievaluasi sepenuhnya. Simpul dengan nilai $f(n)$ terkecil akan selalu berada di urutan teratas untuk diproses.
2. *Closed List*: Kumpulan simpul yang sudah dievaluasi secara permanen. Jika sebuah simpul masuk ke *closed list*, berarti jalur terbaik menuju simpul tersebut sudah ditemukan.

Algoritma ini terus melakukan iterasi dengan mengambil simpul terbaik dari *Open List*, memeriksa tetangganya, menghitung nilai $f(n)$ mereka, dan memindahkan simpul yang sudah diproses ke *Closed List* hingga simpul tujuan ditemukan atau *Open List* kosong (yang berarti tidak ada rute).

Dalam mekanisme eksplorasi simpul, Algoritma A* menerapkan teknik ekspansi tetangga (*neighbor expansion*) yang sangat bergantung pada struktur ketetanggaan dalam graf atau grid. Ketika sebuah simpul sedang diproses (node A), algoritma akan mengidentifikasi seluruh simpul di sekitarnya yang tidak termasuk dalam kategori halangan (*unwalkable*) dan belum terdaftar dalam *Closed List*. Untuk setiap tetangga tersebut, algoritma melakukan kalkulasi ulang terhadap nilai $g(n)$ guna membandingkan apakah jalur yang baru ditemukan melalui node A lebih efisien daripada jalur yang mungkin sudah ditemukan sebelumnya. Jika jalur baru ini terbukti memiliki biaya yang lebih rendah, maka

data simpul tersebut di dalam *Open List* akan diperbarui, menjadikannya rute prioritas yang lebih diutamakan untuk mencapai tujuan.

Keunggulan utama A* dibandingkan algoritma pencarian lainnya terletak pada sifatnya yang *admissible* (layak) dan *consistent* (konsisten), asalkan fungsi heuristik $h(n)$ yang digunakan memenuhi syarat tertentu. Heuristik dikatakan *admissible* jika ia tidak pernah memberikan estimasi yang lebih besar dari biaya nyata yang sebenarnya menuju tujuan (*never overestimates*). Dengan menggunakan heuristik yang tepat, seperti *Euclidean Distance* untuk pergerakan bebas atau *Manhattan Distance* untuk pergerakan berbasis grid kotak, A* mampu memangkas ruang pencarian secara signifikan tanpa mengorbankan kualitas solusi. Hal ini menjadikan A* sangat optimal dalam memproses rute pada graf yang kompleks, karena algoritma tidak perlu mengeksplorasi seluruh simpul yang tersedia, melainkan hanya berfokus pada simpul-simpul yang secara matematis memiliki peluang terbesar menuju titik tujuan dengan jarak terkecil.

Berikut adalah gambaran logika pemrograman (*pseudocode*) dari algoritma A* untuk memberikan gambaran implementasi pada sistem.

```

Initialize Open List
Initialize Closed List
Add starting_node to Open List

while Open List is not empty:
    current_node = node in Open List with the lowest f(n)
    remove current_node from Open List
    add current_node to Closed List

    if current_node == goal_node:
        return reconstruct_path(current_node) // Jalur ditemukan

    for each neighbor of current_node:
        if neighbor is in Closed List or neighbor is unwalkable:
            continue

        new_g_cost = current_node.g + distance(current_node, neighbor)

        if neighbor is not in Open List or new_g_cost < neighbor.g:
            neighbor.g = new_g_cost
            neighbor.h = heuristic(neighbor, goal_node)
            neighbor.f = neighbor.g + neighbor.h
            neighbor.parent = current_node

            if neighbor is not in Open List:
                add neighbor to Open List

```

Adapun kelebihan dari Algoritma A-Star diantaranya:

- **Optimalitas:** Jika fungsi heuristik bersifat *admissible* (tidak pernah melebihi-lebihkan biaya sebenarnya), maka A* dijamin akan menemukan jalur terpendek.
- **Efisiensi Tinggi:** Dibandingkan algoritma Dijkstra, A* jauh lebih cepat karena "terbimbing" oleh heuristik menuju arah tujuan, sehingga tidak perlu mengeksplorasi seluruh graf.
- **Fleksibilitas:** Dapat disesuaikan dengan berbagai jenis hambatan (*obstacles*) dan bobot rute (misalnya rute angkot yang macet diberi bobot $g(n)$ lebih tinggi).

Selain kelebihan yang dimiliki, A-Star juga memiliki beberapa kekurangan, diantaranya:

- **Penggunaan Memori:** Karena A* harus menyimpan semua simpul yang dibangkitkan di dalam memori (*Open dan Closed List*), algoritma ini bisa memakan banyak RAM jika area pencariannya sangat luas dan kompleks.
- **Ketergantungan Heuristik:** Jika fungsi $h(n)$ tidak tepat, performa algoritma bisa menurun drastis menjadi hanya secepat pencarian biasa (BFS).

2.1.8. Framework

Framework adalah sekumpulan fungsi yang membantu *programmer* atau pengembang aplikasi web. Fitur-fitur ini ada di dalam kerangka kerja seperti *plugin* dan konsep. Konsep yang dimaksud tersebut bertujuan untuk menciptakan sistem agar dapat berkembang dengan teratur dan terorganisir. Selain itu, juga menawarkan *library* dan *tools* lain yang diperlukan saat mengembangkan aplikasi web. [RON19].

Manfaat Penggunaan Framework Penggunaan framework dalam pengembangan sistem informasi memberikan beberapa keuntungan signifikan, antara lain:

1. **Efisiensi Waktu:** Pengembang tidak perlu menulis kode dari awal untuk fungsi-fungsi umum seperti validasi form, keamanan, atau koneksi database.
2. **Keamanan Terjamin:** Framework biasanya memiliki fitur keamanan bawaan untuk menangkal serangan umum seperti SQL Injection dan Cross-Site Scripting (XSS).
3. **Struktur Kode yang Rapi:** Framework memaksa pengembang mengikuti standar pengkodean tertentu, sehingga kode lebih mudah dibaca dan dipelihara (*maintainable*) oleh tim atau peneliti selanjutnya.
4. **Skalabilitas:** Memudahkan aplikasi untuk dikembangkan lebih lanjut di masa

depan seiring dengan bertambahnya data rute atau jumlah pengguna.

2.1.9. Laravel

Laravel merupakan salah satu framework PHP yang paling dikenal dan sering dipakai dalam pengembangan web. Dikembangkan oleh Taylor Otwell, Laravel ditujukan untuk mempermudah, mempercepat, dan meningkatkan pengalaman dalam pengembangan aplikasi. Dengan pendekatan yang elegan dan sintaks yang mudah dimengerti, Laravel menawarkan berbagai alat dan fitur yang kuat untuk mendukung pembuatan aplikasi, mulai dari yang sederhana hingga yang lebih rumit[AGU24].

Fitur utama laravel:

1. *Eloquent ORM (Object-Relational Mapping)*: Eloquent mempermudah hubungan dengan database dengan cara yang lebih mudah dipahami dan berorientasi objek.
2. *Migrations*: Memudahkan dalam mengatur struktur database aplikasi.
3. *Blade Template Engine*: Sistem templat yang efektif untuk mengelola tampilan aplikasi dengan lebih baik.
4. *Middleware*: Fitur yang berfungsi untuk menyaring permohonan HTTP yang masuk ke aplikasi.
5. *Routing yang sederhana*: Sistem routing yang dasar namun fleksibel.
6. *Autentikasi*: Sistem autentikasi yang komprehensif dan mudah untuk disesuaikan.
7. *Laravel Mix*: Alat untuk mengatur aset serta melakukan kompilasi pada CSS dan JavaScript. [AGU24].



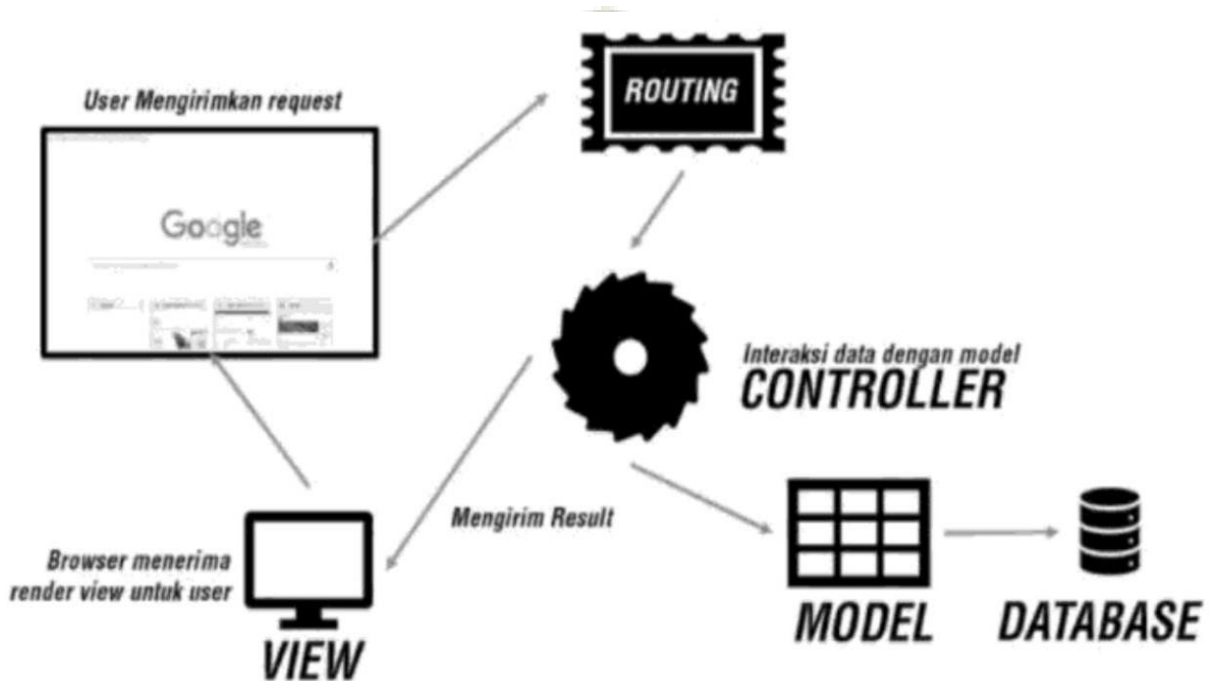
Gambar 2. 2 Laravel

Arsitektur aplikasi Laravel didesain agar mudah dipahami dan fleksibel, memberi kesempatan bagi pengembang untuk membuat aplikasi dari yang sederhana hingga yang rumit dengan efisiensi dan keamanan. Laravel menggunakan berbagai prinsip dan pola desain perangkat lunak, seperti *Model-View-Controller (MVC)*, *Dependency Injection*, serta

prinsip “*convention over configuration*”.

MVC merupakan sebuah struktur dalam pembuatan aplikasi web yang memisahkan aplikasi menjadi tiga komponen, di mana setiap komponen memiliki fungsi dan tanggung jawab masing-masing. Ketiga komponen ini adalah *model*, *view*, dan *controller*.

1. *Model*, bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisasikan data (dari database) sesuai dengan intruksi dari *controller*.
2. *View*, Bertugas untuk menyajikan informasi (yang mudah dimengerti) kepada user sesuai dengan intruksi dari *controller*.
3. *Controller*, bertugas untuk mengatur apa yang harus dilakukan *model* dan *view* mana yang harus ditampilkan berdasarkan permintaan dari *user* tidak selalu memerlukan aksi dari *model*. Misalnya seperti menampilkan halaman *form* untuk registrasi *user*[YUD19].



Gambar 2. 3 MVC (Model, View, Controller)

Selain fitur-fitur utama di atas, Laravel dikenal karena memiliki ekosistem yang luas dan dukungan komunitas yang besar. Hal ini memberikan jaminan keamanan melalui fitur proteksi bawaan terhadap serangan *Cross-Site Request Forgery* (CSRF) dan enkripsi data yang kuat. Bagi pengembang, Laravel juga menyediakan alat baris perintah bernama Artisan, yang berfungsi untuk mengotomatisasi tugas-tugas repetitif seperti pembuatan kontroler, model, dan pengelolaan database migration. Keunggulan ini sangat krusial dalam pembangunan aplikasi rute angkot, karena memungkinkan penulis untuk melakukan iterasi pengembangan sistem secara lebih cepat dan terstruktur.

Dalam praktiknya, alur kerja Laravel dimulai ketika seorang pengguna mengirimkan permintaan (*request*) melalui peramban web. Permintaan tersebut pertama kali diterima oleh *Routing*, yang kemudian mengarahkannya ke *Controller* yang sesuai. Jika permintaan membutuhkan data (seperti data koordinat rute angkot), *Controller* akan memanggil Model untuk mengambil data dari database. Setelah data diperoleh, *Controller* akan mengirimkan data tersebut ke *View* untuk disusun ke dalam templat HTML (menggunakan *Blade*) dan akhirnya ditampilkan kembali kepada pengguna dalam bentuk informasi yang visual dan interaktif.

2.1.10. Tailwind CSS



Gambar 2. 4 Tailwind CSS

Tailwind CSS merupakan *framework* CSS yang mengutamakan utilitas untuk menciptakan desain kustom dengan cepat. Daripada menggunakan komponen yang sudah ada, Tailwind memanfaatkan kelas-kelas bantuan untuk merancang desain langsung di dalam *markup*. Laravel mengadopsi Tailwind CSS sebagai standar di *Jetstream* dan *Breeze*, menunjukkan integrasi yang baik serta pilihan yang modern. Tailwind CSS ideal untuk aplikasi yang memerlukan desain kustom yang luas dan pengendalian penataan yang lebih rinci. [AGU24].

Tailwind CSS adalah *framework* CSS terbaru yang membantu para pengembang untuk membuat desain yang fleksibel dan seragam secara langsung di dalam HTML. Alih-alih bergantung pada pola desain atau komponen yang sudah ada, Tailwind CSS memberikan metode yang lebih mendetail, di mana pengembang bisa mengontrol sepenuhnya tampilan setiap elemen.

Fitur utama dari Tailwind CSS:

1. *Utility-First* CSS, menggunakan kelas-kelas utilitas kecil seperti `text-center`, `bg-blue-500`, atau `p-4` untuk memberikan gaya pada elemen.
2. Sangat bisa disesuaikan, para pengembang dapat dengan mudah merubah skema warna, ukuran dan pengaturan lainnya lewat file konfigurasi.
3. Desain responsif, tailwind sangat memudahkan pembuatan desain responsif dengan *breakpoint* bawaan seperti `sm`, `md`, `lg`, dan `xl`.

4. Desain atomik, setiap kelas utilitas hanya mengubah satu aspek tertentu dari elemen, sehingga lebih fleksibel untuk merancang desain yang rumit[BON25].

Selain keunggulan dalam aspek utilitas dan fleksibilitas desain, Tailwind CSS juga menawarkan efisiensi yang signifikan melalui proses optimasi yang disebut dengan *Purging*. Dalam alur kerja pengembangan, Tailwind akan memindai seluruh file HTML, komponen, dan skrip untuk mengidentifikasi kelas-kelas yang benar-benar digunakan, kemudian secara otomatis menghapus kelas-kelas yang tidak terpakai dari file CSS akhir. Hal ini menghasilkan ukuran file CSS yang sangat kecil dan ringan, sehingga mempercepat waktu pemuatan halaman (*page load*) secara drastis dibandingkan dengan framework CSS tradisional yang menyertakan seluruh komponen dalam satu paket.

Integrasi Tailwind CSS dalam ekosistem modern, khususnya pada pengembangan berbasis komponen seperti Vue.js, React, atau modul *blade* pada Laravel, memperkuat konsep reusabilitas. Meskipun desain dilakukan langsung di dalam markup menggunakan kelas utilitas, pengembang tetap dapat mempertahankan konsistensi dengan cara mengekstraksi pola yang berulang menjadi komponen-komponen kecil. Pendekatan ini selaras dengan metodologi *Atomic Design*, di mana elemen-elemen atomik (kelas utilitas) disusun menjadi molekul atau organisme (komponen), memberikan kendali penuh kepada pengembang untuk membangun antarmuka yang kompleks tanpa kehilangan kendali atas arsitektur kode.

2.1.11. Graph

Graph adalah suatu representasi persoalan dalam bentuk peta. *Graph* tersusun atas beberapa komponen, yaitu kumpulan *vertices (nodes)* dan kumpulan *edges (sisi)*. *Graph* secara garis besar terbagi atas dua macam :

1. *Connected Graph*

Connected Graph adalah suatu *graph* yang setiap *nodes* terhubung satu sama lain dengan suatu *edge*.

2. *Disconnected Graph*

Disconnected Graph adalah suatu *graph* yang tidak setiap *nodes* terhubung satu sama lain dengan suatu *edge*. Dengan kata lain, *graph* ini terpotong.

Untuk representasi *graph* dalam bahasa pemrograman, ada dua macam bentuk representasi, yaitu :

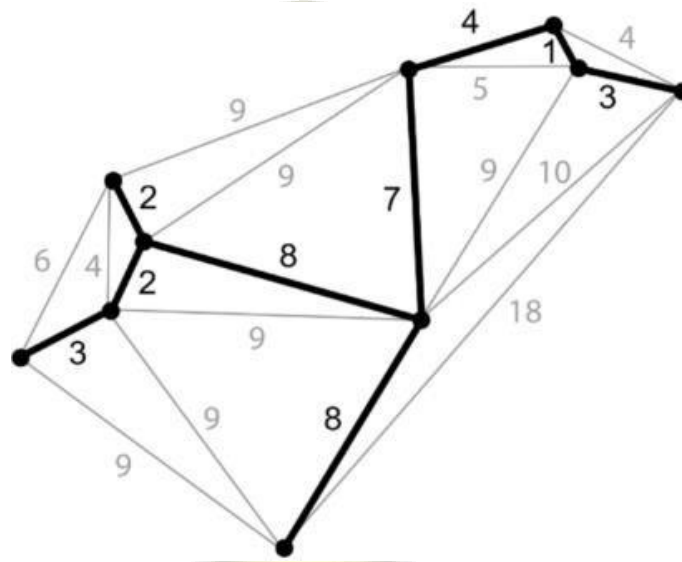
1. *Adjacency Matrix*

Adjacency Matrix adalah suatu representasi *graph* dalam bentuk matrix 2 dimensi. Sangat efisien untuk *Dense Graph* (*graph* yang hubungan antar node-

nya sangat padat). Kelebihannya adalah kecepatan dalam memeriksa apakah dua node terhubung (hanya memerlukan waktu ($O(1)$), namun memakan memori yang besar (V^2) karena harus menyimpan sel kosong untuk node yang tidak terhubung.

2. *Adjacency List*

Adjacency List adalah suatu representasi *graph* dalam bentuk *list*[CHA23]. Sangat efisien untuk *Sparse Graph* (*graph* yang hubungan antar node-nya sedikit/jarang), yang biasanya ditemukan pada peta jalan raya. Representasi ini lebih hemat memori karena hanya menyimpan node yang benar-benar bertetangga. Dalam implementasi A*, *Adjacency List* lebih sering digunakan karena mempercepat proses iterasi tetangga (*neighboring nodes*).



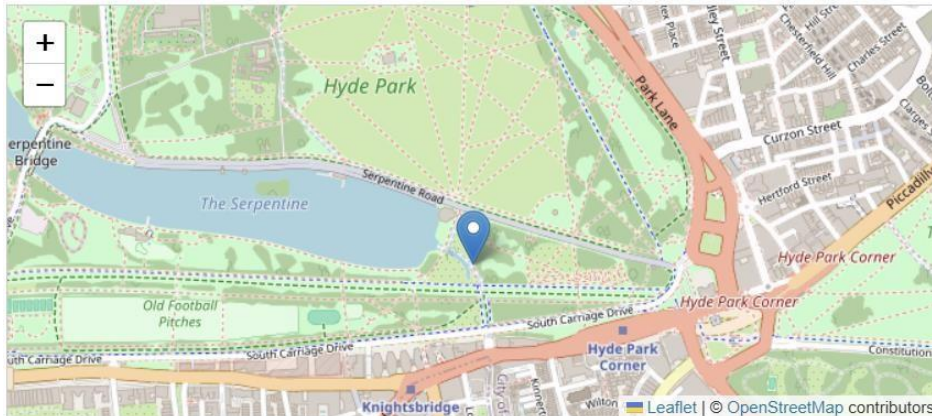
Gambar 2. 5 Graph

2.1.12. LeafletJs

Leaflet merupakan pustaka JavaScript terkemuka yang bersifat sumber terbuka untuk menciptakan peta interaktif yang ramah pada perangkat mobile. Dengan ukuran sekitar 38 KB, pustaka ini mencakup semua fitur pemetaan yang diperlukan oleh sebagian besar pengembang. Keunggulan dari *open source* membuatnya lebih mudah untuk dikembangkan oleh peneliti selanjutnya dan memudahkan adaptasi teknologi baru dalam GIS.

Dalam penerapannya, Sistem Informasi Geografis (SIG) memerlukan data spasial, yaitu informasi yang merujuk pada lokasi suatu objek dalam bentuk koordinat di permukaan bumi. GIS adalah sistem yang dirancang untuk mengumpulkan, menyimpan, memperbarui, memanipulasi, menganalisis, dan menunjukkan semua bentuk informasi yang berhubungan

dengan geografi. Dengan menggunakan Leaflet, data spasial seperti bangunan dan ruangan dalam format geoJson dapat disimpan di server tanpa perlu terhubung ke internet, hanya dengan menggunakan intranet. Untuk mengakses data tersebut, *plugin jQuery* dan *Bootstrap* digunakan untuk menampilkan peta di halaman web. Salah satu kelebihan menggunakan Leaflet adalah bahwa ia menawarkan fungsionalitas untuk menambahkan penanda, *pop-up*, garis *overlay*, dan bentuk menggunakan lapisan, *zoom*, serta pan, meskipun ini hanya merupakan beberapa fitur dari Leaflet.



Gambar 2. 6 Tampilan Peta Leaflet

LeafletJs adalah sekumpulan fungsi berbasis JavaScript yang digunakan untuk menampilkan peta interaktif di halaman web. Leaflet menawarkan Map API (*Application Programming Interface*) yang memudahkan pengembang web untuk menampilkan peta berbasis *Tile* di halaman mereka. Pengguna peta juga dapat berinteraksi menggunakan fungsi yang disediakan oleh Leaflet. Seperti API *webmap* lainnya, Leaflet memiliki beberapa komponen dasar sebagai berikut:

3. Peta adalah komponen utama yang memuat berbagai komponen lainnya. Bayangkan peta sebagai latar kosong yang nantinya akan diisi dengan komponen lain (seperti *tilelayer*, *marker*, dan sebagainya). Pada komponen ini, ukuran peta pada halaman web (melalui fungsi CSS, lebar dan tinggi), koordinat pusat peta (dalam *latitude* dan *longitude*), serta level *zoom* awal (antara 0-20, di mana level 20 menunjukkan perbesaran maksimum) didefinisikan. Komponen-komponen dapat ditambahkan pada peta menggunakan metode *.addTo(Map)*
4. *Tilelayer* (sering disebut *slippy map*), adalah komponen yang menyediakan latar belakang peta untuk *webmap* atau peta online. Peta latar yang tersedia ditampilkan dalam bentuk kotak-kotak (*tile*) dengan tampilan berbeda pada berbagai level *zoom*.
5. *Marker* adalah simbol sederhana yang digunakan untuk menunjukkan titik.

Simbol *default* untuk marker di Leaflet, walaupun LeafletJs dan API Peta lainnya memungkinkan pengembang untuk mengganti simbol tersebut dengan bebas.

6. *Popup* adalah jendela kecil yang berisi informasi terkait dengan *marker* tertentu. *Popup* biasanya digunakan untuk menunjukkan informasi tentang titik itu, seperti teks, gambar, atau grafik.
7. *Event* adalah kejadian yang dapat diamati oleh Leaflet pada peta. Fungsi *event* digunakan untuk menyediakan interaktifitas dengan pengguna[ROL19].

2.2. Penelitian Terdahulu

Penelitian terdahulu merupakan analisis atau tinjauan terhadap studi-studi yang telah dikerjakan oleh peneliti lain yang memiliki hubungan dengan topik yang sedang diteliti. Penelitian terdahulu tidak hanya berfungsi sebagai daftar referensi, tetapi memiliki peran strategis dalam membangun kerangka berpikir yang kokoh. Tinjauan ini bertujuan untuk menghindari plagiarisme dan duplikasi, mengidentifikasi celah penelitian (*research gap*), dan juga sebagai landasan pemilihan metode.

