



**PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION
TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA
DIJKSTRA DAN FRAMEWORK LARAVEL**

SKRIPSI

MUHAMMAD REZA ADI PRATAMA

NIM. 2210511080

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA
2026**



**PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION
TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA
DIJKSTRA DAN FRAMEWORK LARAVEL**

SKRIPSI

MUHAMMAD REZA ADI PRATAMA

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA
2026**

PERNYATAAN ORISINALITAS

Skripsi ini merupakan hasil karya sendiri serta semua sumber referensi yang dikutip maupun yang dirujuk telah saya nyatakan benar.

Nama : Muhammad Reza Adi Pratama
NIM : 2210511080
Program Studi : Informatika Program Sarjana
Tanggal : 23 Juli 2026

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan saya ini, maka bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku.

Jakarta, 23 Juli 2026

Yang Menyatakan,

A handwritten signature in black ink is written over a yellow 10,000 Rupiah Indonesian postage stamp. The stamp features the Garuda Pancasila emblem and the serial number 355ALX010005663.

Muhammad Reza Adi Pratama

LEMBAR PERSETUJUAN SIDANG SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Muhammad Reza Adi Pratama
NIM : 2210511080
Program Studi : Informatika Program Sarjana
Jenis Tugas Akhir : Skripsi

Judul Tugas Akhir Skripsi :

PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION TRANSJAKARTA
BERBASIS GTFS DENGAN ALGORITMA DIJKSTRA
DAN FRAMEWORK LARAVEL.

Dinyatakan telah memenuhi syarat dan menyetujui untuk mengikuti ujian sidang Tugas Akhir Skripsi.

Jakarta, 26 Juni 2026

Menyetujui,

Dosen Pembimbing I,



Ditandatangani secara digital oleh:
Jayanta

Verifikasi di design.upnvj.ac.id

Jayanta, S.Kom., M.Si

NIDN. 0030096111

Dosen Pembimbing II,



Ditandatangani secara digital oleh:
Neny Rosmawarni

Verifikasi di design.upnvj.ac.id

Neny Rosmawarni, S.Kom., M.Kom

NIDN. 0312018701

Mengetahui,

Koordinator Program Studi,



Ditandatangani secara digital oleh:
Ridwan Raafi'udin

Verifikasi di design.upnvj.ac.id

Dr. Ridwan Raafi'udin, S.Kom., M.Kom

NIDN. 002058804

LEMBAR PENGESAHAN

Judul : Prototype Dual-Preference Route Optimization Transjakarta Berbasis GTFS dengan Algoritma Dijkstra dan Framework Laravel
Nama : Muhammad Reza Adi Pratama
NIM : 2210511080
Program Studi : S1 Informatika

Disetujui oleh:

Penguji 1:

Dr. Noor Falih, S.Kom., M.T.

Penguji 2:

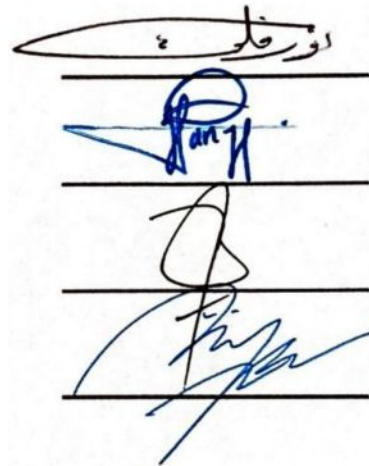
Muhammad Panji Muslim, S.Pd., M.Kom.

Pembimbing 1:

Jayanta, S.Kom., M.Si

Pembimbing 2:

Neny Rosmawarni, S.Kom., M.Kom.

The image shows four horizontal lines for signatures. The first line has a signature in brown ink. The second line has a signature in blue ink. The third line has a signature in black ink. The fourth line has a signature in blue ink.

Diketahui oleh:

Koordinator Program Studi:

Dr. Ridwan Raafi'udin, S.Kom, M.Kom.