Berikut ini dicantumkan dalam tabel di bawah beberapa penelitian yang dijadikan referensi dalam laporan penelitian ini.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Alasan Digunakan
1	Rahmat Kurniawan R, Abdul Halim Hasugian, Putri Hanifah	Penentuan Rute dan Tarif Perjalanan Angkutan Umum di Kota Medan Menggunakan Algoritma A*	Karena secara langsung berfokus pada penentuan jalur dan biaya pada angkutan umum (angkot) dengan memakai algoritma A*. Penelitian ini menguji ide penggunaan A* untuk peningkatan jalur angkot. Penelitian ini berfungsi sebagai pedoman dalam memodelkan jaringan angkot (simpul, sisi, bobot) dan menerapkan Algoritma A* untuk menemukan jalur terbaik, yang selanjutnya menjadi landasan bagi desain sistem.
2	Yusra Fernando, Muhammad Ativ Mustaqov, Dyah Ayu Megawaty	Penerapan Algoritma A-Star pada aplikasi Pencarian Lokasi Fotografi di Bandar Lampung Berbasis Android.	Studi ini menekankan keberhasilan algoritma A* dalam menyelesaikan masalah pencarian lokasi dan jalur terpendek, tanpa menghiraukan konteks khususnya (fotografi). Walaupun platform yang digunakan adalah android, bukan web, penelitian ini memberikan pemahaman tentang cara data spesial diproses dan bagaimana penerapan fungsi heuristik A* dalam aplikasi yang sebenarnya.
3	Hendri Yusriadi, Harun Mukhtar, Soni	Implementasi Algoritma A Star dalam Pencarian Rute Terpendek (Shortest Path Problem) Pada Sistem Pencarian Kantor Pos di K	Penelitian ini menunjukkan bahwa Algoritma A* berhasil diterapkan dalam sistem informasi geografis untuk menyelesaikan masalah jalur terpendek. Pencarian lokasi kantor pos dan penentuan rute angkot

		ota Pekanbaru	<p>melibatkan penentuan jalur terbaik di antara lokasi-lokasi geografis. Penelitian ini memberikan gambaran mengenai arsitektur sistem yang menggunakan A* untuk kebutuhan navigasi serta bagaimana cara memodelkan data geografi ke dalam struktur graf yang dapat diproses oleh algoritma, menjadi acuan dalam tahap perancangan sistem.</p>
4	Arief Bramato Wicaksono Putra, Achmad Aulia Rachman, Adji Santoso, Mulyanto	Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda menggunakan Algoritma A*(star) dan Floyd-Warshall	<p>Studi ini penting untuk mengetahui keunggulan dan keterbatasan Algoritma A* dengan membandingkannya dengan algoritma Floyd-Warshall. Penelitian ini berfungsi sebagai panduan untuk memahami alasan mengapa A* lebih tepat dipilih untuk situasi rute angkot. Contohnya, A* sangat efisien dalam mencari jalur dari satu lokasi ke lokasi lain (jalur terpendek dari sumber tunggal) pada graf yang besar, yang cocok dengan kebutuhan pengguna yang ingin menemukan rute dari titik awal hingga tujuan tertentu.</p>
5	Arif cahyo Prasetyo, Maful Prayoga Armandi, Harish Setyo Hudhanto, Bayu Setiaji	Perbandingan Algoritma Astar dan Dijkstra dalam Menentukan Rute Terdekat	<p>Studi ini memberikan perbandingan antara Algoritma A* dan Dijkstra. Ini berfungsi sebagai panduan untuk menjelaskan kelebihan A* khususnya dalam aspek efisiensi pencarian pada grafik yang besar karena penerapan fungsi heuristik.</p>

BAB 3

SKEMA PENELITIAN

3.1. Alur Penyelesaian Tugas Akhir

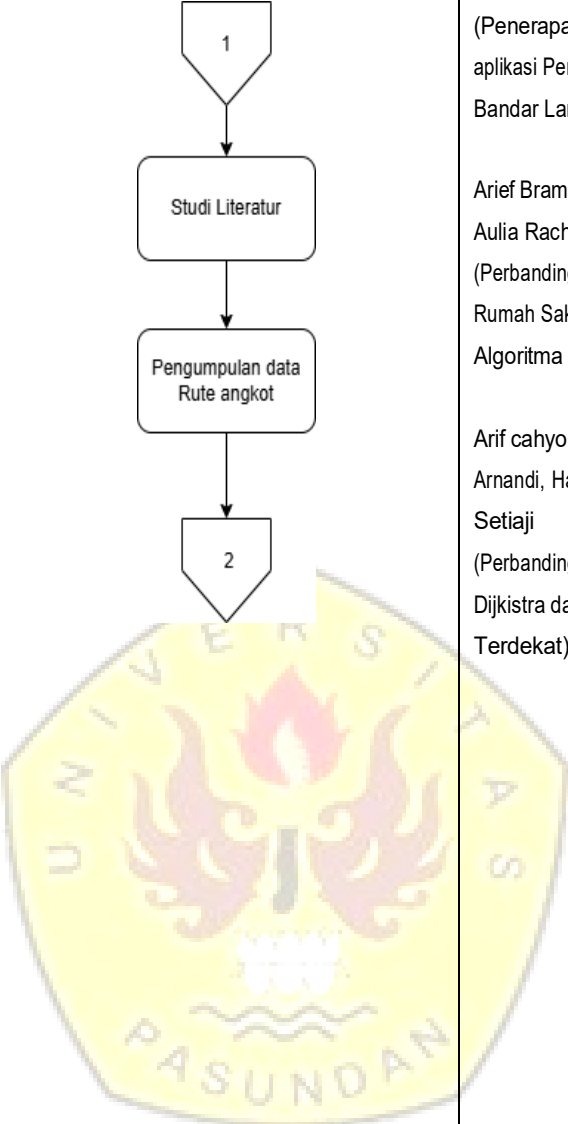
Alur penelitian merupakan serangkaian langkah-langkah teratur yang dijalani oleh seorang peneliti saat menjalankan penelitian, dimulai dari menemukan masalah sampai penyusunan laporan hasil penelitian.

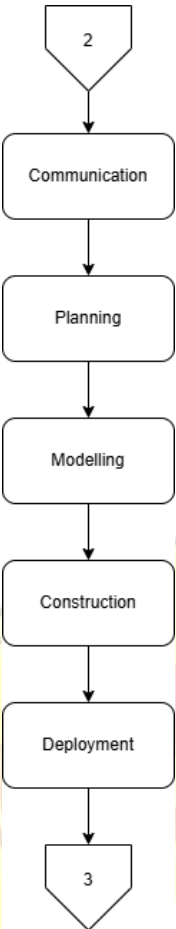
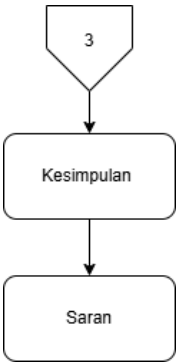
Proses penelitian ini dirancang untuk menjembatani kesenjangan antara permasalahan transportasi yang ada di lapangan dengan solusi berbasis teknologi informasi yang dikembangkan. Secara garis besar, alur ini mencakup fase identifikasi kebutuhan, studi literatur untuk memperkuat landasan teori, hingga proses perancangan sistem menggunakan algoritma A*. Dengan mengikuti alur yang terstruktur, setiap hambatan teknis yang muncul selama proses pengerjaan dapat diidentifikasi dan diselesaikan secara metodelis.

Berikut ini adalah langkah-langkah dalam proses penelitian yang menjadi dasar dalam menyelesaikan tugas akhir. Rincian lebih lanjut mengenai proses penelitian dapat dilihat pada tabel 3.1.

Tabel 3. 1 Alur Penyelesaian Tugas Akhir

No	Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
1.	<p>Tahap 1 :</p> <p>Mengidentifikasi masalah Tugas Akhir</p> <p>Hasil :</p> <p>Masalah teridentifikasi setelah melalui analisis pengumpulan data</p> <p>Kontribusi :</p> <p>Memastikan masalah pada tugas akhir ini lebih terarah dan tidak menyimpang dari tujuan</p>	<pre> graph TD A[Identifikasi Masalah] --> B{1} </pre>	<p>Rahmat Kurniawan R, Abdul Halim Hasugian, Putri Hanifah (Penentuan Rute dan Tarif Perjalanan Angkutan Umum di Kota Medan Menggunakan Algoritma A*)</p> <p>Hendri Yusriadi, Harun Mukhtar, Soni (Implementasi Algoritma A Star dalam Pencarian Rute Terpendek (Shortest Path Problem) Pada Sistem Pencarian Kantor Pos di Kota Pekanbaru)</p> <p>Website BPS KOTA BANDUNG untuk mengambil data trayek angkot resmi.</p>

No	Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
2	<p>Tahap 2:</p> <p>Studi literatur dengan menganalisis jurnal – jurnal yang relevan dengan masalah Tugas akhir yang di ambil.</p> <p>Pengumpulan data yaitu meminta data rute angkot pada badan yang menangani</p> <p>Hasil :</p> <p>Mendapatkan pemahaman lebih mengenai topik yang di ambil</p> <p>Mendapatkan rute angkot yang akurat</p> <p>Kontribusi :</p> <p>Menyediakan dasar pengetahuan atau pemahaman dari jurnal yang relevan dengan topik yang di ambil</p> <p>Memberikan data yang diperlukan untuk menganalisis rute rute angkot</p>	 <pre> graph TD 1{{1}} --> A[Studi Literatur] A --> B[Pengumpulan data Rute angkot] B --> 2{{2}} </pre>	<p>Yusra Fernando, Muhammad Ativ Mustaqov, Dyah Ayu Megawaty (Penerapan Algoritma A-Star pada aplikasi Pencarian Lokasi Fotografi di Bandar Lampung Berbasis Android)</p> <p>Arief Bramato Wicaksono Putra, Achmad Aulia Rachman, Adji Santoso, Mulyanto (Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda menggunakan Algoritma A*(star) dan Floyd-Warshall)</p> <p>Arif cahyo Prasetyo, Maful Prayoga Arnandi, Harish Setyo Hudnanto, Bayu Setiaji (Perbandingan Algoritma Astar dan Dijkstra dalam Menentukan Rute Terdekat)</p>

No	Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
3	<p>Tahap 3 :</p> <p>Pembangunan Perangkat Lunak dengan acuan metode waterfall</p> <p>Hasilnya :</p> <p>Membuat website rute pencarian angkot yang memiliki informasi gambar, warna, dan tarif angkot dengan menggunakan metode waterfall</p> <p>Kontribusi :</p> <p>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[Modelling] D --> E[Construction] E --> F[Deployment] F --> G{{3}} </pre>	<p>Rahmat Kurniawan R, Abdul Halim Hasugian, Putri Hanifah (Penentuan Rute dan Tarif Perjalanan Angkutan Umum di Kota Medan Menggunakan Algoritma A*)</p> <p>Hendri Yusriadi, Harun Mukhtar, Soni (Implementasi Algoritma A Star dalam Pencarian Rute Terpendek (Shortest Path Problem) Pada Sistem Pencarian Kantor Pos di Kota Pekanbaru)</p>
4	<p>Tahap 4 :</p> <p>Kesimpulan dan saran</p> <p>Hasil:</p> <p>Memberikan kesimpulan dan saran pada tugas akhir</p> <p>Kontribusi :</p> <p>Memberikan kesimpulan dan saran pada tugas akhir</p>	 <pre> graph TD A{{3}} --> B[Kesimpulan] B --> C[Saran] </pre>	<p>Arief Bramato Wicaksono Putra, Achmad Aulia Rachman, Adji Santoso, Mulyanto (Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda menggunakan Algoritma A*(star) dan Floyd-Warshall)</p> <p>Arif cahyo Prasetyo, Maful Prayoga Arnandi, Harish Setyo Hudnanto, Bayu Setiaji (Perbandingan Algoritma Astar dan Dijkstra dalam Menentukan Rute Terdekat)</p>

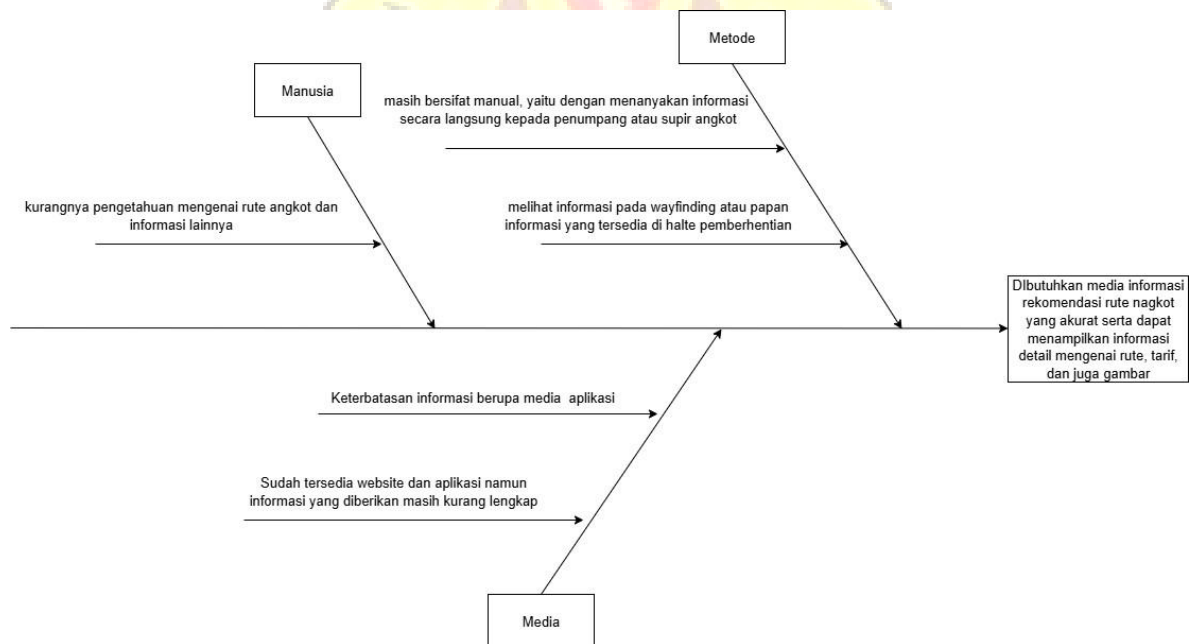
3.2. Perumusan Masalah

Penyusunan masalah umumnya dibagi menjadi dua kategori besar. Kategori pertama menitikberatkan pada identifikasi efek atau hasil dari suatu masalah serta analisis faktor-faktor yang menyebabkannya. Sedangkan kategori kedua mencari berbagai pendekatan yang dapat diterapkan, berdasarkan pada penyebab masalah yang telah diidentifikasi di tahap awal.

Pernyataan permasalahan yang terdapat dalam bagian ini dapat disampaikan dengan berbagai bentuk representasi formal, seperti diagram penyebab dan akibat atau diagram akar permasalahan. Selain itu, penyampaian dalam format narasi atau tabel juga dapat dilakukan.

3.2.1. Analisis Sebab Akibat

Analisis sebab dan akibat adalah langkah terstruktur untuk menemukan, memahami, dan menggambarkan keterkaitan antara suatu masalah (efek) dengan berbagai elemen yang berperan atau menyebabkannya (sebab) yang diilustrasikan melalui fishbone diagram. Berikut ini adalah analisis sebab akibat yang ditemukan dalam penelitian akhir:



Gambar 3. 1 Fishbone Diagram Sebab Akibat

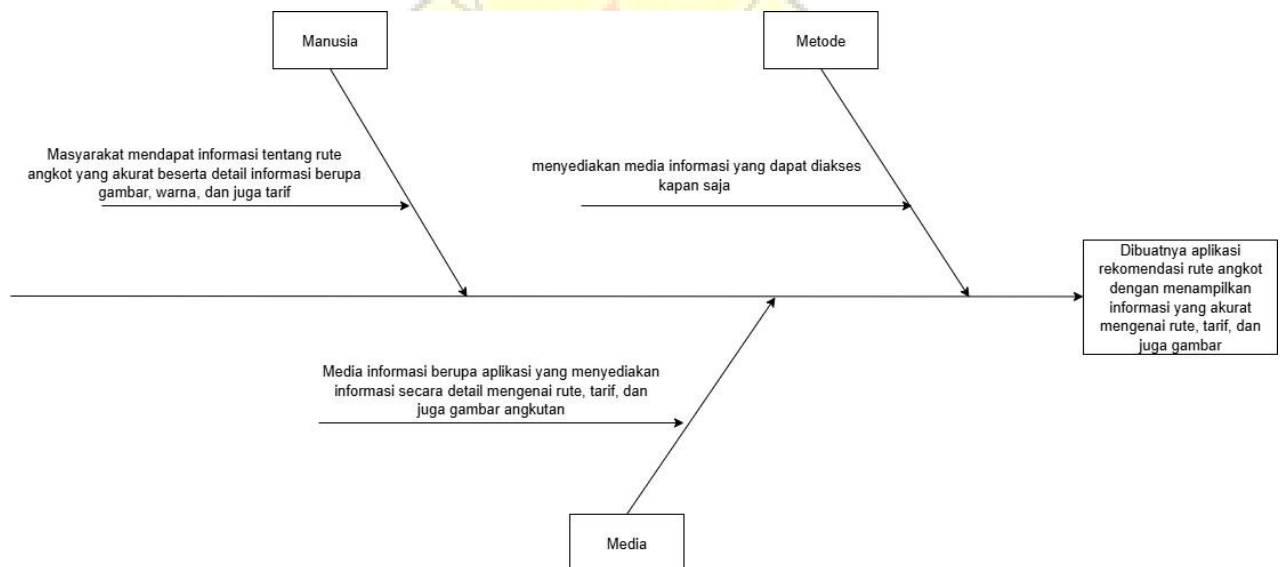
Berdasarkan hubungan sebab akibat yang terlihat pada ilustrasi di atas, berikut adalah penjelasan mengenai penyebab isu yang muncul dalam tugas akhir.

1. Manusia: Penyebab masalah ini pada individu adalah minimnya pengetahuan atau informasi tentang rute angkot dan informasi lain seperti gambar per rute, biaya tarif, dan warna angkot yang harus digunakan.

2. Media: Penyebabnya adalah masih kurangnya informasi yang komprehensif mengenai rute angkot. Meskipun telah ada media seperti situs web dan aplikasi, informasi yang disampaikan masih tidak memadai.
3. Metode: Cara untuk mendapatkan informasi tentang rute angkot hingga saat ini masih bersifat manual, yaitu dengan menanyakan kepada orang lain atau sopir angkot, serta melihat informasi pada petunjuk arah atau papan informasi yang tersedia di halte.

3.2.2. Analisis Solusi Masalah

Analisis solusi masalah merupakan tahap terstruktur untuk menemukan, menilai, dan memilih opsi yang paling efisien dan praktis guna menangani isu yang telah diidentifikasi serta diteliti penyebabnya. Langkah ini memerlukan pemikiran kritis, penilaian berbagai opsi dan mempertimbangkan berbagai elemen sebelum keputusan akhir diambil dan digambarkan dengan menggunakan diagram tulang ikan. Berikut adalah analisis solusi masalah yang muncul dalam penelitian tugas akhir:



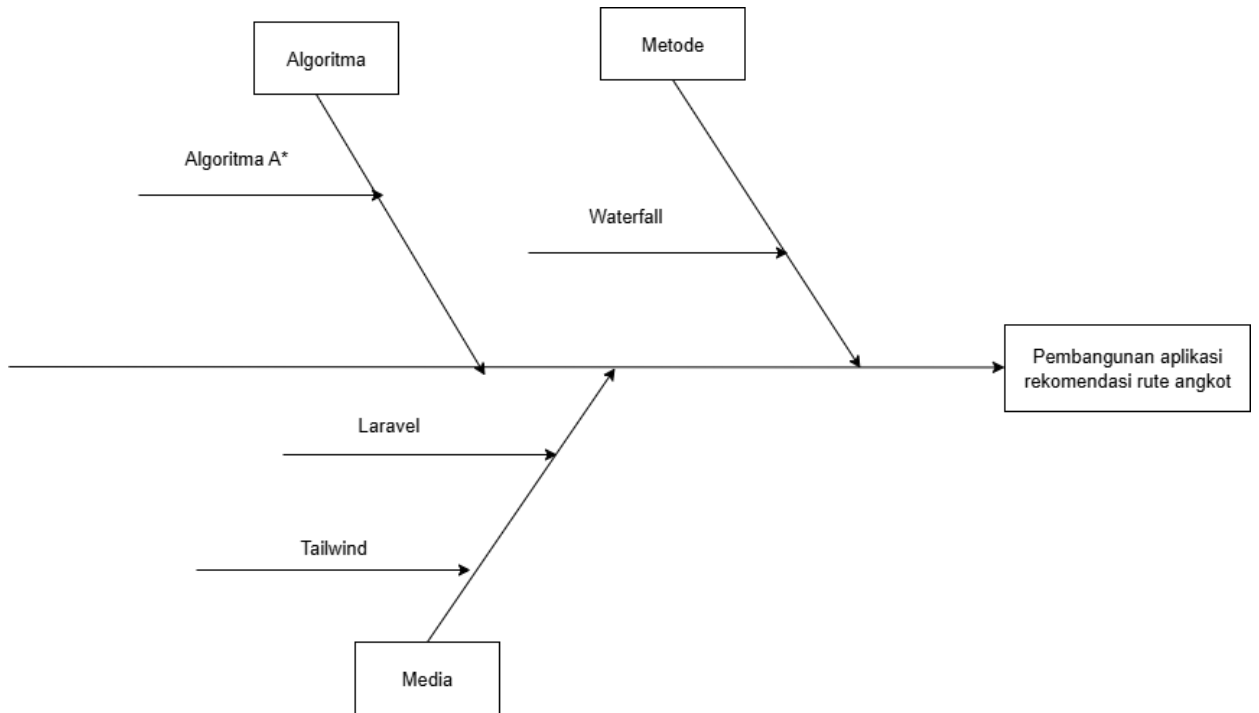
Gambar 3. 2 Fishbone Diagram Analisis Solusi Masalah

Berdasarkan analisis solusi masalah yang terdapat dalam gambar di atas, berikut ini adalah penjelasan mengenai faktor-faktor yang menyebabkan masalah pada proyek akhir.

1. Manusia: Masyarakat memperoleh informasi yang tepat mengenai rute angkutan umum, termasuk gambar, warna, dan tarif angkot.
2. Media: Mengembangkan aplikasi yang lebih informatif dalam memberikan detail tentang transportasi kota di Bandung.
3. Metode: Menyediakan platform yang dapat diakses kapan saja dan di mana saja dalam bentuk situs web.

3.2.3. Kerangka Berpikir Teoritis

Kerangka Berpikir Teoritis merupakan suatu rangka konsep yang diterapkan dalam penelitian untuk mengatur, menerangkan, dan memperkirakan hubungan antara berbagai variabel atau ide yang berkaitan dengan persoalan penelitian. Berikut adalah ilustrasi *fishbone* diagram dari kerangka berpikir teoritis.



Gambar 3. 3 Fishbone Diagram Kerangka Pemikiran Teoritis

BAB 4

ANALISIS DAN PERANCANGAN

4.1. Analisis Perangkat Lunak

Analisis perangkat lunak merupakan kegiatan yang mendalam untuk mengerti secara mendetail tantangan yang ingin diselesaikan oleh perangkat lunak dan mengidentifikasi kebutuhan serta batasan yang harus diperhatikan. Tujuan dari analisis perangkat lunak adalah untuk memperoleh kebutuhan sistem dan kemudian membuat modelnya dengan diagram. Proses ini sangat penting untuk melanjutkan tahap pengembangan perangkat lunak ke tahap berikutnya, yaitu tahap perancangan.

Dalam proses ini, penulis melakukan analisis kebutuhan fungsional (seperti fitur pencarian dan visualisasi rute). Hasil dari analisis ini menjadi fondasi utama dalam penyusunan model sistem melalui diagram-diagram teknis, yang memastikan bahwa setiap baris kode pada tahap konstruksi nanti memiliki tujuan yang jelas dalam memecahkan masalah aksesibilitas informasi angkutan kota.

4.1.1. Algoritma Pencarian Rute (Algoritma A*)

Untuk menentukan jalur terbaik, sistem ini memanfaatkan Algoritma pencarian A*. Algoritma ini dipilih karena keefektifannya dalam mencari rute terpendek dengan biaya terendah melalui kombinasi antara biaya perjalanan yang telah dilalui ($g(n)$) dan perkiraan jarak menuju tujuan ($h(n)$). Agar algoritma dapat berfungsi secara maksimal, maka data rute yang lengkap dan teratur sangat diperlukan.

4.1.1.1. Simulasi Algoritma A*

Simulasi ini bertujuan untuk memvalidasi logika algoritma sebelum diimplementasikan sepenuhnya ke dalam kode program Laravel. Melalui simulasi ini, penulis dapat memantau pergerakan algoritma dalam mengeksplorasi setiap node (simpul) rute dan bagaimana nilai $f(n)$ menentukan keputusan jalur terbaik.