NIP. 198805022021211001

Dekan Fakultas Ilmu Komputer:

Prof. Dr. Ir. Supriyanto, ST., M.Sc., IPM.

NIP. 197605082003121002

Tanggal Ujian Tugas Akhir

6 Juli 2026

The image shows two horizontal lines for signatures. The first line has a signature in black ink. The second line has a signature in black ink. Below the second line is a blue circular official stamp of the Dean of the Faculty of Computer Science, Universitas Pendidikan Indonesia (UPI).

PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa skripsi dengan judul “PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA DIJKSTRA DAN FRAMEWORK LARAVEL” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Universitas Pembangunan Nasional “Veteran” Jakarta.

Jakarta, 26 Juni 2026



Muhammad Reza Adi Pratama

210511080

**PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN
AKADEMIS**

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK
KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Universitas Pembangunan Nasional “Veteran” Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Reza Adi Pratama
NIM : 2210511080
Fakultas : Ilmu Komputer
Program Studi : Informatika Program Sarjana

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan karya ilmiah saya kepada Universitas Pembangunan Nasional "Veteran" Jakarta Hak Bebas Royalti Non-Eksklusif (*Non-Exchange Royalty Free Right*) untuk dipublikasikan dengan judul:

**PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION
TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA
DIJKSTRA DAN FRAMEWORK LARAVEL**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti ini Universitas Pembangunan Nasional "Veteran" Jakarta berhak menyimpan, mengalih media atau memformatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan artikel ilmiah saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Ditetapkan di: Jakarta
Tanggal Persetujuan: 26 Juni 2026
Yang Menyatakan,



Muhammad Reza Adi Pratama

PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA DIJKSTRA DAN FRAMEWORK LARAVEL

Muhammad Reza Adi Pratama

ABSTRAK

Kemacetan Jakarta masih menjadi masalah utama. Meskipun Transjakarta mengoperasikan 243 koridor dengan 8.651 halte, sistem pencarian rute terintegrasi masih terbatas. Penelitian ini bertujuan membangun prototipe pencarian rute Transjakarta berbasis data GTFS dengan algoritma Dijkstra dalam dua mode yaitu prioritas jarak terpendek dan prioritas minimum transfer. Metode menggunakan rekayasa sistem dengan pengumpulan data GTFS dan POI, pembersihan data Python/QGIS, implementasi Laravel 12, serta pengujian *Black box* dan verifikasi Python. Hasilnya, sistem berhasil mengintegrasikan 243 koridor, 8.651 halte, dan 32.000 POI dengan validasi GTFS menurunkan *error* dari 47 menjadi 0 dan *warning* 96%. Dijkstra pada dua mode menghasilkan waktu eksekusi rata-rata 1.200–1.450 ms, sesuai kompleksitas $O((V + E) \log V)$. Verifikasi menunjukkan bahwa sistem menghasilkan jumlah perpindahan koridor yang sama dengan perhitungan manual pada skenario kompleks dan selisih jarak hanya 2,2%. *Black box* terhadap 13 fungsi dan 16 API mencapai 100% keberhasilan. Kesimpulannya, prototipe ini akurat, efisien, dan layak digunakan untuk perencanaan perjalanan Transjakarta dengan dua preferensi pencarian.

Kata Kunci: Transjakarta, GTFS, Algoritma Dijkstra, Laravel, Pencarian Rute

A GTFS-BASED DUAL-PREFERENCE ROUTE OPTIMIZATION PROTOTYPE FOR TRANSJAKARTA USING DIJKSTRA'S ALGORITHM AND THE LARAVEL FRAMEWORK

Muhammad Reza Adi Pratama

ABSTRACT

Traffic congestion in Jakarta remains a major Problem. Although Transjakarta operates 243 corridors with 8,650 stops, an integrated Route search system is still limited. This research aims to build a Transjakarta Route search prototipe based on GTFS data using Dijkstra's algorithm in two modes namely shortest distance and minimum transfer priority. The method uses system engineering with GTFS and POI data collection, Python/QGIS data Cleaning, Laravel 12 implementation, and Black Box Testing with Python verification. The results show the system successfully integrates 243 corridors, 8,650 stops, and 32,000 POI, with GTFS validation reducing errors from 47 to 0 and warnings by 96%. Dijkstra in two modes achieves average execution times of 1,200–1,450 ms, consistent with $O((V + E) \log V)$ complexity. Independent verification demonstrates that the system yields identical transfer counts compared to manual calculation in complex scenarios, with a maximum distance deviation of only 2.2%. Black Box Testing on 13 functions and 16 APIs achieved 100% success. In conclusion, this prototipe is accurate, efficient, and feasible for Transjakarta travel planning with two search preferences.

Keywords: Transjakarta, GTFS, Dijkstra Algorithm, Laravel, Route Search

KATA PENGANTAR

Bismillāhirrahmānirrahīm. Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, taufik, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Shalawat serta salam semoga senantiasa tercurah kepada junjungan Nabi Besar Muhammad SAW., beserta keluarga, para sahabat, dan seluruh pengikut beliau hingga akhir zaman. Skripsi ini dilaksanakan sejak bulan Juni 2025 sampai bulan Juli 2026 dengan judul “PROTOTYPE DUAL-PREFERENCE ROUTE OPTIMIZATION TRANSJAKARTA BERBASIS GTFS DENGAN ALGORITMA DIJKSTRA DAN FRAMEWORK LARAVEL”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta (UPNVJ). Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Kedua orang tua yang penulis hormati, Rahayu Tri Wardhani dan Octa Nugroho Shando, beserta seluruh keluarga besar, saudara, dan kerabat terdekat yang senantiasa memberikan doa, dukungan moril dan material, serta motivasi pada penulis selama penyelesaian studi dan penyusunan skripsi.
2. Prof. Dr. Ir. Supriyanto, S.T., M.Sc., IPM selaku dekan Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta sekaligus Dosen Pembimbing Akademik yang telah memberikan arahan selama perkuliahan.
3. Dr. Ridwan Raafi'udin, S.Kom., M.Kom. selaku Koordinator Program Studi Sarjana Jurusan Informatika Fakultas Ilmu Komputer UPN Veteran Jakarta yang telah memberikan dukungan pada penyelenggaraan kegiatan akademik.
4. Jayanta, S.Kom., M.Si. selaku Dosen Pembimbing 1 yang senantiasa mendorong penulis untuk menjaga kualitas dan urgensi penelitian melalui kritik, saran, serta analisis evaluatif, sekaligus membimbing penulis dalam membangun kedisiplinan, ketelitian, dan pola pikir solutif terhadap pemecahan masalah sehingga penelitian dapat disusun secara sistematis.
5. Neny Rosmawarni, S.Kom., M.Kom selaku Dosen Pembimbing 2 yang telah meluangkan waktu, tenaga, pikiran, serta memberikan bimbingan, perhatian,

motivasi, dan arahan kepada penulis dengan penuh kesabaran selama proses penyusunan sejak tahap awal hingga skripsi ini dapat diselesaikan.