Untuk memberikan gambaran yang nyata, penulis menggunakan data sekunder dari Badan Pusat Statistik (BPS) Kota Bandung yang bersumber dari Dinas Perhubungan Kota Bandung sebagai parameter input. Data ini mencakup lintasan trayek, titik simpul persimpangan antar angkot, serta terminal-terminal utama. Dalam simulasi ini, diasumsikan pengguna ingin menuju ke satu lokasi tujuan dengan melewati beberapa pilihan trayek yang bersinggungan. Penulis akan membandingkan beberapa rute alternatif berdasarkan total bobot $f(n)$ untuk membuktikan bahwa jalur yang direkomendasikan oleh sistem adalah jalur dengan efisiensi jarak dan waktu yang paling optimal bagi penumpang

Kode	Lintasan Trayek	Jarak (km)	Jumlah Armada
1.A	Abdul Muis -Cioaheum Via Binong	16,3	355
1.B	Abdul Muis-Cioaheum Via Aoeh	11,55	100
2	Abdul Muis-Dago	9,3	271
3	Abdul Muis-Ledeng	16	245
4	Abdul Muis-Elang	9,75	101
5	Cioaheum-Ledeng	14,25	214
6	Cioaheum-Ciroyom	17	206
7	Cioaheum-Ciwastra-Derwati	17	200
8	Cioaheum-Cibaduyut	16,1	150
9	Stasiun Hall-Dago	10	52
10	Stasiun Hall-Sadang Serang	11	150
11.A	Stasiun Hall-Ciumbuleuit-Via Eykman	9,8	53
11.B	Stasiun Hall-Ciumbuleuit-Via Cihampelas	8,3	30
12	Stasiun Hall-Gede Bage	21	200
13	Stasiun Hall-Sarijadi	10,2	80
14	Stasiun Hall-Gunung Batu	8,5	53
15	Margahayu Raya-Ledeng	19,8	125
16	Dago-Riung Bandung	20,6	201
17	Pasar Induk Caringin-Dago	19,85	140
18	Panghegar Permai-Dipati Ukur-Dago	19,35	155
19.A	Ciroyom-Sarijadi Via Sukajdi	11,75	88
19.B	Ciroyom-Sarijadi Via Setrasari	10,75	31
20	Ciroyom-Bumi Asri	8,35	115
21	Ciroyom-Cikudapateuh	12,9	140
22	Sederhana-Cipagalo	16,05	276
23	Sederhana-Cijerah	8,9	63
24	Sederhana-Cimindi	9	55
25	Ciwastra-Ujung Berung	13,4	32
26	Cisitu-Tegallega	13,95	82
27	Cijerah-Ciwastra -Derwati	22,3	200
28	Elang-Gede Bage-Ujung Berung	22,45	115
29	Abdul Muis-Mengger	10,55	25
30	Cicadas-Elang	18,05	300
31	Antapani-Ciroyom	13,7	160
32	Cicadas-Cibiru-Pangjileukan	13,65	200
33	Bumi Pangjileukan-Sekemirung	24,35	125
34	Sadang Serang-Caringin	18,1	200
35	Cbaduyut-Karang Setra	16,6	201
36	Cibogo - Elang	7	32

Sumber: Dinas Perhubungan Kota Bandung

Gambar 4. 1 Data Trayek Berdasarkan BPS Kota Bandung

Tabel 4. 1 Data Rute Angkot

Kode	Lintasan Trayek	Jalur
1A	Abdul Muis – 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
1B	Abdul Muis – 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
2	Abdul Muis – 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. Ganेशha, Dago) – Simpang Dago – Terminal Dago
3	Abdul Muis – 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. Cipanganti – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Setiabudi – Terminal Ledeng
5	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. Cipanganti – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Setiabudi – Terminal Ledeng
6	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
7	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
8	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
9	Stasiun Hall – Dago	Jl. Stasiun Barat (Stasiun Bandung) – Jl. Stasiun Timur – Viaduct – Jl. Perintis Kemerdekaan – Jl. Wastu Kencana – Jl. RE. Martadinata – Jl. Ir. H. Juanda – RS Boromeus (Dago) – ITB (Jl. Ganेशha, Dago) – Simpang Dago – Terminal Dago

Kode	Lintasan Trayek	Jalur
10	Stasiun Hall – 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
11.A	Stasiun Hall – 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
11.B	Stasiun Hall – Ciumbuleuit Via Cihampelas	Stasiun Hall – Jalan Kebon Kawung – Jalan Pasir Kaliki – Jalan Pajajaran – Jalan Wastu Kencana – Jalan Cihampelas – Jalan Cipaganti – Jalan Dr. Setiabudi – Jalan Ciumbuleuit
12	Stasiun Hall – Gede Bage	Terminal Stasiun – Jl. Dulatip – Pasar Baru – Jl. Otto Iskandardinata (Otista) – Jl. Kepatihan – Jl. Dewi Sartika – Jl. Dalem Kaum – Alun-Alun (Asia Afrika) – Jl. Banceuy – Jl. ABC – Jl. Naripan – Jl. Sunda – Jl. Veteran – Jl. Ahmad Yani – Jl. Gatot Subroto – Jl. Burangrang – Jl. Halimun – Jl. Malabar – Jl. Talaga Bodas – Jl. Pelajar Pejuang – Jl. Martanegara – Jl. Reog – Jl. Karawitan – Jl. Kliningan – Jl. Buah Batu – Jl. Sukarno-Hatta – Margahayu Raya (Sukarno-Hatta) – Metro (Sukarno-Hatta) – Riung Bandung (Sukarno-Hatta) – Pasar Induk Gede Bage
13	Stasiun Hall – Sarijadi	Terminal Stasiun – Jl. Suniaraja – Jl. Otista – Jl. Stasiun Timur – Viaduct – Jl. Kebon Jukut – Jl. Kebon Kawung – Stasiun Bandung (Kebon Kawun) – Jl. Pasir Kaliki – Istana Plaza (Pasir Kaliki) – Jl. DR. Junjungan (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 Hall – 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 – BTC (Pasteur) – Jl. Westhoff – Jl. DR. Junjungan (Terusan Pasteur) – Jl. Gunung Batu – Terminal Gunung Batu
15	Margahayu Raya – Ledeng	Terminal Margahayu – Jl. Ranca Bolang (Margahayu Raya) – Jl. Sukarno-Hatta – Jl. Kiara Condong – Jl. Jakarta – Jl. WR. Supratman – Jl. Cendana – Jl. Taman Pramuka – Jl. RE. Martadinata – Jl. Wastu Kencana – Jl. Pajajaran – Jl. Cicendo – Jl. Rivai – Jl. Cipaganti – Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Setiabudi – Terminal Ledeng

Kode	Lintasan Trayek	Jalur
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	Pasar Induk Caringin – Dago	Pasar Induk Caringin – Jl. Babakan Ciparay – Jl. Sukarno-Hatta – Jl. Sukamulya – Jl. Terusan Jamika – Jl. Jamika – Jl. Sudirman – Jl. Guanin – Jl. Kebon Jati – Jl. Pasir Kaliki – Jl. Pajajaran – Jl. Cihampelas – Jl. Wastu Kencana – Jl. RE. Martadinata – Jl. Purnawarman – UNISBA & UNPAS (Tamansari) – Jl. Tamansari – Jl. Cikapayang – Jl. Surapati (Suci) – Jl. Pahlawan – Jl. Cikutra Barat – Jl. Cigadung Raya – Terminal Dago
18	Panghegar Permai – Dipati Ukur	Terminal Panghegar – Jl. Cisaranten – Jl. Cicukang – Jl. AH. Nasution (Raya Ujung Berung) – Sindanglaya (Nasution) – Terminal Cicaheum – Jl. Ahmad Yani – Cicadas (Ahmad Yani) – Jl. Kiara Condong – Jl. Jakarta – Jl. Sukabumi – Jl. Laswi – Jl. RE. Martadinata – Jl. Ambon – Masjid Istiqamah – Jl. Cisanggarung – Jl. Cimanuk – Jl. Cimandiri – Jl. Cimalaya – Jl. Diponegoro – Jl. Sulanjana – Jl. Tamansari – Jl. Ganesha – ITB – RS. Boromeus – Jl. Hasanudin – Jl. Dipati Ukur – Terminal Dipati Ukur
19.A	Ciroyom – Sarijadi 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
23	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
24	Sederhana – Cimindi	Jl. Sederhana – Jl. Sukajadi – Jl. Sukagalih – Jl. Cipedes Kidul – Jl. DR. Junjuran (Terusan Pasteur) – Jl. Gunung Batu – Cimindi
25	Ciwastra – Ujung Berung	Terminal Ciwastra – Riung Bandung – Jl. Sukarno-Hatta – Guruminda – Jl. Cisaranten – Jl. Cicukang – Jl. AH. Nasution (Raya Ujung Berung) – Pasar Ujung Berung

Kode	Lintasan Trayek	Jalur
26	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 Tegalega
27	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
30	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
31	Antapani – Ciroyom	Terminal Antapani – Jl. Cibatu (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
32	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
33	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

Kode	Lintasan Trayek	Jalur
34	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
35	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
36	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

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

- $g(n)$ = jarak yang sudah ditempuh dari start ke node/simpul n.
- $h(n)$ = perkiraan jarak dari node n ke tujuan (heuristik).
- $f(n)$ = total biaya (estimated cost) untuk mencapai tujuan lewat node n.

Menentukan trayek angkot paling efisien dari UNPAS Setiabudi – UNPAS Tamansari.

Daftar trayek yang melewati kedua titik :

1. Trayek 3 : Abdul Muis – Ledeng

Jl. Setiabudi – Jl. Cihampelas – Jl. Wastu Kencana

Setiabudi – Cihampelas = 3,9 km

Cihampelas – Wastu kencana = 2,3km

Total jarak Trayek 03:

$$3,9 + 2,3 = 6,2 \text{ km}$$

2. Trayek 15 : Margahayu Raya – Ledeng

Jl. Setiabudi – Jl. Karang Sari – Jl. Sukajadi – Jl. Cemara – Jl. Jurang –

Jl. Sederhana – Jl. Eyckman – Jl. Cihampelas – Jl. Wastu Kencana

Setiabudi – Karangsari = 1,9km

Karangsari – Sukajadi = 2,5 km

Sukajadi – Cemara = 0,45km

Cemara – Jurang = 0,7km

Jurang – Sederhana = 0,65km

Sederhana – Eyckman = 0,5km

Eyckman – Cihampelas = 1,7km

Cihampelas – Wastukencana = 2,3km

Total jarak Trayek 15 :

Tabel 4. 2 Total Jarak Trayek 15

Node	$g(n)$	$h(n)$	$f(n) = g(n) + h(n)$
Setiabudi	0	6,2	6,2
Karangsari	$0 + 1,9 = 1,9$	4,3	6,2
Sukajadi	$1,9 + 2,5 = 4,4$	3,5	7,9
Cemara	$4,4 + 0,45 = 4,85$	3,0	7,85
Jurang	$4,85 + 0,7 = 5,55$	2,3	7,85
Sederhana	$5,55 + 0,65 = 6,2$	1,8	8,0
Eyckman	$6,2 + 0,5 = 6,7$	1,3	8,0
Cihampelas	$6,7 + 1,7 = 8,4$	2,3	10,7
Wastu Kencana	$8,4 + 2,3 = 10,7$	0	10,7

Perbandingan Rute

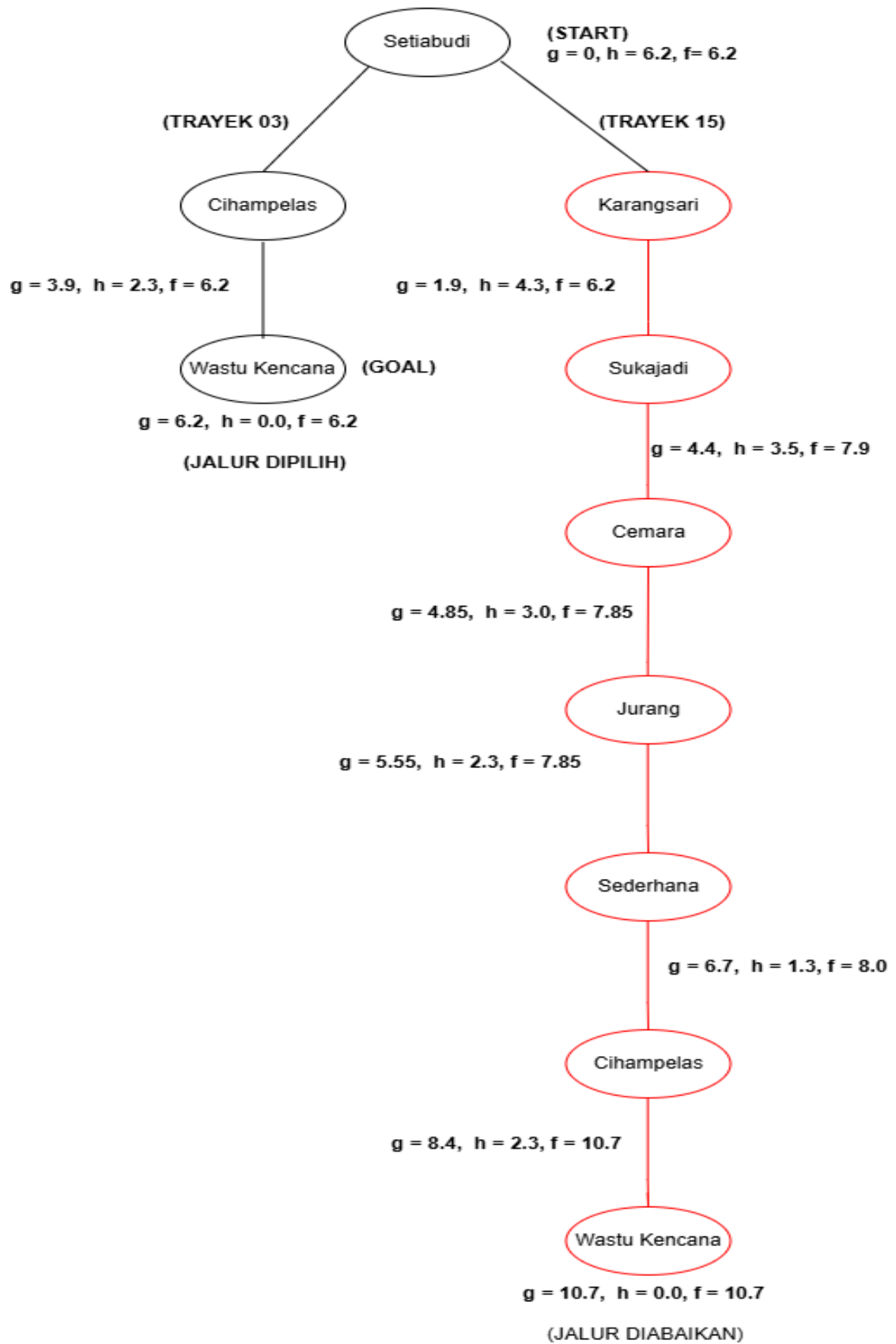
Rute 3 mempunyai jarak keseluruhan 6,2 km

Rute 15 mempunyai jarak keseluruhan 10,7 km

Dari perbandingan itu,

dapat disimpulkan bahwa rute 3 adalah jalur terpendek dan paling efisien, sedangkan rute 15 adalah jalur yang lebih panjang dan kurang efisien.

Penerapan algoritma A* dalam aplikasi web transportasi umum ini diharapkan dapat memberikan rekomendasi rute tercepat bagi penumpang dengan mempertimbangkan titik-titik koordinat trayek angkot sebagai *nodes*. Dengan mengoptimalkan fungsi $g(n)$ sebagai jarak antar titik henti dan $h(n)$ sebagai jarak geografis ke tujuan.



Gambar 4. 2 Graf Pohon Simulasi A*

4.1.2. Model Analisis

Model analisis merupakan gambaran atau struktur sistematis yang digunakan untuk memahami, menjabarkan, meramalkan, atau menilai suatu masalah, sistem, atau fenomena. Model analisis menjadi dasar untuk memahami keinginan pengguna, fitur sistem, data yang terlibat, dan hubungan antara elemen-elemen sistem.

Secara teori, model analisis memiliki beberapa dimensi utama yang harus dijelaskan:

- Tujuan Utama Model Analisis

Model ini dikembangkan dengan tiga sasaran utama:

- Mendeskripsikan Kebutuhan: Menjabarkan apa yang dibutuhkan oleh pengguna secara spesifik dan terukur.
- Membangun Dasar Desain: Memberikan landasan yang kuat untuk tahap perancangan arsitektur, antarmuka, dan struktur data sistem.
- Menetapkan Standar Validasi: Menentukan kumpulan persyaratan yang dapat diuji atau divalidasi setelah sistem selesai dibangun.

- Komponen Struktural Model Analisis

Untuk memahami fenomena atau sistem secara utuh, model analisis biasanya dibagi menjadi tiga representasi:

- Model Data (Data Modeling): Menjelaskan objek data yang terlibat, karakteristik objek tersebut, serta hubungan (relasi) antar objek data di dalam sistem.
- Model Fungsional (Functional Modeling): Menjabarkan fitur-fitur sistem dan bagaimana data bertransformasi saat mengalir melalui sistem.
- Model Perilaku (Behavioral Modeling): Menggambarkan bagaimana sistem memberikan respon terhadap kejadian atau stimulus dari lingkungan eksternal.

4.1.2.1. Diagram Aktivitas

Diagram Aktivitas (*activity diagram*) merupakan gambar yang menunjukkan langkah-langkah kerja dari awal hingga akhir dalam suatu sistem atau bagian dari sistem tersebut. Diagram aktivitas berfungsi untuk menggambarkan tindakan yang melibatkan satu atau lebih pihak dalam sistem. Selain untuk memperlihatkan sifat dinamis dari suatu sistem, diagram aktivitas juga digunakan untuk menunjukkan urutan dari satu kegiatan ke kegiatan yang lain..

Tabel 4. 3 Diagram Aktivitas

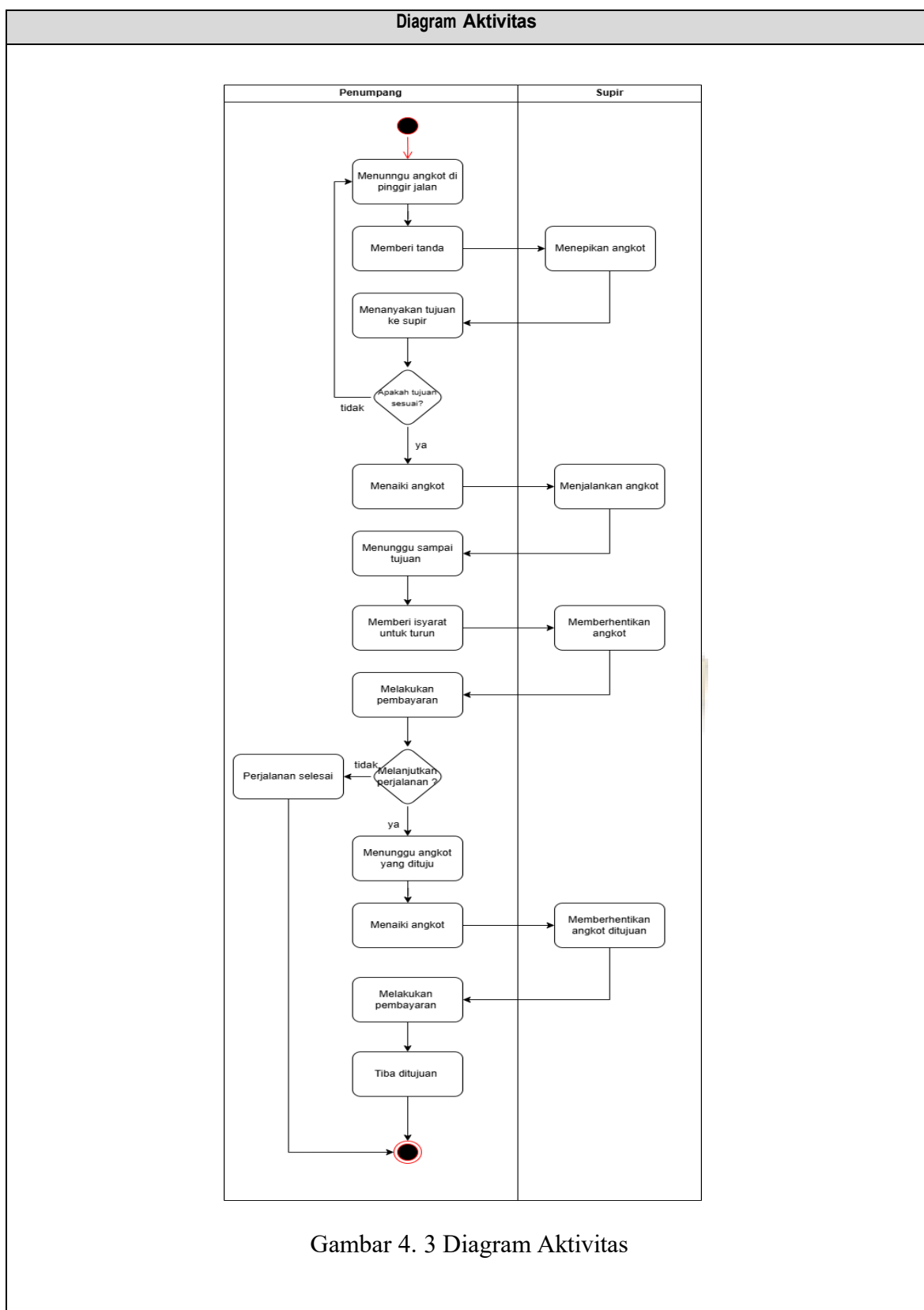


Diagram ini menggambarkan langkah-langkah interaksi antara penumpang dan sopir angkot saat menggunakan sarana transportasi umum (angkot), mulai dari menunggu hingga tiba di tujuan akhir.

Proses dimulai di sisi penumpang, ketika penumpang menunggu angkot di pinggir jalan. Ketika angkot lewat, penumpang memberikan sinyal, dan sopir akan menepikan kendaraan. Setelah itu, penumpang memberitahukan tujuan kepada sopir. Jika tujuan

tidak cocok, penumpang tidak akan menaiki angkot tersebut. Namun, jika tujuannya sesuai, penumpang akan masuk ke dalam angkot dan sopir akan mulai mengemudikan kendaraan. Ketika hampir tiba, penumpang memberikan tanda untuk berhenti, lalu sopir menghentikan angkot. Selanjutnya, penumpang melakukan pembayaran. Setelah itu, penumpang harus memutuskan apakah akan melanjutkan dengan angkot lain (misalnya karena perlu transit) atau tidak. Jika perjalanan sudah selesai, proses akan berakhir di situ. Sebaliknya, jika penumpang ingin melanjutkan perjalanan, ia akan menunggu angkot berikutnya yang sesuai dengan tujuan. Setelah angkot yang diinginkan datang, penumpang masuk ke dalam angkot dan melalui proses pemberhentian serta pembayaran lagi, seperti yang sebelumnya. Sopir akan menghentikan angkot di tempat yang diinginkan penumpang. Akhirnya, penumpang sampai di tujuan dan proses pun selesai. Diagram ini menunjukkan proses perjalanan angkot dengan cara yang simpel namun mendalam, yang melibatkan interaksi langsung antara penumpang dan sopir dalam dua situasi: perjalanan tunggal dan perjalanan berlanjut (*transit*).

4.1.2.2. Identifikasi User Requirement

User Requirement adalah penjelasan mengenai apa yang diharapkan pengguna dari perangkat lunak yang akan dikembangkan. Tabel di bawah ini menyajikan daftar kebutuhan pengguna dari perangkat lunak yang akan dibuat.

Tabel 4. 4 User Requirement

Kode	User Requirement
UR - 01	User masyarakat ingin dapat melakukan pencarian rute angkot berdasarkan lokasi
UR - 02	User masyarakat ingin dapat melihat deskripsi mengenai detail informasi berupa warna, rute, dan juga tarif.

4.1.2.3. Identifikasi Software Requirement

Analisis kebutuhan perangkat lunak merupakan tahapan yang bertujuan untuk mengidentifikasi keinginan dan harapan dari pengguna akhir serta para pemangku kepentingan terkait perangkat lunak yang akan dibuat. Tahapan ini sangat penting agar perangkat lunak yang dikembangkan dapat memenuhi kebutuhan bisnis dan fungsi yang diinginkan.

Tabel 4. 5 Software Requirement

Kode	Software Requirement
SR - 01	Perangkat lunak harus dapat memberikan informasi mengenai rute berdasarkan lokasi yang dicari.
SR - 02	Perangkat Lunak harus dapat menampilkan deskripsi mengenai detail informasi berupa warna, rute, dan tarif angkor

4.1.2.4. Functional Requirement

Kebutuhan Fungsional adalah spesifikasi yang menjelaskan apa saja yang harus dilakukan oleh sebuah sistem untuk memenuhi tujuan pengguna atau proses bisnis.

Tabel 4. 6 Functional Requirement

Kode	Kebutuhan Fungsional	Penjelasan
FR - 01	Perangkat lunak dapat menampilkan informasi rute berdasarkan lokasi	Perangkat lunak harus dapat menampilkan seluruh informasi rute berdasarkan lokasi yang dicari
FR - 02	Perangkat lunak dapat menampilkan detail informasi	Peranglat lunak harus dapat menampilkan seluruh

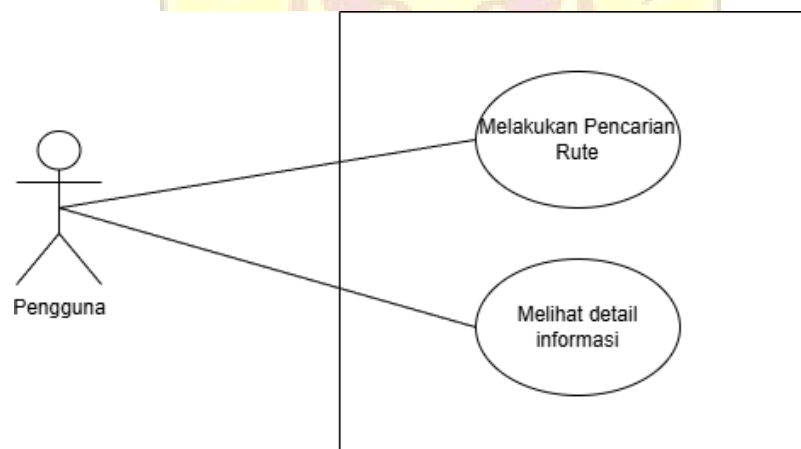
	berupa tarif, rute dan juga gambar serta warna angkutan kota	informasi mengenai tarif, rute, fan juga gambar serta warna angkutan kota
--	--	---

4.1.3. Model Use Case

Model use case adalah salah satu tipe analisis model yang sangat esensial dalam pengembangan perangkat lunak. Tipe ini menitikberatkan pada hubungan antara pengguna (atau sistem luar) dengan sistem yang sedang dikembangkan untuk mencapai tujuan tertentu.

4.1.3.1. Diagram Use Case

Diagram *use case* lazimnya disebut dengan *behavior diagrams* yang berfungsi untuk menjelaskan serangkaian aksi (*use case*) yang perlu atau dapat dilakukan oleh berbagai sistem atau sistem (*subjek*) dalam kolaborasi dengan satu atau lebih pengguna eksternal dari sistem tersebut (aktor). Setiap use case mendefinisikan fungsi yang akan dikembangkan dalam sistem yang direncanakan [NAZ22]. *Use case* diagram dari *requirement* yang ada pada aplikasi pencarian rute transportasi umum ini dapat dilihat pada gambar dibawah ini.