6. Aryajaya Alamsyah, S.Kom., M.Kom. selaku Instruktur Privat yang telah membimbing penulis mengembangkan pola pikir kritis, keterampilan teknis dan komunikasi akademik, serta wawasan yang sangat berharga dalam proses penyusunan skripsi ini dan pengembangan diri penulis.
7. Seluruh tenaga kependidikan dan staf administrasi Fakultas Ilmu Komputer UPN Veteran Jakarta yang telah membantu kelancaran berbagai proses administrasi akademik selama masa studi dan penyusunan skripsi.
8. Universitas Pembangunan Nasional Veteran Jakarta, khususnya Fakultas Ilmu Komputer, yang telah menyediakan sarana, prasarana, fasilitas, serta sumber daya pendukung selama proses penyusunan skripsi.
9. Seluruh responden penelitian serta rekan-rekan yang telah berpartisipasi dalam pengisian kuesioner, memberi masukan selama bimbingan, pengujian, evaluasi, dan penyempurnaan sistem selama penelitian berlangsung.
10. Seluruh rekan seperjuangan mahasiswa Program Studi Informatika angkatan 2022 yang telah memberikan dukungan, semangat, serta menjadi teman berdiskusi dan tempat berbagi pengalaman selama proses penyusunan skripsi.
11. PT Transportasi Jakarta sebagai penyedia data *General Transit Feed Specification* (GTFS), MobilityData sebagai penyedia *GTFS Validator*, serta OpenStreetMap sebagai penyedia data *Point of Interest* (POI) yang menjadi sumber data pendukung dalam penelitian ini.
12. Seluruh pihak yang telah membantu penulis, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu.

Semoga skripsi ini dapat memberikan manfaat bagi pembaca, pihak yang membutuhkan, serta menjadi kontribusi bagi pengembangan ilmu pengetahuan, khususnya di bidang informatika.

Jakarta, Juni 2026



Muhammad Reza Adi Pratama
2210511080

DAFTAR ISI

SKRIPSI.....	ii
PERNYATAAN ORISINALITAS.....	iii
LEMBAR PERSETUJUAN SIDANG SKRIPSI	iv
LEMBAR PENGESAHAN	v
PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	vi
PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	vii
ABSTRAK	viii
ABSTRACT.....	ix
KATA PENGANTAR.....	x
DAFTAR ISI	xii
DAFTAR TABEL.....	xix
DAFTAR GAMBAR	xxi
DAFTAR RUMUS.....	xxiii
DAFTAR LAMPIRAN	xxiv
DAFTAR SIMBOL.....	xxv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	4
1.4. Manfaat Penelitian	4
1.4.1. Bagi Peneliti:.....	4
1.4.2. Bagi Pengguna Umum:	4
1.5. Batasan Masalah.....	5
1. Wilayah dan Operator Transportasi.....	5
2. Data GTFS Statis	5
3. Estimasi waktu tempuh tidak dinamis.....	5
4. Tidak mendukung pencarian berbasis waktu	5
5. Keterbatasan parameter optimasi	5
6. Data <i>shape_dist_traveled</i> tidak digunakan sebagai bobot	6
7. Akurasi visualisasi <i>shape</i> koridor	6
8. Lingkungan pengujian terbatas	6
1.6. Sistematika Penulisan	6
BAB 2 TINJAUAN PUSTAKA.....	10
2.1. Kajian Pustaka.....	10

2.2.	Sistem Transportasi Multi-rute.....	17
2.3.	<i>General Transit Feed Specification (GTFS)</i>	17
2.4.	<i>Point of Interest (POI)</i>	18
2.5.	Integrasi Data GTFS dan Transformasi Multi-Rute.....	18
2.5.1.	Canonical GTFS Schedule Validator	19
2.5.2.	Pandas pada Python	19
2.6.	Inovasi Sistem Transportasi Berbasis GTFS.....	19
2.7.	Algoritma Optimasi Rute	20
2.7.1.	Algoritma Dijkstra	20
2.7.2.	Algoritma A* (A-Star)	21
2.7.3.	Algoritma Bellman-Ford.....	22
2.7.4.	Algoritma Floyd-Warshall	22
2.7.5.	Algoritma Genetika.....	22
2.8.	Teori Graf.....	24
2.9.	<i>Framework</i> Laravel.....	26
2.9.1.	<i>Geocoding dan Reverse Geocoding</i>	28
2.9.2.	JSON (<i>JavaScript Object Notation</i>)	28
2.10.	Metode Perhitungan Jarak Geodesik.....	28
2.10.1.	Rumus Jarak Euclidean (Pythagoras)	29
2.10.2.	Rumus Jarak Equirectangular	29
2.10.3.	Rumus Haversine (<i>Great-Circle</i>)	29
2.10.4.	Rumus Vincenty (<i>Inverse Problem</i>).....	30
2.11.	API dan Web Services.....	32
2.12.	<i>Quality Assurance, Black Box Testing, dan Usability Testing</i>	32
2.13.	Diagram Pemodelan Sistem (UML dan Non-UML).....	33
2.13.1.	<i>Unified Modeling Language (UML)</i>	33
2.13.2.	Diagram Non-UML.....	34
2.14.	Penelitian Terdahulu terhadap Transportasi Multi-rute.....	35
BAB 3 METODOLOGI PENELITIAN.....		36
3.1.	Jenis Penelitian.....	36
3.2.	Metode Penelitian.....	36
3.3.	Tahapan Penelitian	41
1.	Studi Literatur dan Analisis Masalah	41
2.	Pengumpulan dan Integrasi Data GTFS.....	41