Gambar 4. 4 Diagram Use Case

Tabel 4. 7 Deskripsi Use Case

Kode	Nama Use Case	Deskripsi
UC - 01	Mekakukan pencarian rute	Pengguna ingin dapat melakukan pencarian rute angkot di website
UC - 02	Melihat detail Informasi	Pengguna ingin dapat melihat detail informasi yang tersedia pada website

4.1.3.2. Definisi Aktor

Aktor adalah entitas eksternal yang berinteraksi dengan sistem tetapi tidak termasuk dalam sistem itu sendiri. Aktor menunjukkan peran yang dijalankan oleh entitas tersebut saat berinteraksi dengan sistem untuk mencapai tujuan tertentu (yang ditunjukkan oleh *use case* diagram). Untuk memperjelas, dapat dilihat pada tabel di bawah ini yang memuat daftar aktor lengkap dengan deskripsinya.

Tabel 4. 8 Definisi Aktor

Aktor	Deskripsi
Pengguna	Orang yang akan mengakses website untuk melihat rute angkutan kota (angkot) dan membaca artikel atau informasi yang tersedia pada halaman beranda.

4.2. Perancangan

Perancangan merupakan langkah yang diambil setelah melakukan analisis kebutuhan. Sasaran dari tahap ini adalah mengubah kebutuhan dari pengguna serta spesifikasi fungsional menjadi sebuah representasi yang teratur dan rinci dari solusi perangkat lunak.

- Tingkatan Perancangan (*Design Levels*)

Perancangan tidak dilakukan sekaligus, melainkan melalui beberapa lapisan kedalaman:

- Perancangan Arsitektur (*Architectural Design*): Menetapkan struktur keseluruhan sistem. Ini melibatkan identifikasi komponen perangkat lunak (sub-sistem), hubungan antar komponen, serta pemilihan pola arsitektur seperti *Client-Server*, *Microservices*, atau *Model-View-Controller (MVC)*.
- Perancangan Antarmuka (*Interface Design*): Fokus pada bagaimana sistem berkomunikasi dengan pengguna (UI/UX) dan bagaimana komponen sistem saling berkomunikasi satu sama lain melalui API.
- Perancangan Prosedural/Komponen: Menentukan algoritma, logika kontrol, dan struktur data yang akan digunakan di dalam setiap modul secara spesifik.

4.2.1. Skenario Use Case

Skenario use case merupakan langkah-langkah proses use case dari perspektif aktor dan sistem. Setiap use case terkecil menghasilkan skenario use case, misalnya untuk generalisasi, maka skenario yang dianalisis adalah use case yang lebih spesifik. Skenario normal terjadi ketika sistem berfungsi tanpa kesalahan atau error. Sementara itu, skenario alternatif muncul ketika sistem tidak berfungsi dengan baik atau mengalami masalah. Baik skenario normal maupun alternatif bisa memiliki lebih dari satu variasi. Alur dari skenario inilah yang akan menjadi dasar untuk membuat sequence diagram atau diagram urutan.

4.2.1.1. UC – 01 Melakukan Pencarian Rute

Tabel 4. 9 Skenario Use Case UC – 01 Melakukan Pencarian Rute

Identifikasi	
Kode	UC – 01
Nama	Melakukan Pencarian Rute
Tujuan	Membantu pengguna menemukan dan memilih rute angkot dengan jarak yang paling pendek menuju tujuan yang diinginkan.
Deskripsi	Use case melakukan pencarian rute digunakan pengguna untuk mencari rekomendasi rute yang lebih efisien sesuai dengan rute yang ingin dituju. Sistem akan membantu merekomendasikan rute yang lebih efisien berdasarkan jarak yang lebih pendek kepada pengguna.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal : Berada di halaman home	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol cari rute	2. Menampilkan halaman pencarian rute
3. Mengisi field titik awal dan titik yang ingin dituju	
4. Menekan tombol cari rute	
	5. Menampilkan card berupa informasi singkat trayek angkutan yang berbasis gambar angkutan, nama angkutan, estimasi jarak dan tarif sesuai dengan rute yang ingin dituju
6. Mnekan tombol lihat detail trayek	
	7. Menampilkan halaman informasi trayek persegment yang terdiri dari nama rute, estimasi jarak, estimasi tarif, daftar titik-titik yang dilewati, visualisasi peta, dan juga gambar angkutan sesuai dengan rute yang dicari. (Terminal Kebon Kelapa–Jl. Pungkur)
8. Menekan tombol kembali	
	9. Menampilkan halaman pencarian
10. Menekan teks link beranda pada navbar	
	11. Menampilkan halaman beranda aplikasi
Kondisi akhir : Berada di halaman beranda	

4.2.1.2. UC – 02 Melihat detail informasi

Tabel 4. 10 Skenario Use Case UC – 02 Melihat detail informasi

Identifikasi	
Kode	UC – 02
Nama	Melihat detail informasi
Tujuan	Memberikan akses kepada pengguna agar pengguna dapat memperoleh informasi detail keseluruhan mengenai angkot yang tersedia pada sistem.
Deskripsi	Use case melihat detail informasi menyediakan informasi detail mengenai angkot untuk membantu pengguna menemukan angkot yang sesuai dengan kebutuhan pengguna.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal : Berada di halaman beranda	
Aksi Aktor	Reaksi Sistem
1. Menekan teks link kategori yang berada di navbar	2. Menampilkan halaman kategori yang berisi card informasi trayek yang terdiri dari gambar angkot dan nama angkot.
3. Menekan tombol detail pada card trayek sesuai keinginan	4. Menampilkan informasi berupa nama rute, tarif, jarak, titik-titik rute yang dilewati, visualisasi peta, dan juga gambar angkot sesuai detail trayek yang dipilih.
Kondisi akhir : Berada di halaman detail rute	

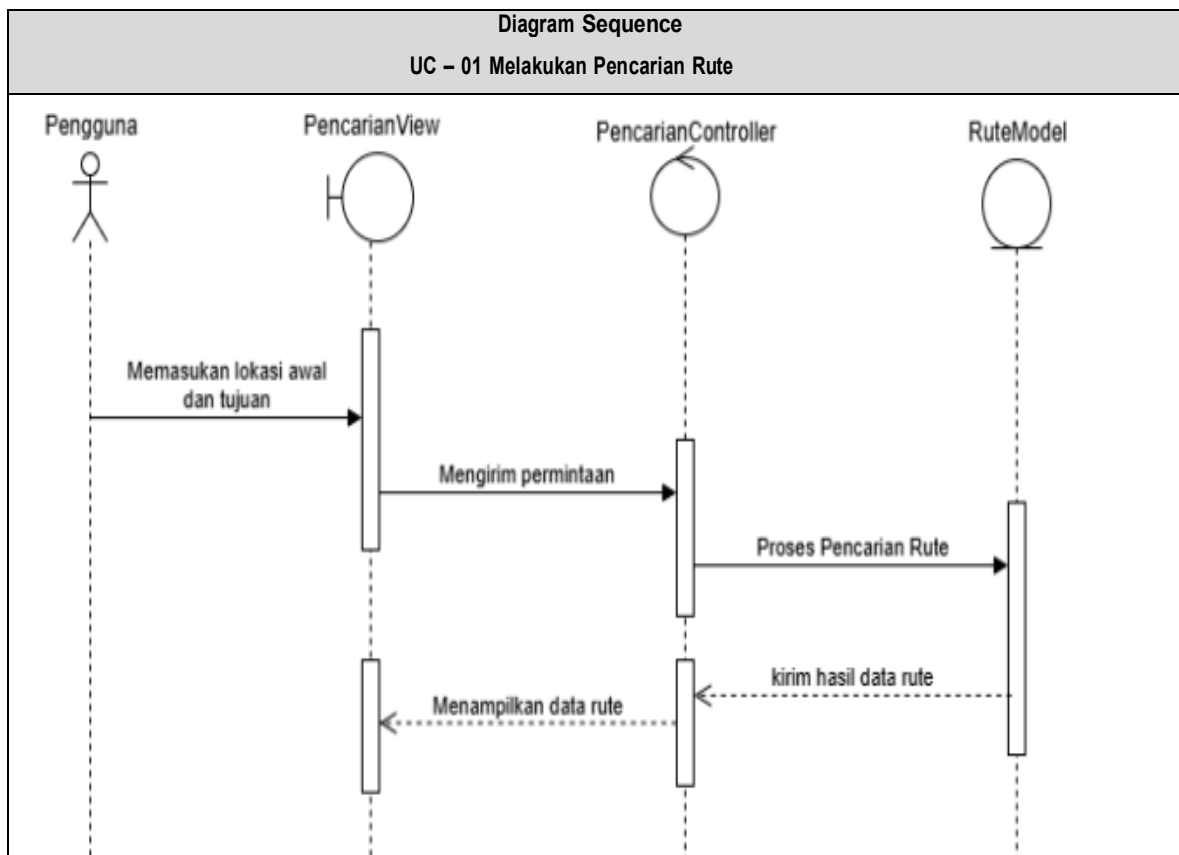
4.2.2. Diagram Sequence

Diagram sequence adalah jenis diagram perilaku yang paling sering digunakan, yang menekankan pada pertukaran informasi di antara berbagai alur sistem. Diagram urutan menunjukkan interaksi dengan menekankan pada rangkaian pesan yang saling diterima serta spesifikasi kemunculannya yang sesuai dalam jalannya sistem.

Pertukaran informasi dalam diagram ini dilakukan melalui berbagai jenis pesan yang mendefinisikan sifat komunikasi antar komponen. Pesan Sinkron digambarkan dengan panah tertutup untuk menunjukkan bahwa pengirim menunggu respon, sementara pesan Asinkron menggunakan panah terbuka untuk komunikasi yang tidak menghalangi proses pengirim. Selain itu, terdapat elemen Combined Fragments yang berfungsi untuk memodelkan logika kompleks seperti percabangan (*alt*), pilihan (*opt*), dan pengulangan (*loop*), sehingga diagram mampu merepresentasikan alur sistem yang tidak hanya linier tetapi juga kondisional.

4.2.2.1. UC – 01 Melakukan Pencarian Rute

Tabel 4. 11 Diagram Sequence UC – 01 Melakukan Pencarian Rute

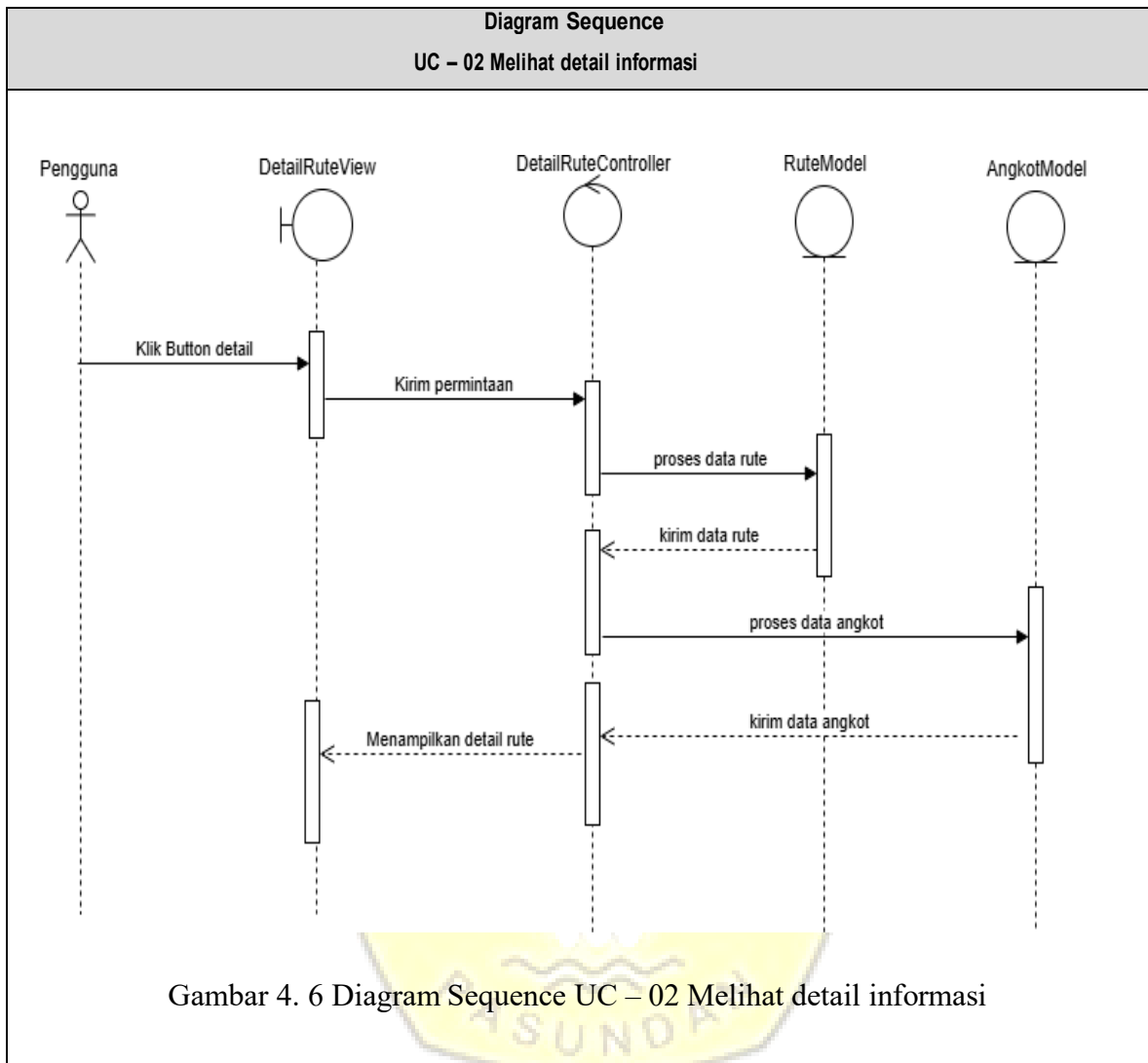


Gambar 4. 5 Diagram Sequence UC – 01 Melakukan Pencarian Rute

Pengguna menginput data tentang lokasi keberangkatan dan tujuan menggunakan antarmuka pengguna (PencarianView). Setelah data berhasil diinput, PencarianView akan mengirimkan permintaan pencarian kepada komponen logika sistem yang dinamakan PencarianController. Controller ini bertugas untuk menangani permintaan yang masuk, kemudian meneruskannya ke RuteModel untuk mencari rute. Selanjutnya, RuteModel akan mencari rute sesuai dengan informasi yang telah diterima. Setelah proses pencarian selesai, RuteModel akan mengirimkan kembali hasil rute yang telah ditemukan kepada PencarianController. Lalu, controller akan mengalihkan data hasil pencarian kepada PencarianView. PencarianView akan menampilkan informasi mengenai rute yang ditemukan untuk pengguna.

4.2.2.2. UC – 02 Melihat detail informasi

Tabel 4. 12 Diagram Sequence UC – 02 Melihat detail informasi



Pengguna mengklik tombol informasi di antarmuka pengguna (DetailRuteView). Tindakan ini mengirimkan permintaan kepada pengendali yang mengurus detail rute, yaitu DetailRuteController. Pengendali tersebut kemudian menangani permintaan dengan pertama-tama mengambil informasi rute dari RuteModel. Setelah data rute berhasil diambil, RuteModel mengirimkan hasilnya kembali ke pengendali. Selanjutnya, DetailRuteController melanjutkan dengan meminta data tambahan dari AngkotModel tentang angkot yang beroperasi di rute tersebut, termasuk warna kendaraan, tarif, dan tipe kendaraan. Setelah proses data angkot selesai, AngkotModel mengirimkan informasi angkot kembali kepada pengendali. Akhirnya, DetailRuteController mengembalikan semua data yang sudah terakumulasi (rute dan angkot) kepada DetailRuteView untuk ditampilkan kepada pengguna sebagai rincian lengkap rute.

4.2.3. Pemodelan Berbasis Kelas

Pemodelan berbasis kelas adalah metode pemodelan perangkat lunak yang menggunakan konsep kelas untuk merepresentasikan struktur dan perilaku sistem. Dalam metode ini, sistem dijelaskan sebagai sekumpulan kelas yang memiliki atribut (untuk menyimpan informasi atau karakteristik) dan operasi/metode (untuk melaksanakan fungsi atau tindakan tertentu).

4.2.3.1. Mengidentifikasi Kelas

Proses mengidentifikasi kelas merupakan langkah krusial dalam menyusun model analisis karena menjadi dasar untuk memahami elemen-elemen sistem dan hubungan yang terjalin di dalamnya. Melalui tahapan ini, setiap objek dalam sistem didefinisikan secara sistematis untuk menjabarkan data serta fitur yang terlibat dalam penyelesaian masalah yang diteliti. Hal ini memastikan bahwa struktur sistem yang dibangun dapat menjabarkan serta menilai keakuratan informasi secara terstruktur sesuai dengan kebutuhan pengguna.

Tabel 4. 13 Identifikasi Kelas

No.	Nama Kelas	Jenis	Deskripsi	Keterangan
View				
1.	Cari-RuteView	View	Kelas ini berfungsi untuk menampilkan halaman pencarian rute angkot yang diminta oleh pengguna. Halaman ini menyediakan formulir untuk memasukkan titik awal dan titik tujuan perjalanan.	Menampilkan form input pencarian rute.
2.	DetailTrayekView	View	Kelas ini berfungsi untuk menampilkan informasi detail trayek angkot, seperti kode trayek, jalur lintasan, dan jalan yang dilalui.	Menampilkan detail informasi trayek angkot.
3.	DetailView	View	Kelas ini berfungsi untuk menampilkan hasil detail rute yang direkomendasikan oleh sistem berdasarkan proses pencarian rute.	Menampilkan hasil rute, angkot, transit, jarak, dan tarif.
4.	ContactView	View	Kelas ini berfungsi untuk menampilkan halaman kontak yang digunakan oleh pengguna untuk mengirimkan pesan, kritik, atau saran terkait aplikasi.	Menampilkan form input kontak pengguna.
5.	HomeView	View	Kelas ini berfungsi untuk menampilkan halaman utama aplikasi sebagai halaman awal sebelum pengguna mengakses fitur lainnya.	Menampilkan halaman beranda aplikasi.
6.	KategoriView	View	Kelas ini berfungsi untuk menampilkan daftar kategori trayek angkot yang tersedia dalam sistem.	Menampilkan kategori trayek angkot.

No.	Nama Kelas	Jenis	Deskripsi	Keterangan
Controller				
11.	AngkotController	Controller	Kelas ini berfungsi untuk mengelola permintaan data angkot dan trayek angkot dari pengguna.	Mengatur logika data angkot dan trayek.
12.	ContactController	Controller	Kelas ini berfungsi untuk memproses data pesan yang dikirimkan oleh pengguna melalui halaman kontak.	Menyimpan data pesan ke database.
13.	HomeController	Controller	Kelas ini berfungsi untuk mengatur data yang dibutuhkan pada halaman utama aplikasi.	Mengatur tampilan halaman beranda.
14.	KategoriController	Controller	Kelas ini berfungsi untuk mengelola data kategori trayek angkot yang ada dalam sistem.	Mengatur dan menampilkan kategori trayek.
15.	PencarianController	Controller	Kelas ini berfungsi untuk memproses permintaan pencarian rute dari pengguna berdasarkan titik awal dan tujuan.	Memanggil A* dan penghitungan tarif.
16.	RuteController	Controller	Kelas ini berfungsi untuk mengelola data rute dan titik lintasan angkot.	Mengatur data rute dan titik rute.
Service				
17.	AstarService	Service	Kelas ini berfungsi untuk menghitung rute terbaik menggunakan algoritma A* berdasarkan titik awal dan tujuan.	Perhitungan $g(n)$, $h(n)$, dan $f(n)$.
18.	TarifService	Service	Kelas ini berfungsi untuk menghitung total tarif perjalanan berdasarkan rute yang dipilih.	Menghitung tarif dan jumlah transit.
Model				
19.	AngkotModel	Model	Kelas ini berfungsi untuk merepresentasikan data angkot dan trayek angkot dalam database.	Akses data trayek angkot.
20.	ContactModel	Model	Kelas ini berfungsi untuk menyimpan data pesan atau kontak dari pengguna.	Menyimpan data kontak.
21.	RuteModel	Model	Kelas ini berfungsi untuk merepresentasikan data rute angkot dalam sistem.	Akses data rute angkot.
22.	TitikRuteModel	Model	Kelas ini berfungsi untuk menyimpan data titik lintasan angkot yang digunakan sebagai node pencarian rute.	Node graph pada algoritma A*.

4.2.3.2. Menentukan Atribut

Menentukan atribut berarti mengidentifikasi data atau properti apa saja yang dimiliki oleh suatu objek (kelas) agar objek tersebut bisa mewakili keadaan atau identitasnya secara lengkap. Atribut berfungsi untuk memberikan rincian teknis yang membedakan satu objek dengan objek lainnya dalam kategori yang sama. Penetapan atribut yang tepat sangat krusial dalam model analisis karena menjadi variabel utama dalam menjabarkan dan meramalkan perilaku sistem saat mengolah data. Untuk lebih jelasnya dapat dilihat pada tabel dibawah ini.

Tabel 4. 14 Menentukan Atribut

No.	Nama Kelas	Atribut	Tipe Data	Contoh Nilai	Deskripsi
Model					
1.	AngkotModel	Id_angkot (PK)	INT	1	Primary key sebagai identitas unik angkot, bersifat auto-increment.
		nama_angkot	VARCHAR	Abdul Muis - Cicaheum	Atribut yang digunakan untuk menyimpan nama trayek angkot.
		kode	VARCHAR	01A	Atribut yang digunakan untuk mendeskripsikan kode trayek angkot yang bersifat unik.
		warna	VARCHAR	Hijau Strp Kuning	Atribut yang digunakan untuk mendeskripsikan warna fisik angkot.
		tarif	INT	7000	Atribut yang digunakan untuk menyimpan tarif angkot dalam satuan Rupiah.
		gambar	VARCHAR	angkot1A.png	Atribut yang digunakan untuk menyimpan path atau nama file gambar angkot.
2.	RuteModel	Id_rute (PK)	INT	12	Primary key sebagai identitas unik rute angkot.
		Id_angkot (FK)	INT	1	Foreign key yang menghubungkan rute dengan angkot tertentu.
		Nama_rute	VARCHAR	Terminal Kebon Kelapa – Terminal Cicaheum (Via Binong)	Atribut yang digunakan untuk menyimpan nama jalur rute angkot.
		jarak	DOUBLE	12.5	Atribut yang digunakan

					untuk menyimpan total jarak rute dalam satuan kilometer.
3.	TitikRuteModel	id_titik_rute (PK)	INT	100	Primary Key, auto-increment, simpul graf.
		id_rute (FK)	INT	10	Foreign key yang menghubungkan titik rute dengan rute tertentu.
		urutan	INT	3	Atribut yang menunjukkan urutan titik dalam suatu rute.
		nama_titik	VARCHAR	Jl Cihampelas	Atribut yang digunakan untuk menyimpan nama lokasi atau titik lintasan.
		latitude	DOUBLE	-6.921234	Atribut yang digunakan untuk menyimpan koordinat lintang titik rute.
		longitude	DOUBLE	107.607891	Atribut yang digunakan untuk menyimpan koordinat bujur titik rute.
4.	ContactModel	Id	INT	3	Primary key sebagai identitas unik pesan kontak.
		Nama	VARCHAR	Rika Wenda	Atribut yang digunakan untuk menyimpan nama pengirim pesan.
		Email	VARCHAR	rikawenda@gmail.com	Atribut yang digunakan untuk menyimpan alamat email pengirim.
		pesan	TEXT	Aplikasi sangat membantu	Atribut yang digunakan untuk menyimpan isi pesan, kritik, atau saran dari pengguna.

4.2.3.3. Menentukan Operasi

Menentukan operasi adalah proses untuk mengidentifikasi fungsi, tindakan, atau perilaku apa saja yang bisa dilakukan oleh sebuah objek (class) agar objek tersebut mampu menjalankan perannya di dalam sistem secara optimal.

Tabel 4. 15 Menentukan Operasi

No.	Nama Kelas	Nama Operasi	Kandidat Perilaku
View			
1	Cari-RuteView	showSearchForm()	Menampilkan form pencarian rute yang berisi input titik awal, titik tujuan, dan tombol cari rute.

		inputAsal()	Menyediakan field input untuk titik awal perjalanan yang dapat diisi pengguna.
		inputTujuan()	Menyediakan field input untuk titik tujuan perjalanan.
		submitSearch()	Mengirim data titik awal dan tujuan ke PencarianController melalui metode POST.
		showRecommendation(trayeks)	Menampilkan daftar rute angkot hasil rekomendasi algoritma A*.
		showTransitRecommendation(transitResults)	Menampilkan rekomendasi rute dengan transit jika rute langsung tidak tersedia.
		showMiniMap(mapSegment)	Menampilkan peta mini Leaflet untuk setiap hasil rute rekomendasi.
		showEstimation(jarak, tarif)	Menampilkan estimasi jarak dan tarif untuk setiap rute rekomendasi.
		createDetailLink(ruteld, awal, tujuan)	Membuat tautan menuju halaman detail trayek atau detail segmen rute.
		autocompleteLokasi()	Menyediakan fitur autocomplete lokasi titik berdasarkan AJAX.
2.	DetailTrayekView	showTrayekInfo(rute)	Menampilkan informasi utama trayek seperti nama rute dan nama angkot.
		showDistance()	Menampilkan jarak total rute trayek.
		showAngkotImage(gambar)	Menampilkan gambar angkot sesuai trayek.
		showTitikRuteList(titik_rutes)	Menampilkan daftar titik lintasan yang dilewati angkot secara berurutan.
		showTarif(tarif)	Menampilkan tarif angkot berdasarkan trayek.
		renderRouteMap(titik_rutes)	Menampilkan peta trayek angkot menggunakan Leaflet dan polyline.
		createBackLink()	Menyediakan tombol kembali ke halaman hasil rekomendasi sebelumnya.
3.	DetailView	showSegmentInfo(rute)	Menampilkan informasi rute segmen yang dipilih pengguna.
		showSegmentDistance(jarak_segment)	Menampilkan jarak segmen perjalanan hasil A*.
		showSegmentTarif(tarif_segment)	Menampilkan tarif segmen rute perjalanan.
		showTransitInfo()	Menampilkan informasi transit jika rute terdiri dari lebih dari satu trayek.
		showSegmentTitik(titik_segment)	Menampilkan titik lintasan segmen yang dilewati.
		renderSegmentMap(titik_segment)	Menampilkan peta segmen rute menggunakan Leaflet.
		createNextTransitLink(id_rute, tujuan)	Menyediakan tombol untuk melanjutkan ke trayek berikutnya (transit manual).
4.	TransitManualView	showTrayekAwal(trayekAwal)	Menampilkan informasi trayek awal yang digunakan pengguna.

		showTransitPoint(titikTransit)	Menampilkan titik transit tempat pengguna harus berpindah angkot.
		showTrayekLanjutan(trayekLanjutan)	Menampilkan daftar trayek lanjutan yang tersedia dari titik transit.
		selectNextTrayek(id_rute)	Menyediakan tombol untuk memilih trayek lanjutan dan melihat detail rute.
5.	KategoriView	showCategoryList(rutes)	Menampilkan daftar trayek angkot dalam bentuk grid kartu.
		showSearchBar()	Menampilkan input pencarian trayek atau titik angkot.
		filterByKeyword(keyword)	Menyaring trayek berdasarkan kata kunci pencarian.
		sortTrayek(criteria)	Mengurutkan trayek berdasarkan nama atau tarif.
		filterByWarna(warna)	Menyaring trayek berdasarkan warna angkot.
		showTrayekCard(rute)	Menampilkan kartu trayek berisi nama, angkot, tarif, dan titik awal-akhir.
		createDetailLink(ruteld)	Membuat tautan menuju halaman detail trayek.
6.	ContactView	showContactForm()	Menampilkan form kontak yang berisi input nama, email, dan pesan.
		inputNama()	Menyediakan field input nama pengirim pesan.
		inputEmail()	Menyediakan field input email pengirim pesan.
		inputPesan()	Menyediakan field textarea untuk isi pesan.
		submitContact()	Mengirim data kontak ke ContactController untuk disimpan.
		showSuccessMessage()	Menampilkan notifikasi bahwa pesan berhasil dikirim.
7.	HomeView	showHomePage()	Menampilkan halaman utama aplikasi Rute.in.
		showHeroSection()	Menampilkan section hero berisi deskripsi singkat aplikasi.
		showFeatureList()	Menampilkan keunggulan sistem seperti A*, peta interaktif, dan info angkot.
		createSearchLink()	Menyediakan tombol navigasi menuju halaman Cari Rute.
Controller			
7.	AngkotController	getLokasi(request)	Menerima parameter pencarian lokasi dari AJAX autocomplete dan mengembalikan daftar nama titik rute yang cocok.
			Melakukan query ke TitikRute menggunakan pencarian LIKE berdasarkan nama titik.
			Mengembalikan data hasil pencarian dalam

			format JSON untuk kebutuhan autocomplete di View.
8.	ContactController	index()	Menampilkan halaman Contact melalui <code>contact.blade.php</code> .
		store(request)	Melakukan validasi input nama, email, dan pesan dari pengguna.
			Menyimpan data pesan ke database menggunakan <code>ContactModel</code> .
			Mengirimkan email notifikasi ke admin menggunakan <code>Laravel Mail</code> .
		Mengembalikan respon sukses ke halaman sebelumnya setelah pesan berhasil dikirim.	
9.	HomeController	index()	Menampilkan halaman beranda aplikasi melalui <code>home.blade.php</code>
10.	KategoriController	index(request)	Mengambil data rute beserta relasi angkot dan titik rute.
			Menyaring data rute berdasarkan kata kunci pencarian trayek, angkot, atau titik rute.
			Melakukan filter rute berdasarkan wilayah (titik awal tertentu).
			Melakukan pengurutan data rute berdasarkan tarif (murah/mahal).
			Melakukan pagination data rute untuk tampilan kategori.
			Mengambil daftar nama titik rute untuk kebutuhan autocomplete.
			Mengirimkan data rute dan titik ke <code>KategoriView</code> .
11.	PencarianController	construct(aster, tarif)	Melakukan dependency injection <code>AStarService</code> dan <code>TarifService</code> .
		cariRute(request)	Melakukan validasi input titik awal dan titik tujuan pengguna.
			Mencari trayek langsung yang melewati titik awal dan tujuan.
			Mengurutkan titik rute berdasarkan urutan lintasan.
			Mengurutkan titik rute berdasarkan urutan lintasan.
			Menghitung jarak segmen menggunakan rumus Haversine.
			Memanggil <code>TarifService</code> untuk menghitung tarif berdasarkan jarak segmen.
			Menyusun data <code>map_segment</code> untuk visualisasi peta Leaflet.

			Mengurutkan hasil trayek berdasarkan jarak terpendek.
			Mengirimkan hasil rekomendasi rute ke Cari-RuteView.
		haversine(lat1, lon1, lat2, lon2)	Menghitung jarak antar dua titik koordinat menggunakan metode Haversine.
12.	RuteController	index()	Menampilkan halaman pencarian rute awal beserta daftar titik rute.
		search(request)	Memvalidasi input titik awal dan tujuan berdasarkan ID titik rute.
			Mencari rute langsung yang melewati titik awal dan tujuan.
			Menyusun segmen titik rute berdasarkan urutan lintasan.
		detailSegment(request, id)	Menampilkan detail segmen rute berdasarkan titik awal dan tujuan.
			Menghitung jarak segmen dan tarif perjalanan.
			Mengirim data detail rute segmen ke DetailView.
		transitManual(request, id)	Menampilkan trayek lanjutan berdasarkan titik transit yang dipilih pengguna.
		transit(request)	Menampilkan daftar trayek untuk kebutuhan transit otomatis.
detailTrayek(id)	Menampilkan detail trayek lengkap tanpa segmentasi		
haversine(lat1, lon1, lat2, lon2)	Menghitung jarak antar titik lintasan menggunakan rumus Haversine.		
Service			
9.	AStarService	findPath (startNode, goalNode, criterion)	Menerima ID Node dan kriteria (distance atau time), mengimplementasikan logika $f(n) = g(n) + h(n)$, dan mengembalikan jalur terbaik.
		calculateHeuristic (nodeA, nodeB)	Menghitung Jarak Haversine sebagai nilai Heuristik $h(n)$.
		checkConnectivity (startId, goalId)	Memeriksa koneksi awal untuk mencegah A* berjalan pada Graf yang terputus.
		getNeighbors (nodeId)	Mengambil semua Node yang terhubung dari KoneksiNodeModel dan bobotnya ($g(n)$).
10.	TarifService	hitungTarif(km)	Menerima input jarak perjalanan dalam satuan kilometer dan menentukan tarif dasar angkot.
			Mengecek apakah jarak perjalanan kurang dari atau sama dengan 3 km.
			Mengembalikan tarif tetap sebesar Rp3.000

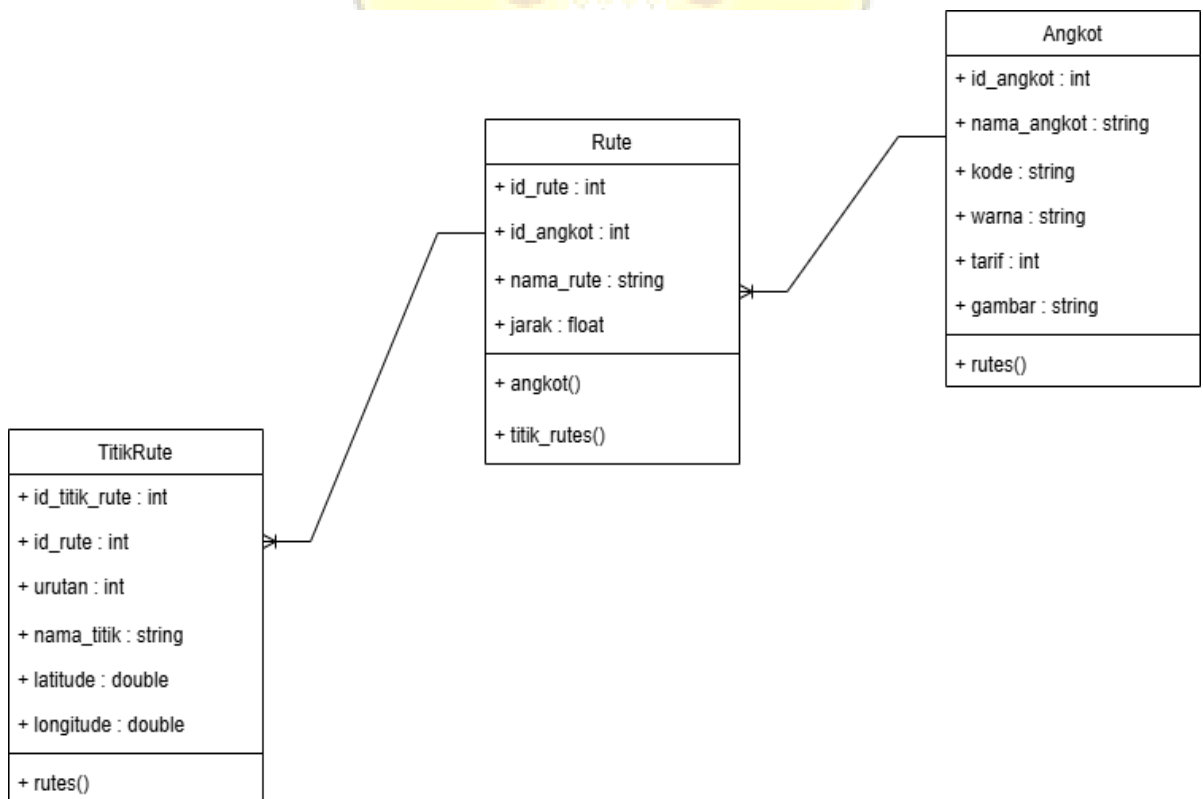
			jika jarak \leq 3 km.
			Menghitung selisih jarak di atas 3 km jika perjalanan melebihi batas tarif dasar.
			Membulatkan ke atas setiap tambahan 1 km menggunakan fungsi ceil.
			Menambahkan Rp1.000 untuk setiap 1 km tambahan di atas 3 km.
			Mengembalikan total tarif perjalanan dalam bentuk nilai integer (Rupiah).
Model			
10.	AngkotModel	routes()	Mengambil seluruh data rute yang dimiliki oleh satu angkot berdasarkan relasi one-to-many. Menghubungkan tabel angkot dengan tabel rute menggunakan foreign key id_angkot.
		create(data)	Mengubah tarif dan memvalidasi nilai.
		getRoutes ()	Mengambil semua Rute yang dilayani.
		create (data)	Menyimpan data angkot baru.
		update (id, data)	Memperbarui data angkot berdasarkan ID.
		delete (id)	Menghapus data angkot.
11.	RuteModel	angkot()	Mengambil data angkot yang terkait dengan suatu rute.
		titik_rutes()	Mengambil daftar titik lintasan pada suatu rute secara berurutan.
		create(data)	Menyimpan data rute angkot baru ke database.
		all()	Menyimpan data angkot baru.
		update(data)	Memperbarui data rute angkot yang sudah ada.
		delete (id)	Menghapus data rute angkot dari database.
		whereHas(condition)	Menyaring rute berdasarkan relasi titik rute atau angkot tertentu.
12.	ContactModel	save()	Menyimpan perubahan data pesan kontak
		delete())	Menghapus data pesan kontak dari database.
		all()	Mengambil seluruh data pesan kontak (biasanya untuk admin).
		create(data)	Menyimpan data pesan kontak yang dikirimkan pengguna ke database.
13.	TitikRuteModel	rute()	Mengambil data rute yang memiliki titik lintasan tersebut
		create(data)	Menyimpan data titik lintasan baru ke database.
		update(data)	Memperbarui data titik lintasan.

	delete()	Menghapus data titik lintasan dari database.
	orderBy(urutan)	Mengambil titik lintasan berdasarkan urutan perjalanan.
	where(nama_titik)	Menyaring titik lintasan berdasarkan nama lokasi tertentu.

4.2.3.4. Diagram Kelas

Diagram kelas adalah salah satu jenis diagram UML yang bersifat statis dan menggambarkan struktur dari suatu sistem dengan menunjukkan kelas, atribut, operasi, serta hubungan antar kelas.

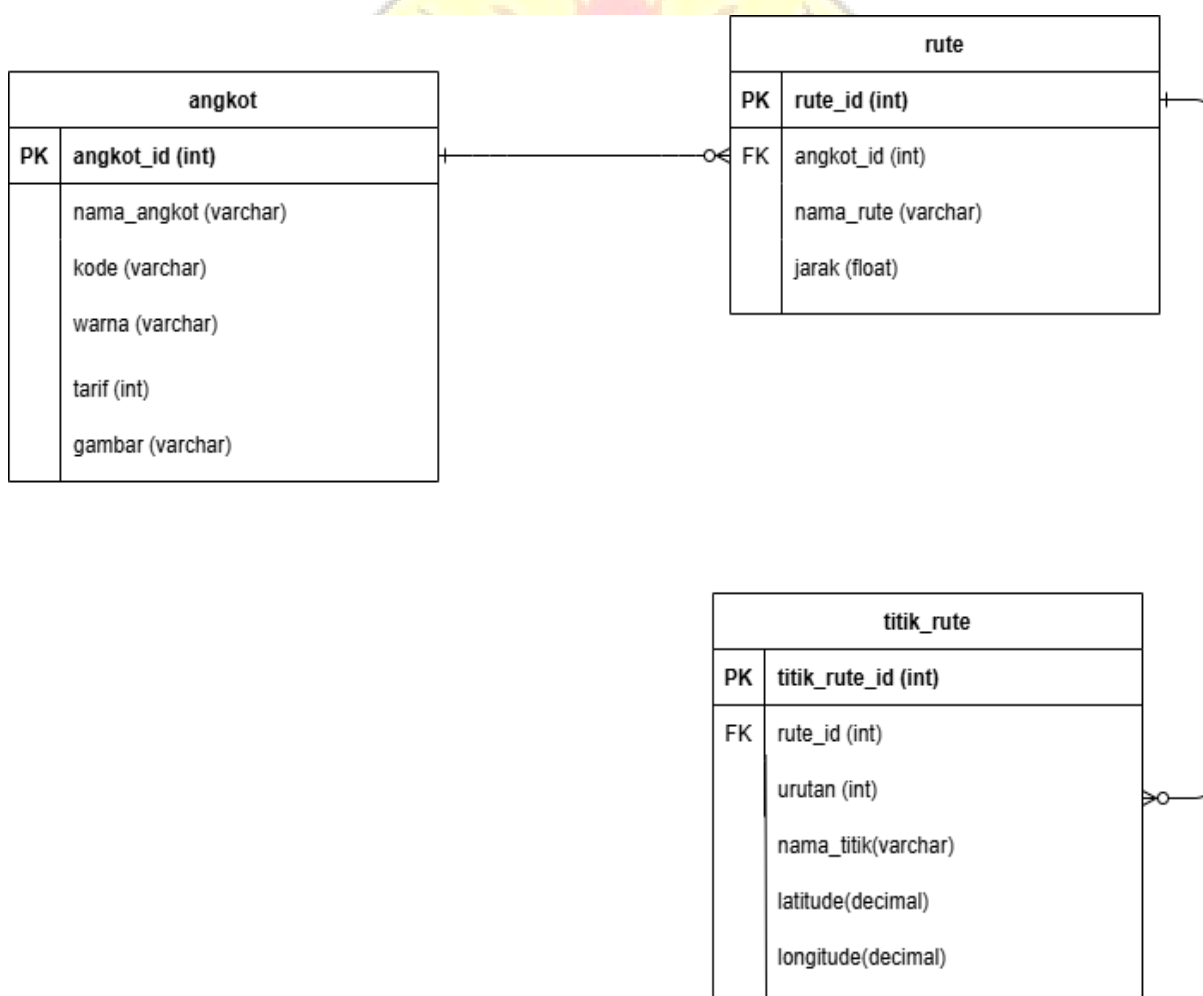
Diagram kelas tidak hanya digunakan untuk memvisualisasikan dan mendokumentasikan berbagai aspek dari suatu sistem, tetapi juga digunakan untuk membangun kode program dari aplikasi perangkat lunak. Diagram kelas juga digunakan untuk membangun fondasi desain arsitektur sistem yang memungkinkan pengembang untuk memahami bagaimana data saling berinteraksi secara teknis. Dengan memetakan hubungan antar elemen secara sistematis, diagram ini membantu dalam menilai dan memvalidasi struktur data agar sesuai dengan fitur yang diinginkan pengguna.



Gambar 4. 7 Diagram Kelas

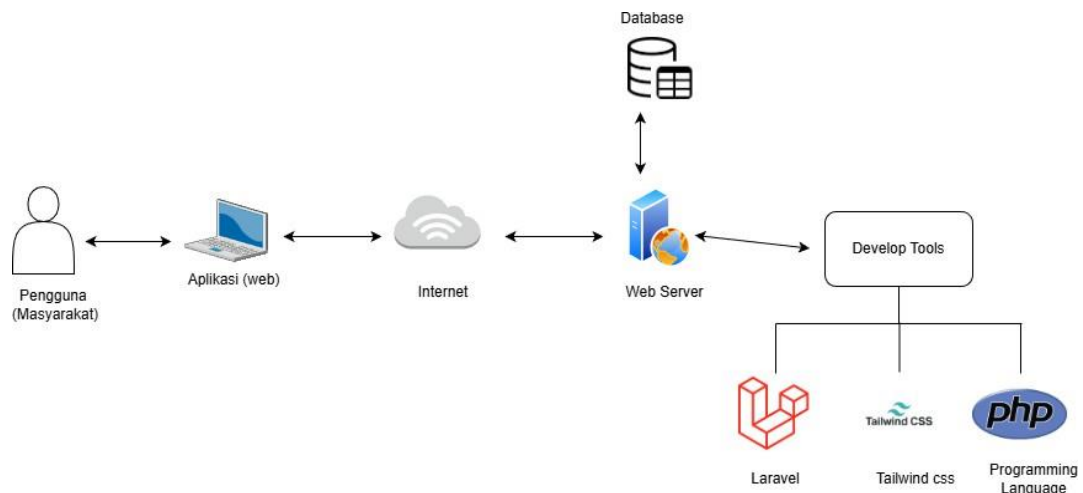
4.2.3.5. Skema Database

Skema basis data adalah definisi atau struktur logis dari basis data, yang mencakup tabel, kolom, relasi, dan aturan yang berlaku. Skema ini biasanya didefinisikan selama tahap perancangan dan memberikan panduan bagaimana data akan diorganisir dan diakses. Selain itu, skema ini berfungsi untuk memastikan integritas data serta meminimalkan redundansi dalam penyimpanan informasi. Keberadaan skema yang terstruktur memungkinkan pengembang untuk memahami hubungan antar elemen sistem secara lebih mendalam guna mendukung fungsionalitas fitur yang dibutuhkan pengguna. Secara teknis, skema basis data bertindak sebagai cetak biru (*blueprint*) dalam menerjemahkan identitas setiap objek ke dalam penyimpanan fisik yang sistematis.



Gambar 4. 8 Skema Database

4.2.3.6. Perancangan Arsitektur



Gambar 4. 9 Perancangan Arsitektur

Perancangan arsitektur sistem pada gambar tersebut menerapkan model berbasis web yang memungkinkan pengguna mengakses aplikasi secara langsung melalui jaringan internet. Pendekatan ini dipilih karena bersifat fleksibel, mudah dikembangkan, serta memiliki pemisahan peran yang jelas antara sisi pengguna dan sisi sistem. Pengguna berinteraksi melalui antarmuka web untuk memasukkan informasi lokasi awal dan lokasi tujuan dalam pencarian rute angkot. Data permintaan tersebut dikirimkan melalui internet ke server web untuk diproses lebih lanjut.

Pada sisi server, sistem dikembangkan menggunakan framework Laravel dengan bahasa pemrograman PHP sebagai backend. Laravel berperan dalam mengelola alur aplikasi, menangani logika bisnis, serta mengimplementasikan algoritma A-Star sebagai metode pencarian rute angkot yang optimal berdasarkan titik awal dan tujuan. Algoritma A-Star digunakan untuk menentukan jalur terbaik dengan mempertimbangkan biaya perjalanan secara efisien. Sementara itu, perancangan tampilan antarmuka dilakukan menggunakan Tailwind CSS agar aplikasi memiliki tampilan yang responsif, konsisten, dan mudah digunakan oleh pengguna dari berbagai perangkat.

4.3. Perancangan Antarmuka Pengguna

Perancangan antarmuka adalah proses desain antarmuka pengguna untuk perangkat lunak dengan fokus pada tampilan dan interaksi. Tujuannya adalah untuk membuat interaksi pengguna efisien dan intuitif, di mana tampilan antarmuka menyediakan akses mudah ke fungsi-fungsi perangkat lunak dengan cara yang dapat dipahami pengguna.

Selain itu, perancangan antarmuka juga mencakup penataan elemen visual secara sistematis guna memastikan bahwa setiap informasi yang ditampilkan memiliki hierarki

yang jelas. Hal ini sangat krusial agar pengguna dapat membedakan antara informasi utama, seperti hasil rekomendasi rute, dengan informasi pendukung lainnya tanpa mengalami kebingungan kognitif. Dengan pendekatan desain yang berpusat pada pengguna (*user-centered design*), antarmuka diharapkan mampu menjembatani fungsionalitas sistem yang kompleks ke dalam bentuk interaksi yang sederhana, sehingga meningkatkan pengalaman pengguna (*user experience*) secara keseluruhan dalam mengoperasikan perangkat lunak.

Tabel 4. 16 Perancangan Antarmuka - Tampilan Halaman Home

Desain Tampilan Halaman Home	
Deskripsi	
<p>Tampilan antarmuka ini memiliki tata letak horizontal yang sederhana dan terstruktur. Bagian header berada di atas dengan logo <i>Rute.in</i>, menu navigasi, dan tombol <i>Contact-US</i>. Bagian hero terbagi dua kolom, yaitu teks judul, deskripsi singkat, dan tombol <i>cari rute</i> di sisi kiri, serta gambar ilustrasi angkot di sisi kanan. Di bagian bawah, terdapat tiga kartu fitur yang menampilkan informasi singkat mengenai Algoritma A*, Haversine Distance, dan Peta Interaktif. Secara keseluruhan, desain ini menekankan kejelasan navigasi dan kemudahan penggunaan bagi pengguna.</p>	

Gambar 4. 10 Perancangan Antarmuka - Tampilan Halaman Home

Tabel 4. 17 Perancangan Antarmuka - Tampilan Halaman Pencarian

Desain Tampilan Halaman Pencarian			
Deskripsi			
<p>Tampilan ini memiliki struktur vertikal yang sederhana dan informatif. Di bagian atas terdapat header navigasi dengan logo <i>Rute.in</i>, menu <i>Beranda</i> dan <i>Kategori</i>, serta tombol <i>Contact-US</i>. Bagian utama menampilkan judul “<i>Cari Rute Angkot</i>”, deskripsi singkat, dan form pencarian yang terdiri dari input <i>Titik Awal</i>, <i>Titik Tujuan</i>, serta tombol <i>Cari Rute</i>.</p> <p>Pada tampilan hasil, ditampilkan kartu rute angkot berisi ilustrasi angkot, peta rute, informasi titik transit, dan tombol <i>Lihat detail rute</i>. Footer di bagian bawah memuat informasi aplikasi, navigasi, teknologi yang digunakan, serta kontak. Secara keseluruhan, desain ini menekankan kemudahan pencarian rute dan kejelasan informasi hasil.</p>			

Tabel 4. 18 Perancangan Antarmuka - Tampilan Halaman Hasil Pencarian

Desain Tampilan Halaman Hasil Pencarian	
Deskripsi	
<p>Tampilan ini memiliki struktur vertikal yang sederhana dan informatif. Di bagian atas terdapat header navigasi dengan logo <i>Rute.in</i>, menu <i>Beranda</i> dan <i>Kategori</i>, serta tombol <i>Contact-US</i>. Bagian utama menampilkan judul “<i>Cari Rute Angkot</i>”, deskripsi singkat, dan form pencarian yang terdiri dari input <i>Titik Awal</i>, <i>Titik Tujuan</i>, serta tombol <i>Cari Rute</i>.</p> <p>Pada tampilan hasil, ditampilkan kartu rute angkot berisi ilustrasi angkot, peta rute, informasi titik transit, dan tombol <i>Lihat detail rute</i>. Footer di bagian bawah memuat informasi aplikasi, navigasi, teknologi yang digunakan, serta kontak. Secara keseluruhan, desain ini menekankan kemudahan pencarian rute dan kejelasan informasi hasil.</p>	

Tabel 4. 19 Perancangan Antarmuka - Tampilan Halaman Detail

Desain Tampilan Halaman Detail

Rute.in
Beranda Kategori
Contact-US

Terminal Kebon Kelapa - Terminal Cicaheum (Via Binong)
Abdul Mujs (Kebon Kelapa – Cicaheum via Cibinong)

langsung - Tanpa transit t

JAWAK

13.99

km

Perjalanan singkat


TARIF


Rp 14,000

Tarif standar kota

Lanjutkan Perjalanan

Peta Trayek Angkot





Titik yang Dilewati

1. Terminal Kebon Kelapa
2. Jl Dewi Sartika
3. Jl Kautamaan Istri
4. Jl Pungkur
5. Jl Karapitan
6. Jl Buah Batu
7. Jl Banteng
8. Jl Sancang
9. Jl Lodaya
10. Jl Martanegara
11. Jl Turangga
12. Jl Gator Subroto
13. Jl Binong
14. Jl Kiara Condong
15. Stasiun Kiara Condong
16. Jl Jakarta
17. Jl WR Supratman
18. Jl Katamso
19. Jl Pahlawan
20. Jl Cikutra
21. Jl PHH Mustofa (Suci)
22. Terminal Cicaheum

Rute.in

Aplikasi rekomendasi rute angkot Bandung berbasis algoritma A*.