3.4.	Pengumpulan Data	41
1.	Data Sekunder GTFS Transjakarta	41
2.	Data Sekunder <i>Point of Interest</i> (POI) Jakarta.....	41
3.5.	Pembersihan Data.....	42
3.5.1.	Karakteristik <i>Dataset</i> Penelitian	42
3.5.2.	Pembersihan Data GTFS Transjakarta	43
3.5.3.	Pembersihan Data POI menggunakan Software QGIS	45
3.6.	Pemodelan dan Analisis Data.....	48
3.6.1.	Analisis Karakteristik Data GTFS	48
3.6.2.	Pemodelan Algoritma.....	49
3.6.3.	Diagram Pemodelan dan Analisis Data.....	56
1.	ERD <i>Database</i>	56
2.	<i>Use Case Diagram</i>	58
3.	<i>Component Diagram</i>	60
3.7.	Implementasi dan Pengujian Sistem	62
3.7.1.	Justifikasi Pemilihan Tools dan Teknologi.....	62
1.	Perbandingan Laravel dengan <i>framework backend</i> lain.	62
2.	Perbandingan Python dengan alternatif <i>preprocessing/verifikasi</i>	63
3.	Pemilihan teknologi penelitian dari Laravel, Python, MySQL, GitHub, Railway beserta alternatifnya.....	63
4.	Peran setiap teknologi dalam penelitian.....	64
5.	Pengukuran internal <i>database</i> :.....	65
3.7.2.	Spesifikasi Peralatan	65
1.	Lingkungan pengujian perangkat keras. Laptop dengan spesifikasi:....	65
2.	Lingkungan pengembangan perangkat lunak.....	65
3.	Pengujian sistem dilakukan untuk menilai:	66
3.7.3.	Diagram Alur Sistem.....	67
1.	<i>Activity Diagram</i> Global Sistem Terintegrasi	67
2.	<i>Activity Diagram</i> <i>Middleware Layer</i>	69
3.	<i>Activity Diagram</i> <i>Cache Management</i>	70
4.	<i>Activity Diagram</i> <i>Geocoding</i> dan Validasi POI.	70
5.	<i>Activity Diagram</i> Eksekusi Algoritma Dijkstra.....	71
6.	<i>Activity Diagram</i> Penyimpanan Log Pencarian.	73
3.8.	Tempat dan Jadwal Penelitian.....	73