Navigasi

- [Beranda](#)
- [Kategori](#)
- [Cari Rute](#)

Teknologi

- Algoritma A*
- Haversine Formula
- Leaflet Map

Kontak

Bandung, Jawa Barat
support@rutein.id

© 2025 Rute.in — All rights reserved

Gambar 4. 13 Perancangan Antarmuka - Tampilan Halaman Detail

Deskripsi

Deskripsi Tampilan Halaman Detail Rute Angkot

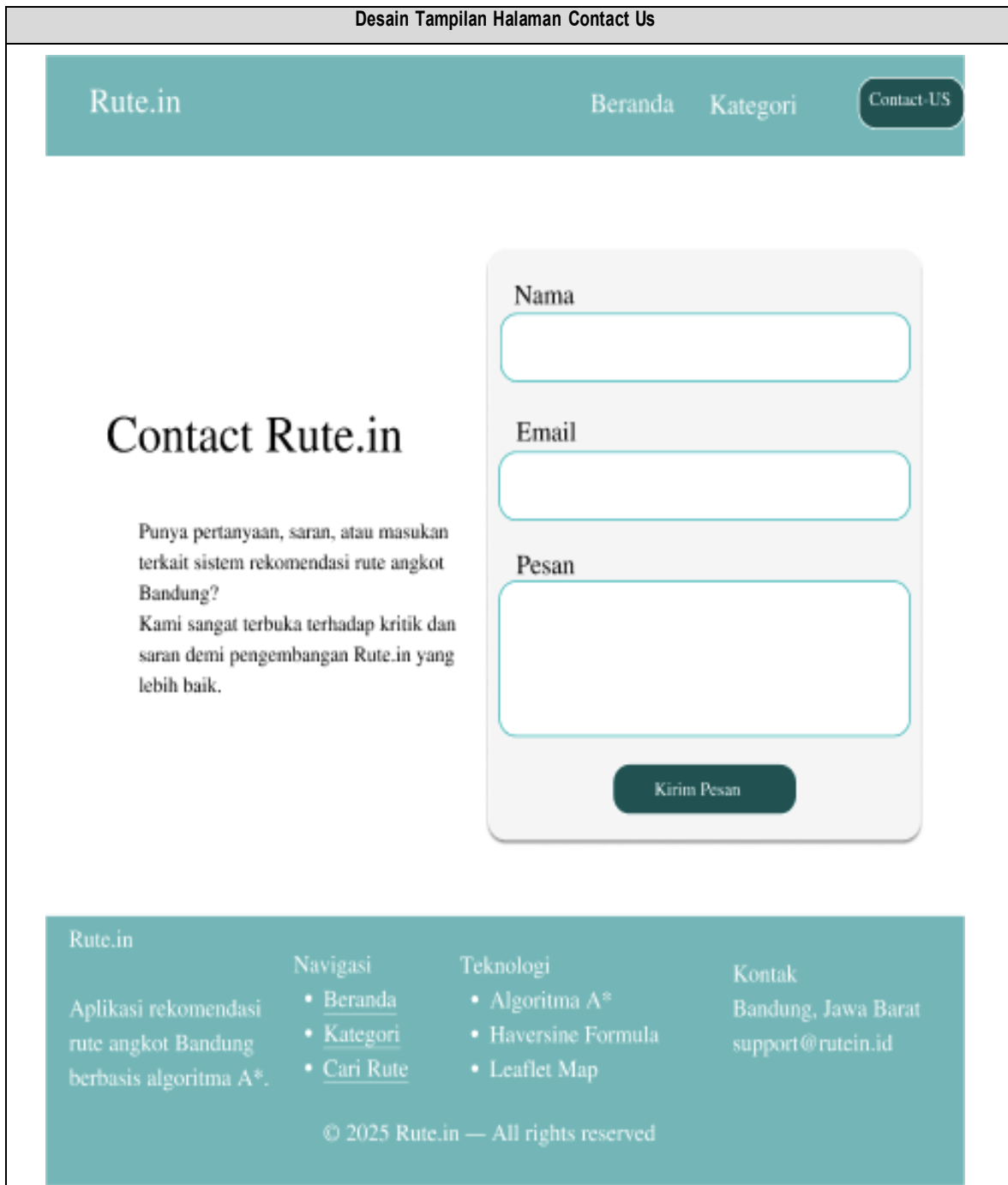
Tampilan antarmuka ini menggunakan tata letak dua kolom yang jelas dan informatif. Bagian kiri menampilkan kartu ringkasan rute yang berisi nama rute angkot, informasi jalur (langsung/tanpa transit), jarak tempuh, estimasi tarif, serta tombol *Lanjutkan Perjalanan*. Di bawahnya ditampilkan ilustrasi angkot dan daftar titik yang dilewati secara berurutan dari titik awal hingga tujuan.

Bagian kanan menampilkan peta trayek angkot interaktif yang memperlihatkan jalur perjalanan beserta penanda titik-titik halte atau persimpangan yang dilewati. Footer di bagian bawah memuat informasi aplikasi, menu navigasi, teknologi yang digunakan, dan kontak. Secara keseluruhan, tampilan ini menekankan kejelasan detail rute, visualisasi jalur pada peta, dan kemudahan pemahaman perjalanan bagi pengguna.

Tabel 4. 20 Perancangan Antarmuka - Tampilan Halaman Kategori

Desain Tampilan Halaman Kategori
Deskripsi
<p>Deskripsi Tampilan Halaman Kategori Angkot</p> <p>Tampilan antarmuka ini disusun dengan layout vertikal yang sederhana dan rapi. Di bagian atas terdapat header navigasi yang memuat logo <i>Rute.in</i>, menu <i>Beranda</i> dan <i>Kategori</i>, serta tombol <i>Contact-US</i>. Bagian utama menampilkan judul “Kategori Angkot” disertai deskripsi singkat dan kolom pencarian untuk mencari trayek atau titik angkot.</p> <p>Di bawahnya ditampilkan daftar angkot dalam bentuk kartu yang tersusun secara horizontal. Setiap kartu berisi ilustrasi angkot dan tombol <i>Lihat detail trayek</i> untuk mengakses informasi lebih lanjut. Footer di bagian bawah memuat informasi aplikasi, menu navigasi, teknologi yang digunakan, serta kontak. Secara keseluruhan, tampilan ini menekankan kemudahan pencarian dan pengelompokan angkot berdasarkan kategori.</p>

Tabel 4. 21 Perancangan Antarmuka - Tampilan Halaman Hubungi Kami



Gambar 4. 15 Perancangan Antarmuka - Tampilan Halaman Hubungi Kami

Deskripsi

Deskripsi Tampilan Halaman Kategori Angkot

Tampilan antarmuka ini disusun dengan layout vertikal yang sederhana dan rapi. Di bagian atas terdapat header navigasi yang memuat logo *Rute.in*, menu *Beranda* dan *Kategori*, serta tombol *Contact-US*. Bagian utama menampilkan judul “Kategori Angkot” disertai deskripsi singkat dan kolom pencarian untuk mencari trayek atau titik angkot.

Di bawahnya ditampilkan daftar angkot dalam bentuk kartu yang tersusun secara horizontal. Setiap kartu berisi ilustrasi angkot dan tombol *Lihat detail trayek* untuk mengakses informasi lebih lanjut. Footer di bagian bawah memuat informasi aplikasi, menu navigasi, teknologi yang digunakan, serta kontak. Secara keseluruhan, tampilan ini menekankan kemudahan pencarian dan pengelompokan angkot berdasarkan kategori.

BAB 5

IMPLEMENTASI DAN PENGUJIAN

5.1 Kontruksi Perangkat Lunak

Konstruksi perangkat lunak merupakan langkah dalam proses rekayasa perangkat lunak yang menekankan pada aktivitas pembuatan atau pembangunan sistem sesuai dengan desain yang telah dihasilkan sebelumnya. Pada langkah ini, semua rencana yang berupa diagram, model data, dan spesifikasi kebutuhan diubah menjadi kode program yang bisa dioperasikan.

5.1.1. Kebutuhan Kakas Perangkat Lunak

Berikut merupakan perangkat lunak yang digunakan dalam pembangunan aplikasi rekomendasi rute angkutan kota berbasis web diuraikan pada tabel dibawah ini.

Tabel 5. 1 Kebutuhan Kakas Perangkat Lunak

No.	Nama Perangkat Lunak	Sumber	Deskripsi
1.	Visual Studio Code	https://code.visualstudio.com/	Editor kode utama untuk menuliskan program aplikasi (Laravel, PHP, Tailwind CSS).
2.	Laragon	https://www.apachefriends.org/	Paket web server lokal (Apache, PHP, MySQL) untuk menjalankan aplikasi pada tahap pengembangan
3.	Laravel Framework	https://laravel.com/	Framework PHP untuk membangun aplikasi web berbasis MVC.
4.	Composer	https://getcomposer.org/	Dependency manager untuk mengelola library laravel
5.	Tailwind CSS	https://tailwindcss.com/	Framework CSS untuk mendesain antarmuka aplikasi agar lebih responsif
6.	MySQL	https://www.mysql.com	Sistem manajemen basis data relasional (RDBMS) untuk menyimpan data rute angkot.
7.	Leaflet	https://leafletjs.com/	Leaflet JS adalah library JavaScript open-source yang digunakan untuk menampilkan peta interaktif pada aplikasi web, seperti menampilkan lokasi, marker, dan jalur/rute pada peta berbasis OpenStreetMap.

5.1.2. Kebutuhan Kakas Perangkat Keras

Berikut merupakan perangkat keras yang digunakan dalam pembangunan aplikasi rekomendasi rute angkutan kota berbasis web diuraikan pada tabel dibawah ini.

Tabel 5. 2 Kebutuhan Kakas Perangkat

No.	Perangkat Keras	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 Aplikasi

Pada langkah ini, semua rencana dan logika sistem yang telah disusun akan diubah menjadi kode program. Proses ini mencakup penulisan *source code* dengan menggunakan bahasa pemrograman yang telah dipilih. Aplikasi dikembangkan dengan bahasa pemrograman PHP menggunakan *framework* Laravel untuk *backend*, sedangkan *framework* Tailwind digunakan untuk *frontend*, dan MySQL berfungsi sebagai Sistem Manajemen Basis Data.

5.2.1. Struktur Aplikasi

Pembangunan aplikasi ini menerapkan pendekatan arsitektur MVC (*Model, View, Controller*) yang membagi tiga bagian fungsional, yakni pengelolaan data pada bagian *Model*, antarmuka pengguna (*user interface*) pada bagian *View*, dan perantara alur logika oleh *Controller*.

5.2.2. Model

Dalam struktur MVC (*Model, View, Controller*) Model bertindak sebagai elemen utama yang berfungsi sebagai gambaran data dan logika bisnis dari aplikasi. Model memiliki tanggung jawab sepenuhnya untuk mengatur data aplikasi. Di dalam laravel, Model diterapkan dengan menggunakan *Eloquent ORM (Object – Relational Mapper)*. File Model umumnya terletak di folder `app/Models`.

Tabel 5. 3 Model

No.	Nama Model	Deskripsi
1.	Angkot.php	Model Angkot berfungsi untuk menggambarkan kendaraan angkot yang berjalan di sistem Rute.in. Model ini menyimpan semua data dasar tentang angkot yang akan ditunjukkan kepada pengguna.

No.	Nama Model	Deskripsi
2.	Rute.php	Model Rute berfungsi sebagai pusat informasi mengenai jalur angkot yang mengaitkan angkot dengan lokasi perjalanan, menjadi landasan untuk pencarian rute, dan dimanfaatkan secara langsung dalam sistem saran.
3.	TitikRute.php	Model TitikRute berperan sebagai simpul graf untuk algoritma A, yang menentukan rute, jarak, dan arah perjalanan. Selain itu, model ini digunakan untuk peta, penghitungan rute, dan validasi pencarian.
4.	Contact.php-	Model Contact berfungsi untuk menyimpan dan mengelola data komunikasi dari pengguna aplikasi Rute.in. Model ini menampung informasi seperti nama, alamat email, subjek, dan pesan yang dikirimkan pengguna melalui fitur kontak atau kritik dan saran. Data yang tersimpan pada model ini digunakan sebagai sarana umpan balik (feedback) untuk pengelola sistem dalam rangka evaluasi, pengembangan, serta peningkatan kualitas layanan aplikasi.

5.2.3. View

View memiliki tanggung jawab untuk menampilkan data dan membuat output HTML yang terlihat oleh pengguna. *View* hanya menerima data yang telah disiapkan dan dikirimkan oleh *Controller* serta berfokus untuk menunjukkan data tersebut tanpa melakukan verifikasi logika bisnis. Laravel menggunakan sistem *template* yang dikenal sebagai blade. Lokasi file *View* terletak di dalam direktori *resources/views*.

Tabel 5. 4 View

No.	Nama View	Deskripsi
1.	beranda.blade.php	Beranda merupakan halaman utama aplikasi Rute.in yang pertama kali diakses pengguna. Halaman ini berfungsi sebagai landing page sekaligus media pengenalan sistem. Di dalamnya terdapat navbar untuk navigasi ke halaman lain seperti Kategori dan Contact Us. Beranda berperan sebagai pintu masuk utama serta sarana branding aplikasi.
2.	cari-rute.blade.php	Halaman cari rute merupakan fitur inti aplikasi untuk melakukan pencarian rute angkot. Pengguna memilih titik awal dan tujuan melalui dropdown yang datanya diambil dari basis data. Setelah form dikirim, sistem menampilkan hasil rekomendasi rute. Halaman ini juga dilengkapi peta interaktif Leaflet untuk memvisualisasikan jalur angkot.
3.	detail.blade.php	Halaman detail digunakan untuk menampilkan informasi lengkap dari satu rute angkot. Informasi yang ditampilkan meliputi nama rute, angkot, tarif, jarak, dan deskripsi perjalanan. Selain data teks, halaman ini menampilkan peta interaktif dengan marker dan polyline. Halaman detail membantu pengguna memahami rute secara menyeluruh.

5.2.4. Controller

Controller adalah komponen utama dalam arsitektur MVC (Model–View–Controller) yang berperan sebagai penghubung antara View dan Model. Controller bertugas menerima permintaan (request) dari pengguna, baik melalui form, klik tombol, maupun URL

tertentu, kemudian memproses permintaan tersebut dengan memanggil Model yang sesuai, dan akhirnya menentukan View apa yang harus ditampilkan beserta data yang dibutuhkan. Dengan kata lain, controller menjadi “otak pengatur alur” aplikasi, memastikan setiap aksi pengguna diproses dengan logika yang benar sebelum hasilnya ditampilkan ke layar.

Tabel 5. 5 Contreller

No.	Nama Controller	Deskripsi
1.	HomeController	HomeController bertugas mengelola halaman beranda aplikasi Rute.in. Controller ini menampilkan tampilan awal yang berisi informasi singkat mengenai sistem dan fitur utama. HomeController berperan sebagai penghubung antara pengguna dan halaman awal aplikasi. Controller ini memanggil view beranda.blade.php.
2	PencarianController	PencarianController menangani proses pencarian rute angkot berdasarkan input pengguna. Controller ini menerima dan memvalidasi data titik awal serta tujuan. Selanjutnya, controller memproses pencarian rute dengan memanggil algoritma A-Star. Hasil pencarian kemudian dikirim ke view untuk ditampilkan kepada pengguna.
3.	RuteController	RuteController berfungsi mengelola data rute angkot dalam sistem. Controller ini menangani penampilan daftar rute, pencarian rute, dan detail rute angkot. RuteController berinteraksi langsung dengan model Rute, TitikRute, dan Angkot. Perannya penting dalam penyajian data rute secara terstruktur.

5.2.5 Services

Dalam arsitektur MVC, Service digunakan sebagai lapisan khusus untuk menyimpan dan menjalankan logika algoritma pencarian rute agar controller tetap fokus pada pengelolaan alur aplikasi. Pada aplikasi Rute.in, Service berperan penting dalam mengimplementasikan algoritma A-Star (A*) yang digunakan untuk menentukan rute angkot terbaik dari titik awal ke titik tujuan.

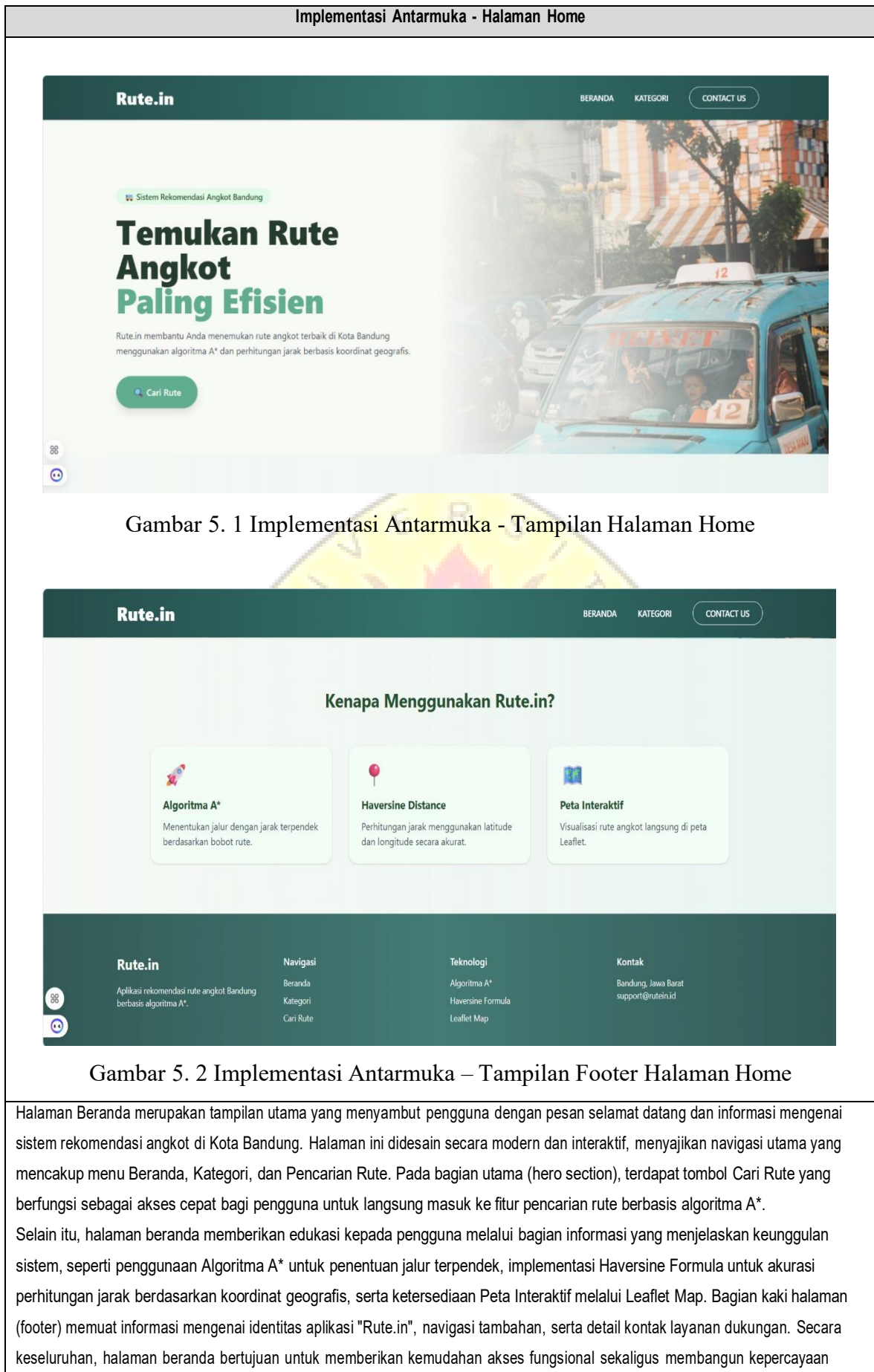
Tabel 5. 6 Services

No.	Nama Model	Deskripsi
1.	A-Star Service	A-Star Service digunakan untuk menyimpan dan menjalankan logika algoritma A* dalam proses pencarian rute angkot. Service ini mengolah data titik rute sebagai node dan hubungan antar titik sebagai edge dalam bentuk graf.
2.	Tarif Service	Tarif Service berfungsi untuk mendukung proses pencarian rute dengan menyediakan informasi biaya perjalanan angkot. Service ini membantu membangun struktur graf dengan mengaitkan titik-titik rute dengan data tarif dan trayek angkot. Informasi tarif digunakan sebagai bagian dari perhitungan perjalanan dalam Algoritma A*

5.3 Implementasi Antarmuka

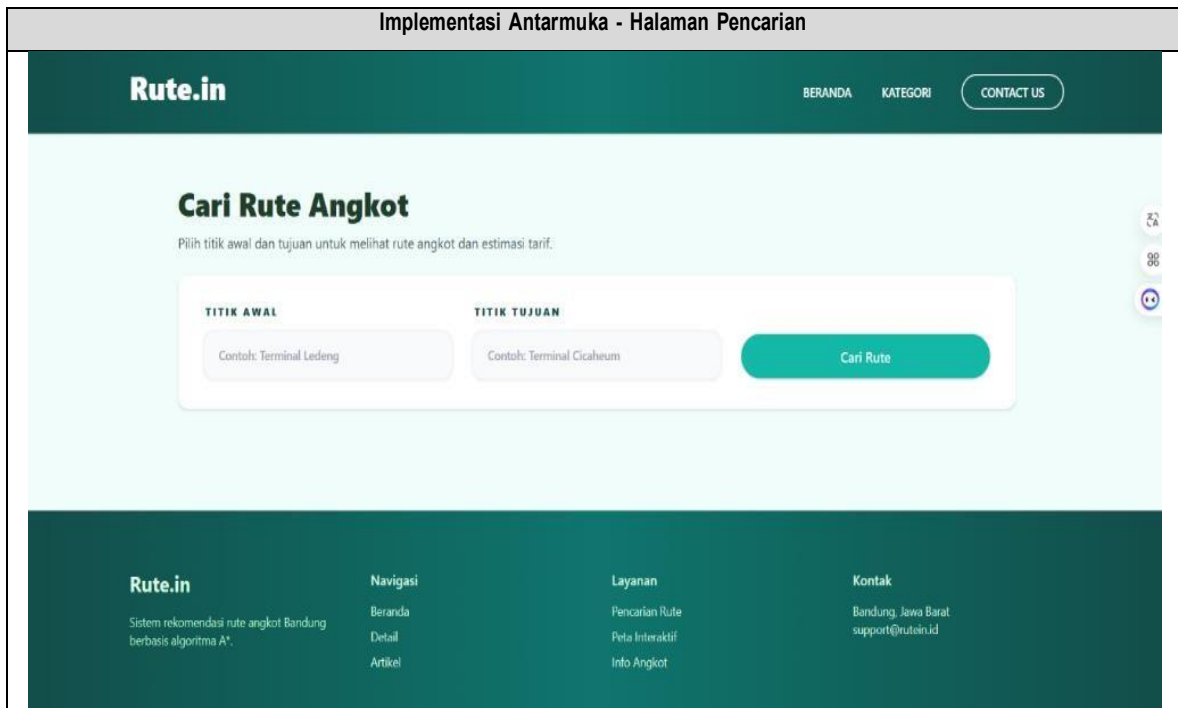
Implementasi antarmuka pengguna ini didasarkan pada rancangan yang telah disusun pada tahap analisis dan perancangan sistem. Antarmuka pengguna dirancang sebagai penghubung antara pengguna dan sistem agar seluruh fungsi aplikasi dapat digunakan secara mudah dan intuitif. Pada tahap implementasi, rancangan tersebut diterjemahkan ke dalam bentuk tampilan aplikasi web yang interaktif, responsif, dan konsisten dengan kebutuhan pengguna.

Tabel 5. 7 Implementasi Antarmuka - Tampilan Halaman Home



pengguna terhadap efisiensi rekomendasi rute yang diberikan oleh sistem.

Tabel 5. 8 Implementasi Antarmuka – Tampilan Halaman Pencarian



Gambar 5. 3 Implementasi Antarmuka - Tampilan Halaman Pencarian

Pada bagian ini, pengguna disediakan dua kolom input yaitu Titik Awal dan Titik Tujuan, disertai tombol Cari Rute untuk memulai proses pencarian. Desain form dibuat ringkas dan jelas agar pengguna dapat langsung memahami alur penggunaan tanpa kebingungan.

Setelah pencarian dilakukan, sistem menampilkan Hasil Rekomendasi Rute dalam bentuk beberapa kartu (card). Setiap kartu memuat ilustrasi angkot, informasi rute perjalanan, estimasi jarak, serta estimasi tarif. Selain itu, pada sisi kanan kartu ditampilkan peta rute interaktif yang menunjukkan jalur perjalanan dari titik awal ke titik tujuan. Pengguna juga dapat menekan tombol Lihat Detail Trayek untuk memperoleh informasi rute yang lebih lengkap.

Pada bagian footer, ditampilkan informasi singkat mengenai aplikasi Rute.in, menu navigasi tambahan, layanan yang tersedia, serta informasi kontak. Susunan ini membantu pengguna mendapatkan informasi pendukung tanpa mengganggu fokus utama pada pencarian rute. Secara keseluruhan, tampilan antarmuka ini dirancang untuk mendukung sistem rekomendasi rute angkot berbasis algoritma A* secara intuitif, informatif, dan mudah digunakan.

Tabel 5. 9 Implementasi Antarmuka - Tampilan Halaman Hasil Pencarian

Implementasi Antarmuka – Tampilan Halaman Hasil Pencarian

Rute.in BRANDA KATEGORI CONTACT US

Cari Rute Angkot
Pilih titik awal dan tujuan untuk melihat rute angkot dan estimasi tarif.
Anda berada di titik Terminal Kebon Kelapa dan ingin menuju Ji Pungkur

TITIK AWAL: Contoh: Terminal Ledeng
TITIK TUJUAN: Contoh: Terminal Cicaheum
Cari Rute

Hasil Rekomendasi

- Terminal Kebon Kelapa - Terminal Cicaheum (Via Binong)**
 Angkot: Abdul Muis (Kebon Kelapa - Cicaheum via Binong)
 Estimasi Jarak: 0,84 km
 Estimasi Tarif: Rp 3,000
 Lihat Detail Trayek
- Terminal Kebon Kelapa - Terminal Cicaheum (Via Aceh)**
 Angkot: Abdul Muis (Kebon Kelapa - Cicaheum via Aceh)
 Estimasi Jarak: 0,89 km
 Estimasi Tarif: Rp 3,000
 Lihat Detail Trayek
- Terminal Kebon Kelapa - Dago**
 Angkot: Abdul Muis (Kebon Kelapa - Dago)
 Estimasi Jarak: 0,89 km
 Estimasi Tarif: Rp 3,000
 Lihat Detail Trayek
- Terminal Kebon Kelapa - Terminal Ledeng**
 Angkot: Abdul Muis (Kebon Kelapa - Ledeng)
 Estimasi Jarak: 0,89 km
 Estimasi Tarif: Rp 3,000
 Lihat Detail Trayek

Rute.in
Sistem rekomendasi rute angkot Bandung berbasis algoritma A*.

Navigasi
Beranda
Detail
Artikel

Layanan
Pencarian Rute
Peta Interaktif
Info Angkot

Kontak
Bandung, Jawa Barat
support@ruteind.com

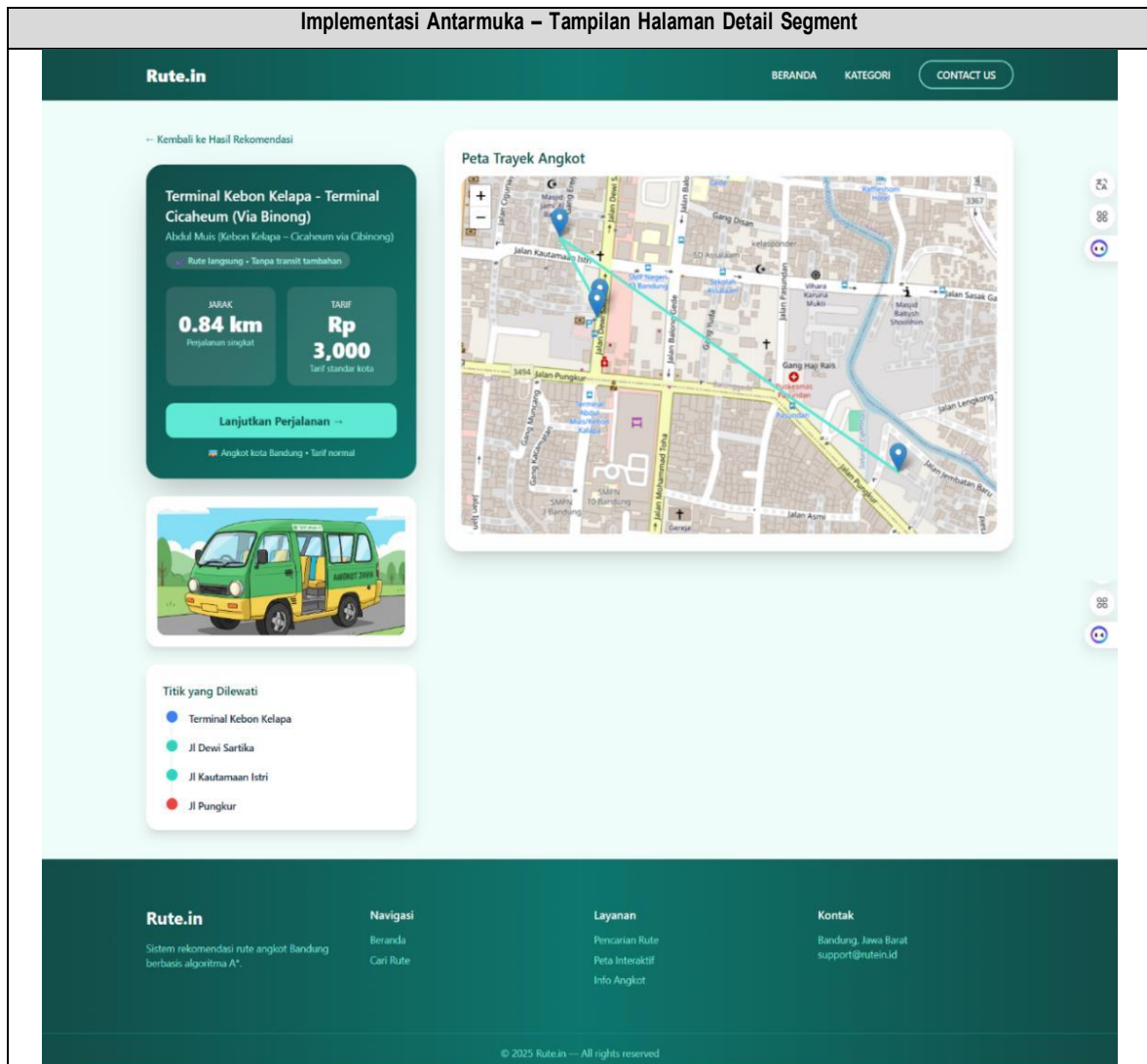
© 2025 Rute.in - All rights reserved

Gambar 5. 4 Implementasi Antarmuka - Tampilan Halaman Hasil Pencarian

Setelah pencarian dilakukan, sistem menampilkan Hasil Rekomendasi Rute dalam bentuk beberapa kartu (card). Setiap kartu memuat ilustrasi angkot, informasi rute perjalanan, estimasi jarak, serta estimasi tarif. Selain itu, pada sisi kanan kartu ditampilkan peta rute interaktif yang menunjukkan jalur perjalanan dari titik awal ke titik tujuan. Pengguna juga dapat menekan tombol Lihat Detail Trayek untuk memperoleh informasi rute yang lebih lengkap.

Pada bagian footer, ditampilkan informasi singkat mengenai aplikasi Rute.in, menu navigasi tambahan, layanan yang tersedia, serta informasi kontak. Susunan ini membantu pengguna mendapatkan informasi pendukung tanpa mengganggu fokus utama pada pencarian rute. Secara keseluruhan, tampilan antarmuka ini dirancang untuk mendukung sistem rekomendasi rute angkot berbasis algoritma A* secara intuitif, informatif, dan mudah digunakan.

Tabel 5. 10 Implementasi Antarmuka – Tampilan Halaman Detail Segment



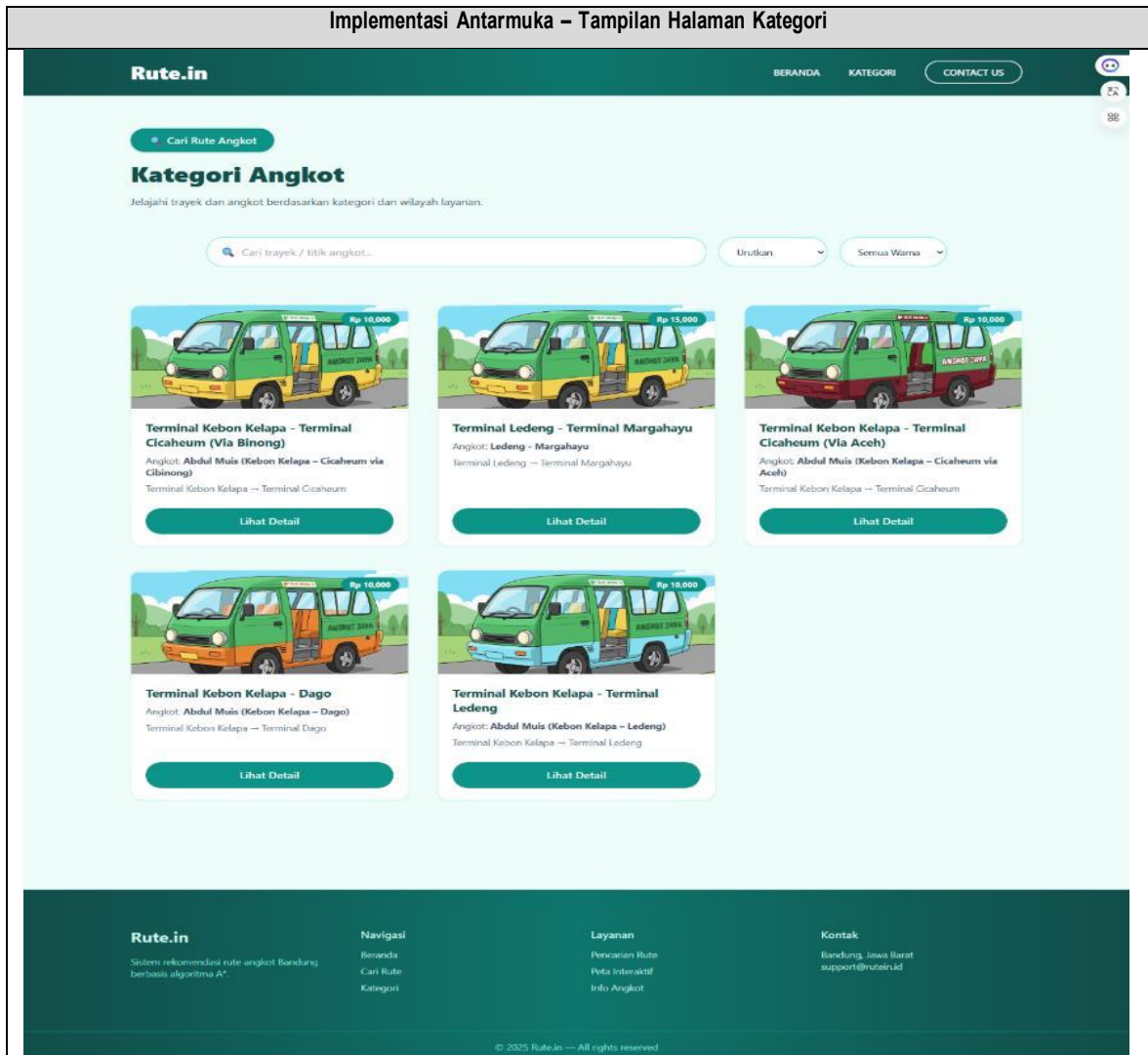
Gambar 5. 5 Implementasi Antarmuka - Tampilan Halaman Detail Segment

Halaman Detail Rute Per Segmen merupakan antarmuka yang menyajikan rincian hasil pencarian rute spesifik dari titik asal ke titik tujuan yang diinput oleh pengguna. Halaman ini mengintegrasikan data dinamis hasil perhitungan algoritma A* ke dalam komponen visual yang informatif. Di sisi kiri, panel rute menampilkan nama trayek yang direkomendasikan beserta estimasi jarak tempuh yang akurat (misalnya 0.84 km) dan estimasi tarif perjalanan (misalnya Rp 3.000). Panel ini juga memuat indikator "Rute langsung" untuk menginformasikan apakah pengguna perlu melakukan transit atau tidak.

Pada bagian kanan, Peta Trayek Angkot interaktif menyajikan visualisasi geografis dari segmen perjalanan yang dicari. Berbeda dengan halaman detail keseluruhan, peta pada halaman ini hanya memfokuskan tampilan pada jalur yang relevan dengan pencarian pengguna, lengkap dengan *marker* biru yang menandai titik-titik henti strategis dan garis *polyline* yang menghubungkan rute tersebut secara presisi.

Di bawah panel informasi utama, terdapat komponen Titik yang Dilewati. Bagian ini menjabarkan urutan lokasi spesifik yang akan dilalui pengguna, mulai dari Terminal asal hingga titik tujuan akhir. Penyajian daftar ini berfungsi sebagai panduan navigasi tekstual yang melengkapi visualisasi peta, sehingga pengguna dapat memverifikasi setiap ruas jalan yang akan dilewati dalam perjalanan mereka. Keseluruhan desain antarmuka ini bertujuan untuk memvalidasi hasil perhitungan algoritma A* menjadi instruksi perjalanan yang intuitif dan mudah dipahami oleh pengguna aplikasi Rute.in.

Tabel 5. 11 Implementasi Antarmuka – Tampilan Halaman Kategori

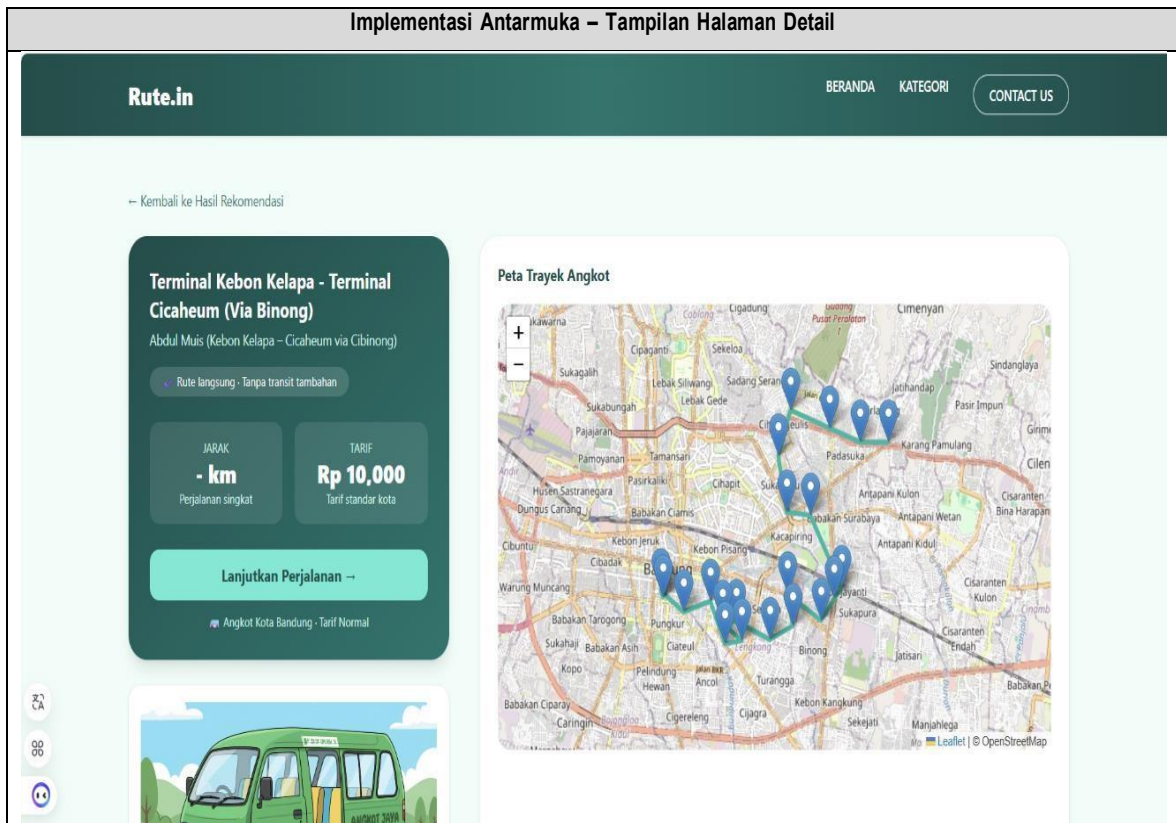


Gambar 5. 6 Implementasi Antarmuka - Tampilan Halaman Kategori

Halaman Kategori Angkot menyajikan daftar trayek angkot yang tersedia di Kota Bandung dalam format kartu informasi (*card*) yang disusun secara sistematis. Pada bagian atas halaman, terdapat fitur interaktif berupa kolom pencarian yang memungkinkan pengguna mencari trayek berdasarkan nama jalan atau terminal, serta menu *drop-down* untuk mengurutkan data dan menyaring trayek berdasarkan kategori warna angkot. Setiap kartu informasi memuat elemen visual berupa ilustrasi angkot yang sesuai dengan identitas aslinya, nama rute utama (seperti Terminal Kebon Kelapa – Terminal Cicaheum), informasi titik-titik yang dilewati, serta label harga tarif standar kota yang ditampilkan pada pojok kanan atas ilustrasi.

Selain itu, setiap kartu dilengkapi dengan tombol *Lihat Detail* berwarna hijau yang berfungsi sebagai akses utama bagi pengguna untuk mendapatkan informasi lebih mendalam mengenai rute tersebut secara keseluruhan. Struktur navigasi pada halaman ini tetap menjaga konsistensi dengan menyertakan *header* yang memuat logo "Rute.in" dan menu utama (Beranda, Kategori, Contact Us), serta *footer* yang berisi detail teknologi yang digunakan seperti Algoritma A* dan Leaflet Map. Halaman ini dirancang untuk memudahkan pengguna dalam menjelajahi berbagai pilihan transportasi umum secara intuitif, sehingga mendukung proses pengambilan keputusan perjalanan yang lebih efisien sebelum pengguna masuk ke tahap pencarian rute yang lebih spesifik.

Tabel 5. 12 Implementasi Antarmuka – Tampilan Halaman Detail

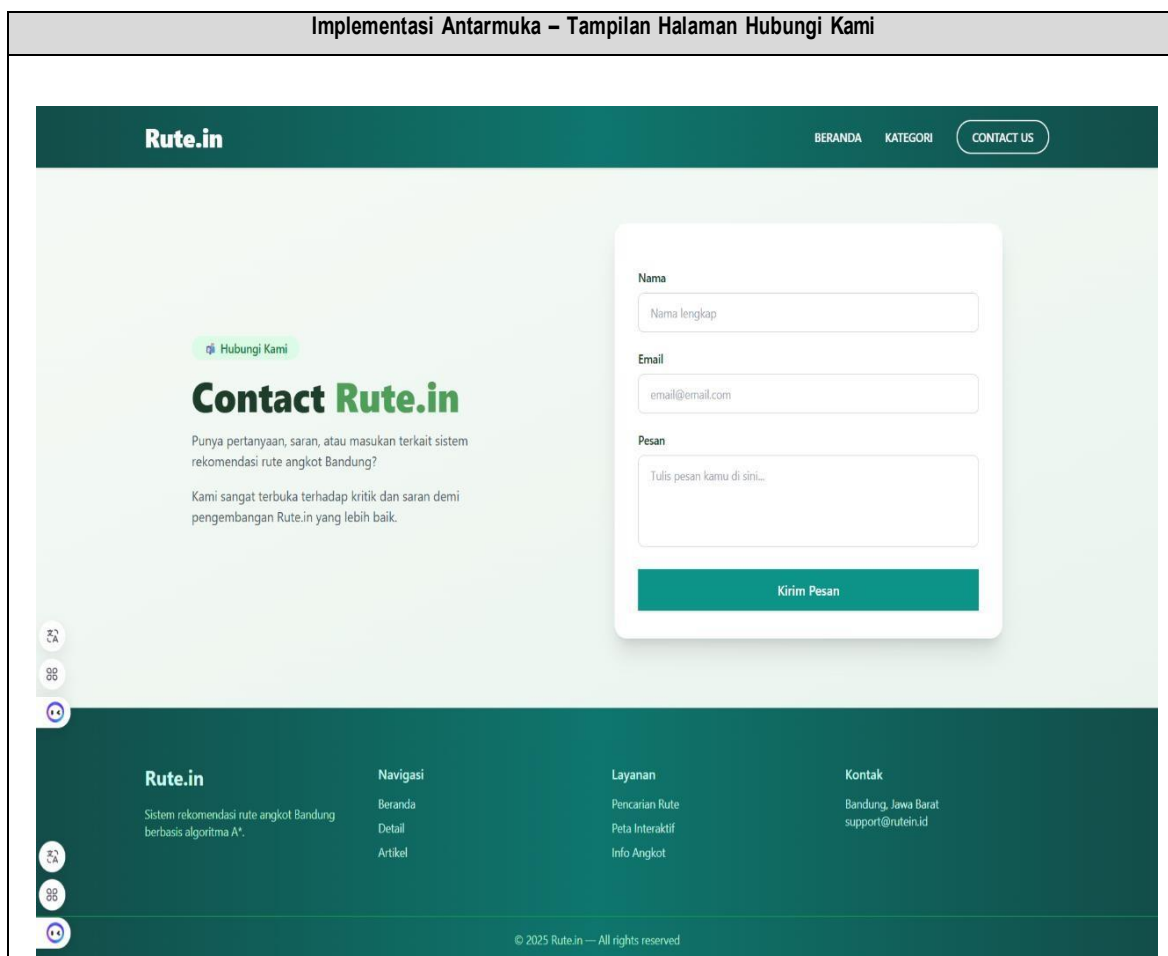


Gambar 5. 7 Implementasi Antarmuka - Tampilan Halaman Detail Trayek

Halaman Detail Trayek menyajikan informasi komprehensif mengenai satu rute angkot secara utuh, bukan per segmen perjalanan. Pada bagian kiri, terdapat panel informasi utama yang menampilkan nama terminal asal dan tujuan, nama populer trayek, serta label status rute seperti "Rute Langsung". Panel ini juga memuat data statistik penting seperti estimasi total jarak tempuh dalam kilometer dan informasi tarif standar kota yang berlaku. Terdapat tombol navigasi interaktif untuk melanjutkan pencarian atau melihat jadwal operasional.

Pada sisi kanan, halaman ini menyediakan fitur Peta Trayek Angkot interaktif berbasis *Leaflet Map* yang memvisualisasikan seluruh lintasan angkot tersebut. Peta ini menampilkan garis rute (*polyline*) berwarna hijau yang menghubungkan puluhan *marker* biru yang merepresentasikan setiap titik henti atau halte yang dilewati oleh trayek tersebut dari awal hingga akhir. Visualisasi ini sangat membantu pengguna dalam memahami cakupan wilayah yang dilalui oleh angkot yang dipilih secara sistematis. Selain itu, terdapat elemen ilustrasi visual angkot pada bagian bawah panel untuk memberikan identitas visual yang jelas sesuai dengan warna asli angkot di lapangan.

Tabel 5. 13 Implementasi Antarmuka – Tampilan Halaman Hubungi Kami



Gambar 5. 8 Implementasi Antarmuka - Tampilan Halaman Hubungi Kami

Halaman Hubungi Kami dirancang sebagai saluran komunikasi interaktif yang memungkinkan pengguna untuk memberikan pertanyaan, saran, maupun masukan terkait sistem rekomendasi rute angkot. Secara visual, halaman ini terbagi menjadi dua bagian utama yang seimbang untuk menjaga fokus pengguna. Pada sisi kiri, terdapat elemen teks yang menyambut pengguna dengan judul "Contact Rute.in" disertai penjelasan singkat mengenai keterbukaan pengembang terhadap kritik dan saran demi pengembangan sistem yang lebih baik di masa depan.

Pada sisi kanan, tersedia sebuah formulir kontak (*contact form*) yang minimalis dan intuitif, yang terdiri dari tiga kolom input utama: Nama, Email, dan Pesan. Pengguna dapat mengisi identitas diri dan rincian pesan secara terstruktur sebelum menekan tombol "Kirim Pesan" yang terletak di bagian bawah formulir. Desain formulir ini mengutamakan kemudahan penggunaan (*usability*) agar setiap masukan dari pengguna dapat tersampaikan dengan cepat secara sistematis.

Konsistensi navigasi tetap dipertahankan melalui *header* yang memuat menu utama serta *footer* yang menyajikan informasi kontak teknis seperti alamat di Bandung dan alamat email resmi dukungan (support@rutein.id). Halaman ini memiliki peran penting dalam aspek evaluasi sistem, di mana masukan yang diterima melalui antarmuka ini dapat dijadikan sebagai dasar untuk menilai serta meningkatkan kualitas fitur-fitur pada aplikasi Rute.in secara berkelanjutan.

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, penguji 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. 14 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" tetapi memiliki dua jalur / 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

Kode Test Skenario	Test Skenario	Kode Test Case	Test Case	Test Data	Expected Result	Actual Result	Pass/Fail
		TC01 – 5	Pengguna menginputkan titik awal dan titik tujuan berdasarkan nama “jalan”.	Titik awal : Jl. Setiabudi Titik Tujuan; Jl. Ahmad Yani	Sistem menampilkan card hasil dengan informasi angkot yang sesuai dengan nama jalan.	Sistem menampilkan card hasil dengan informasi angkot yang sesuai dengan nama jalan.	Pass
		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

Kode Test Skenario	Test Skenario	Kode Test Case	Test Case	Test Data	Expected Result	Actual Result	Pass/Fail
		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 penelitian, pembangunan, serta pengujian yang telah dilakukan terhadap sistem rekomendasi rute angkutan kota, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi algoritma A* (A-Star) dalam sistem ini telah berfungsi dengan baik dalam menentukan rute terpendek melalui perhitungan fungsi evaluasi $f(n)$ yang menggabungkan jarak tempuh nyata (g) dan estimasi jarak udara (h).
2. Sistem mampu memberikan rekomendasi rute yang efisien bagi pengguna dengan menyajikan data pendukung yang lengkap, meliputi visualisasi lintasan pada peta interaktif, estimasi total jarak tempuh, hingga informasi tarif perjalanan.
3. Aplikasi ini memiliki tingkat aksesibilitas yang tinggi karena dapat dioperasikan secara langsung oleh masyarakat tanpa melalui prosedur pendaftaran akun, sehingga memudahkan pencarian informasi rute secara cepat dan praktis.
4. Berdasarkan seluruh pengujian fungsionalitas yang telah dilakukan, aplikasi rute.in dinyatakan layak dan mampu membantu pengguna dalam menentukan pilihan transportasi angkutan kota yang paling optimal di wilayah operasionalnya.

6.2. Saran

Sebagai upaya untuk meningkatkan kualitas dan fungsionalitas aplikasi di masa yang akan datang, maka terdapat beberapa saran pengembangan yang diajukan sebagai berikut:

1. Pengembangan penelitian selanjutnya disarankan untuk mengoptimalkan algoritma agar mampu menangani skenario pencarian rute multi-transit (2 hingga 3 kali perpindahan). Hal ini mencakup pengembangan fungsi evaluasi yang tidak hanya menghitung jarak tempuh, tetapi juga mempertimbangkan titik temu (simpang) antar trayek yang berbeda sebagai simpul (*node*) transit guna menghasilkan kombinasi rute yang lebih dinamis.
2. Sistem dapat dikembangkan lebih lanjut dengan mengintegrasikan data kondisi lalu lintas secara aktual, sehingga parameter penentuan rute tidak hanya terpaku pada jarak terpendek tetapi juga mempertimbangkan waktu tempuh yang paling singkat.
3. Diharapkan adanya perluasan database trayek yang mencakup seluruh wilayah Bandung Raya, termasuk wilayah kabupaten dan kota satelit lainnya, guna memberikan jangkauan informasi yang lebih luas bagi para penumpang.

DAFTAR PUSTAKA

- [ACH20] A. Jauhari, Devie R.A., F.A.M., ANALISIS DAN DESAIN SISTEM INFORMASI: Pendekatan Pengembangan System, Media Nusa Creative, 2020.
- [ADH24] W. L. Adhtiya Prayogi, "Rancang Bangun Sistem Informasi Pemesanan Tiket Event Asian Matsuri Palangkaraya Berbasis Web," *JOINTECOMS(Journal of Information Technology and Computer Science)*, vol. 4, no. 1, pp. 29-37, March 2024.
- [AGU24] A. Kurniawan, *Laravel Workshop: Belajar dengan Membangun Proyek Aplikasi Web*, Ilmu Data Publisher, 2024.
- [ALT24] Al Tafakur L.O., Anriani S., A.I.S., B.A.A., Syukuriah, Ahmad Y.Y., Syaiful, Rafilus S., R.M.R., *Transportasi Publik*, E. Ari Kusuma, Ed., Makassar: CV. Tohar Media, 2024, p. 18.
- [AMA24] A. Hanifa, M. Eka, Chaeron N., Novia H., Rozzi K.D., Aminah, *RPL (Rekayasa Perangkat Lunak)*, S. M. C. J. Prayoga, Ed., Serasi Media Teknologi, 2024, p. viii + 174.
- [BON25] Bonda S., Apriyanto, Putu M.P., M. Nur, I Made Dwi A., Abdul Rezha E.N., Haryono, *BUKU AJAR PEMROGRAMAN WEB*, Sepriano, Ed., Jambi: PT. Sonpedia Publishing Indonesia, 2025.
- [RAD20] Raden, Budiarto H., *Rekayasa Perangkat Lunak*, RBH, 2020.
- [CHA21] A. M. Chairul Insani Ilham, *Teknik Pemilihan Rute dan Moda Angkutan Perairan Daratan (Teori dan Implementasi)*, Kodri, Ed., Indramayu: CV. Adanu Abimata, 2021, p. 1.
- [DAL18] Dalem, Ida B.G., Wahyu A., "PENERAPAN ALGORITMA A* (STAR) MENGGUNAKAN GRAPH UNTUK MENGHITUNG JARAK TERPENDEK," *JURNAL RESISTOR*, vol. 1, no. 1, pp. 41-47, April 2018.
- [DRK21] S. M. C. Dr. Kadarudin, *KATALOG DALAM TERBITAN (KDT) Perpustakaan Nasional Republik Indonesia Penelitian Bidang Ilmu Hukum (Sebuah Pemahaman Awal)*, Semarang: Formaci, 2021.
- [EKK21] A. P. Ekkal Prasetyo, "Implementasi Waterfall Model Dalam Pengembangan Sistem Informasi Eksekutif Penduduk," *Journal of Information Systems and Informatics*, vol. 3, no. 1, pp. 213-224, March 2021.
- [ELV23] E. Y. N. H. Elvi Rahmi, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review," *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 7, no. 1, pp. 1-10, Januari 2023.
- [HAR21] Harming P.K., Windu G., Siti N., Andi J.S.R., Deviana S.W., "ANALISA PENCARIAN RUTE TERCEPAT MENUJU TEMPAT WISATA PULAU KUMALA KOTA TENGGARONG MENGGUNAKAN ALGORITMA

- GREEDY," *Jurnal Ilmiah Ilmu Komputer*, vol. 7, no. 1, pp. 9-15, April 2021.
- [LAB21] L. S. Labuan Nababan, "IMPLEMENTASI METODE ALGORITMA C4.5 DALAM PENENTUAN TARIF ANGKOT DI KOTA MEDAN," *Jurnal Sistem Informasi Kaputama (JSIK)*, vol. 5, no. 2, pp. 106-114, Juli 2021.
- [NAJ24] N. Umar, Hermawan S., Hariadi Y., Prilian A.M., Marcello S., Ir. Samuel Y..P., Fegie Y.W., BUKU AJAR REKAYASA PERANGKAT LUNAK, Efitra, Ed., PT. Sonpedia Publishing Indonesia, 2024.
- [NAZ22] Nazaruddin A., Erly K., F.G.J.R., Anita M., Nur Syamsiah, Kraugusteeliana, B.D..C., Yesi S., Titus K., Irwanto, Guntoro, ANALISA & PERANCANGAN SISTEM INFORMASI BERORIENTASI OBJEK, N. Rismawati, Ed., Kabupaten Bandung: Widina Media Utama, 2022.
- [NUR08] Nursalam, KONSEP DAN PENERAPAN METODOLOGI PENELITIAN ILMU KEPERAWATAN, Jakarta: Ariyanto, Edward Tanujaya, 2008.
- [RAM21] M. Ramdhan, METODE PENELITIAN, Surabaya: Cipta Media Nusantara (CMN), 2021.
- [RON20] A. S. Roni Habibi, APLIKASI PREDIKSI JUMLAH KEBUTUHAN PERUSAHAAN BUKU TUTORIAL, A. S. Roni Habibi, Ed., Bandung: Kreatif Industri Nusantara, 2020.
- [RON19] Roni H., D. Irga B. Naufal F., F.S Damayanti, Penggunaan Framework Laravel Untuk Membuat Aplikasi Absensi Terintegrasi Mobile, D. I. B. N. F. F. S. D. Roni Habibi, Ed., Bandung: Kreatif Industri Nusantara, 2019.
- [YUD19] H. A. P. Yudho Yudhanto, Mudah Menguasai Framework Laravel, Jakarta: PT Elex Media Komputindo, 2019.
- [YUH09] M. R. H. Yuhefizar, Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla Edisi Revisi, Jakarta: PT Elex Media Komputindo.
- [ROL19] R. M. Awangga, Pengantar Sistem Informasi Geografis, Bandung: Kreatif Industri Nusantara 2019.
- [CHA23] C. T. Setyobudhi, B.Eng., M.T., H. Novindaning, Ed., Y. S. Milah BAHAN AJAR ALGORITMA DAN PEMROGRAMAN, Bandung: CV. Mega Press Nusantara, Mei 2022

LAMPIRAN



Berita Acara

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

Nama : Rika Febriyanti E. Wenda
NRP : 213040125
Fakultas / Prodi : Teknik / Teknik Informatika
Universitas : Universitas Pasundan

Telah melakukan wawancara dengan :

Nama : Anto
Usia : 27 tahun
Profesi : Supir Angkot

Guna keperluan penelitian untuk penyusunan tugas akhir dengan judul Pembangunan Aplikasi Rekomendasi Rute Angkutan Kota menggunakan Algoritma A* Berbasis Laravel. Demikian berita acara ini dibuat untuk digunakan sebagaimana mestinya.

Bandung, 16 Juli 2025

Narasumber



(Anto)

Pewawancara



(Rika Febriyanti E. Wenda)

Pengumpulan Data dan Wawancara

Trayek Angkutan Kota Bandung yang Mengalami Perubahan

1. Kode Trayek 03 (Abdul Muis – Ledeng)
2. Kode Trayek 05 (Cicaheum – Ledeng)
3. Kode Trayek 06 (Cicaheum – Ciroyom)
4. Kode Trayek 11.A (St. Hall – Ciumbuleuit via Eyckman)
5. Kode Trayek 11.B (St. Hall – Ciumbuleuit via Cihampelas)
6. Kode Trayek 15 (Margahayu Raya – Ledeng)
7. Kode Trayek 19.A (Ciroyom – Sarijadi via Sukajadi)
8. Kode Trayek 19.B (Ciroyom – Sarijadi via Setrasari Mall)
9. Kode Trayek 22 (Sederhana – Cipagalo)
10. Kode Trayek 24 (Sederhana – Cimindi)
11. Kode Trayek 35 (Cibaduyut – Karang Setra)

*Jalan Wastukencana Nomor 2 Telp. (022) 432338-432369-432370 Bandung,
Provinsi Jawa Barat*

Lampiran 1 Data Trayek Angkutan Kota Bandung

Kuesioner Penumpang Angkot

Halo semuanya!

Saya mohon izin untuk meminta sedikit waktu Anda untuk mengisi formulir kuesioner penumpang angkot.

Formulir ini merupakan bagian dari penelitian tugas akhir dengan tema tentang angkutan kota yang bertujuan untuk mengetahui sejauh mana masyarakat mengetahui informasi seputar angkutan kota (angkot), seperti rute, tarif, warna angkot, trayek, dan informasi terkait lainnya.

Partisipasi Anda sangat berarti dalam upaya mengembangkan sistem informasi transportasi publik yang lebih baik di masa depan. Seluruh jawaban bersifat anonim dan hanya akan digunakan untuk keperluan penelitian.

Terima kasih banyak atas waktu dan kesediaan Anda!

1. Jenis Kelamin

2. Usia

3. Domisili

4. Apakah anda menggunakan angkota kota(angkot)

Lampiran 2 Kuesioner Penumpang Angkot

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)

5. Apakah anda sering anda menggunakan angkot sebagai transportasi umum ?

Tandai satu oval saja.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

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

Tandai satu oval saja.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

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

Tandai satu oval saja.

- Ya
 Tidak

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

Tandai satu oval saja.

- Ya
 Tidak

Lampiran 3 Kuesioner Penumpang Angkot

9. Darimana biasanya anda mendapatkan informasi tentang angkot

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

Tandai satu oval saja.

Ya

Tidak

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

Centang semua yang sesuai.

jadwal operasional

jalur lengkap

warna kendaraan

Gambar

Tarif

Peta visual

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

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

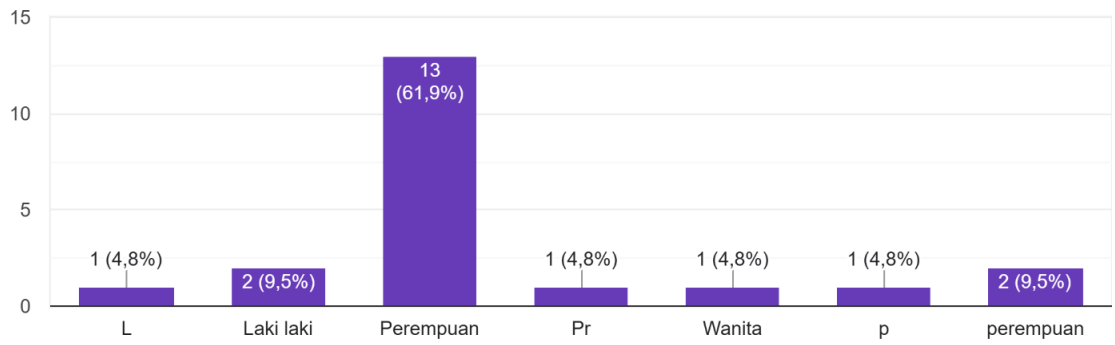
Konten ini tidak dibuat atau didukung oleh Google.

Google Formulir

Lampiran 4 Kuesioner Penumpang Angkot

Jenis Kelamin

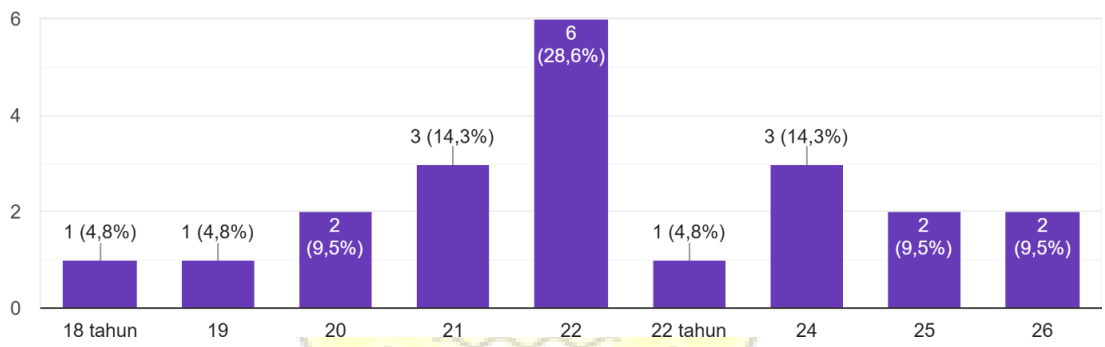
21 jawaban



Lampiran 5 Persentase Jenis Kelamin Responden

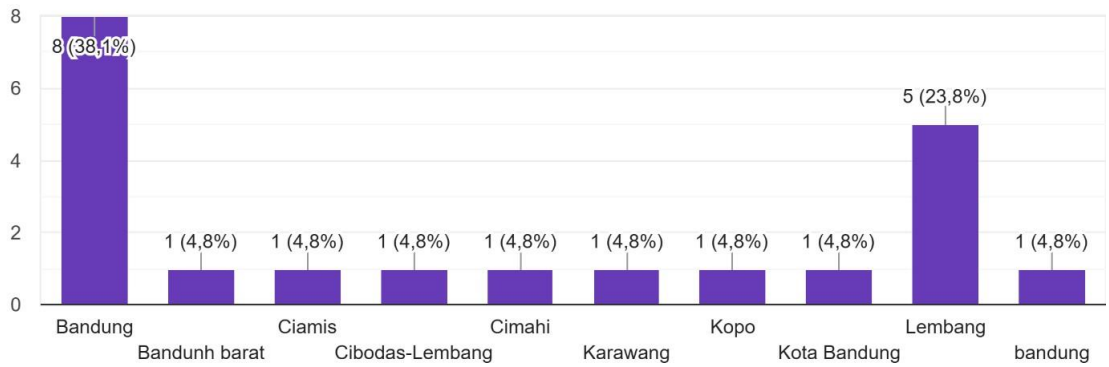
Usia

21 jawaban



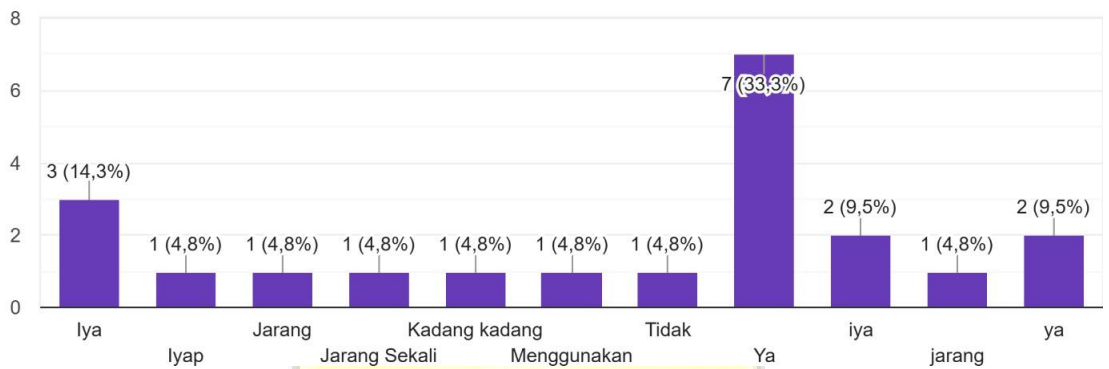
Lampiran 6 Persentase Usia Responden

Domisili
21 jawaban



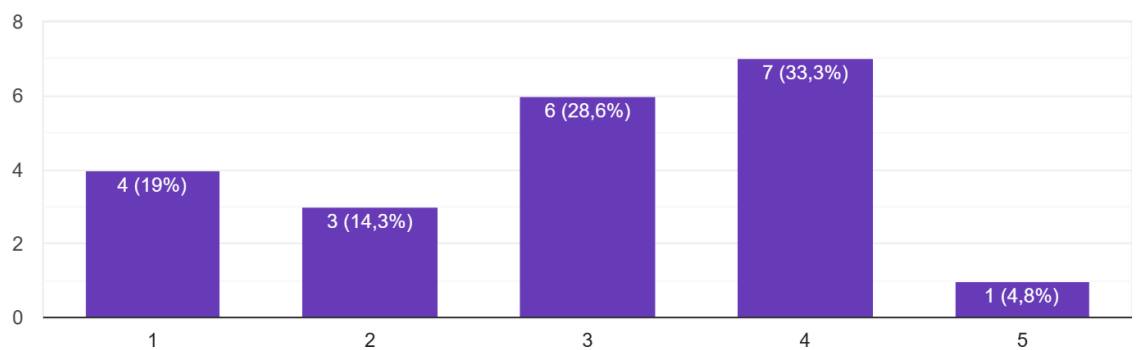
Lampiran 7 Persentase Domisili Responden

Apakah anda menggunakan angkota kota(angkot)
21 jawaban



Lampiran 8 Persentase Penggunaan Angkot

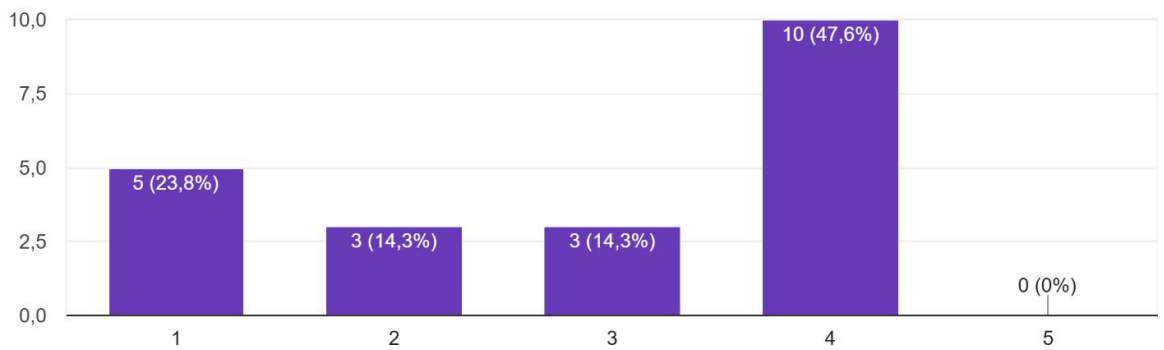
Apakah anda sering anda menggunakan angkot sebagai transportasi umum ?
21 jawaban



Lampiran 9 Persentase Penggunaan Angkot

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

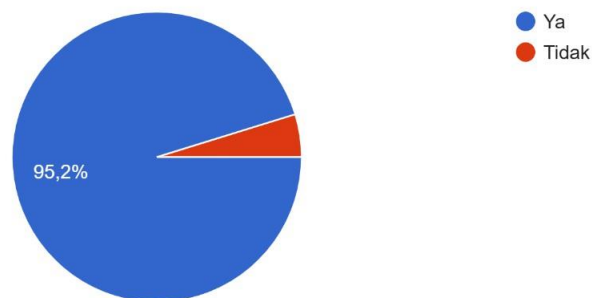
21 jawaban



Lampiran 10 Persentase Pencarian Informasi

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

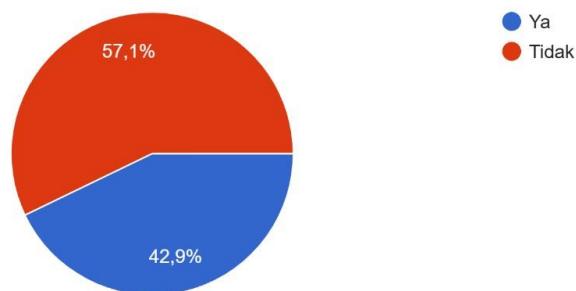
21 jawaban



Lampiran 11 Persentase Informasi Warna Angkot

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

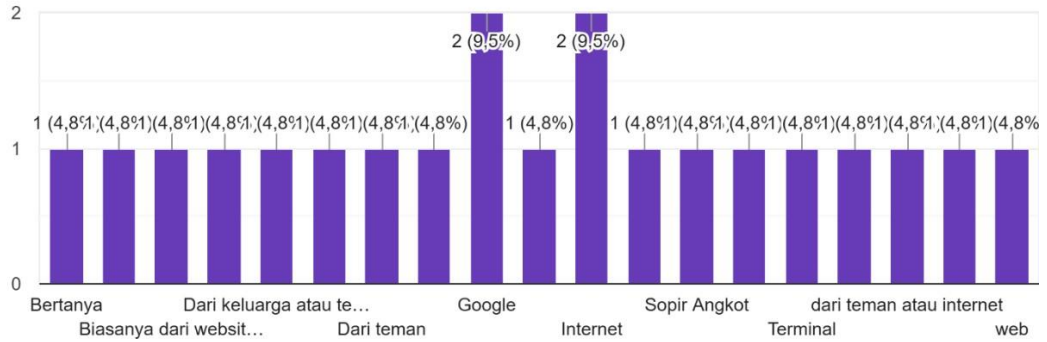
21 jawaban



Lampiran 12 Persentase Mengenai Detail Informasi

Darimana biasanya anda mendapatkan informasi tentang angkot

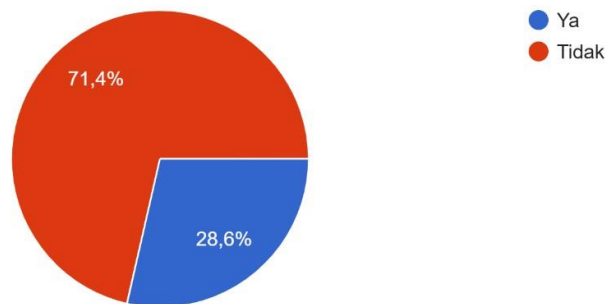
21 jawaban



Lampiran 13 Persentase Sumber Informasi Angkot

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

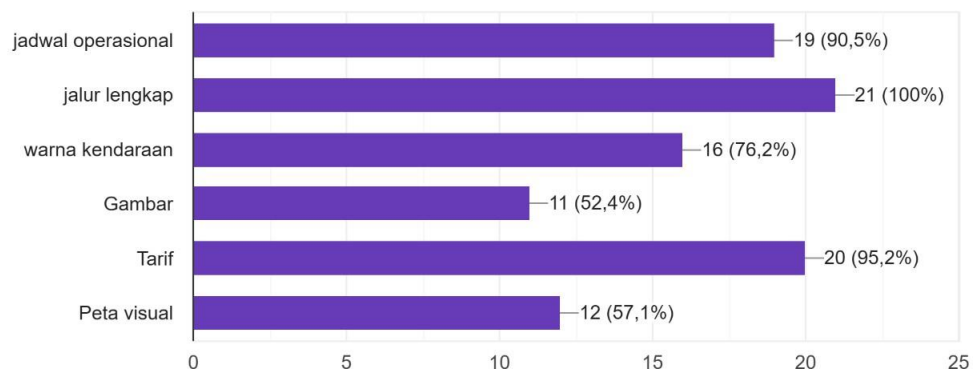
21 jawaban



Lampiran 14 Persentase Keakuratan Informasi Angkot

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

21 jawaban



Lampiran 15 Persentase Saran Responden

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

Untuk informasi tarif angkot setiap rutenya lebih jelas

promosikan aplikasinya menggunakan sosmed

Semoga fasilitas umum bisa lebih lengkap lagi informasi tarif dan rute nya

Bisa dibuatkan website khusus informasi lengkap terkait angkot untuk lebih memudahkan

bisa dibuat aplikasi yang menunjang akses informasi mengenai angkot

Sarannya agar jalur lengkap warna kendaraan maupun tarif lebih difokuskan

Tersedianya website khusus yang berisi rute angkot dan informasi mengenai bagaimana rute angkot yang harus dinaiki apabila ingin menuju ke tempat tertentu secara langsung

ada sebuah pihak yang memberikan informasi mengenai angkot baik itu di sosmed atau pun web karena saat ini kurang jelas buat informasi nya

Lampiran 16 Saran Responden





Lampiran 17 Dokumentasi Wawancara