BAB 4 HASIL DAN PEMBAHASAN.....	76
4.1. Pendahuluan	76
4.2. Lingkungan Implementasi.....	76
4.2.1. Perangkat Keras	76
4.2.2. Perangkat Lunak.....	77
4.2.3. Pustaka Pendukung	79
4.2.4. Arsitektur Sistem.....	81
1. Alur Persiapan dan Import Data.....	82
2. Struktur <i>Database</i>	82
3. Alur Sisi Pengguna dan <i>Frontend</i>	82
4. Alur Sisi <i>Server</i> Laravel 12.....	83
4.3. Konfigurasi Data GTFS	83
4.3.1. Komposisi <i>Dataset</i>	83
4.3.2. Validasi <i>Dataset</i> GTFS dan POI	84
4.3.3. Proses Impor Data	88
4.4. Implementasi <i>Database</i>	89
4.4.1. Struktur Tabel.....	89
4.4.2. Relasi Antar Tabel	90
4.4.3. Mekanisme <i>Caching Database</i>	91
4.4.4. Struktur Direktori dan Modul Implementasi.....	92
4.4.5. Pemetaan Arsitektur ke Komponen Implementasi	95
4.4.6. <i>Traceability Requirements</i>	97
4.4.7. Validasi <i>Use Case</i>	99
4.4.8. Implementasi <i>Database</i>	100
4.5. Implementasi Fitur Pencarian POI (<i>Point of Interest</i>)	104
4.5.1. Latar Belakang Penambahan POI	104
4.5.2. Sumber Data POI	104
4.5.3. Arsitektur Pencarian POI	104
4.5.4. Seleksi Halte Terdekat dari POI	105
4.5.5. Notifikasi Jarak Jauh	107
4.5.6. Tampilan Hasil POI dalam <i>Autocomplete</i>	107
4.6. Implementasi <i>Autocomplete</i> Pencarian	107
4.6.1. Mekanisme <i>Autocomplete</i>	107
4.6.2. Hasil Pencarian.....	108

4.7.	Implementasi Visualisasi Peta.....	109
4.7.1.	<i>Library</i> Peta.....	109
4.7.2.	Batas Wilayah Peta.....	109
4.7.3.	<i>Layer</i> Koridor Transjakarta	109
4.7.4.	<i>Layer</i> Halte.....	109
4.7.5.	<i>Layer</i> POI (<i>Point of Interest</i>)	109
4.8.	Implementasi Pencarian Rute dengan Algoritma Dijkstra.....	110
4.8.1.	Representasi Graf.....	110
4.8.2.	Rumus Haversine untuk Perhitungan Jarak	110
4.8.3.	Pembangunan Graf.....	110
4.8.4.	Dua Preferensi Pencarian	111
4.8.5.	Proses Eksekusi Algoritma.....	113
4.8.6.	Diagram Interaksi Teknis	114
4.8.7.	Integrasi Data <i>Frequencies</i> untuk Optimasi Waktu	121
4.9.	Siklus Status <i>Cache</i> & Pencarian.....	121
4.10.	Implementasi <i>Filter</i> Jadwal Berdasarkan Hari (<i>Calendar</i>).....	123
4.10.1.	Latar Belakang Operasional Rute	123
4.10.2.	Data <i>Calendar</i>	123
4.10.3.	Implementasi <i>Filter</i> Kalender GTFS	124
4.10.4.	Dampak <i>Filter</i>	125
4.10.5.	Tampilan Koridor Tidak Aktif.....	125
4.10.6.	Hasil Pengujian Simulasi <i>Filter</i> Kalender.....	125
4.11.	Implementasi Sistem Autentikasi dan Manajemen Pengguna	131
4.11.1.	Arsitektur Autentikasi	131
4.11.2.	Proses <i>Login</i> dan <i>Register</i>	131
4.11.3.	<i>Middleware Role</i> dan Proteksi <i>Route</i>	132
4.11.4.	Manajemen Pengguna dan Riwayat Pencarian	132
4.11.5.	CRUD POI (<i>Point of Interest</i>).....	132
1.	Batasan dan Validasi Wilayah untuk Entri Data POI	132
2.	Penggunaan Tools Pendukung untuk Entri POI.....	133
4.12.	Implementasi Penyimpanan Log Pencarian	134
4.12.1.	Tujuan Log Pencarian	134
4.12.2.	Struktur Data Log.....	134
4.12.3.	Detail Log dalam Format JSON.....	135

4.12.4.	Visualisasi Log di Halaman Algoritma	137
4.13.	Hasil Pengujian Sistem	138
4.13.1.	Pengujian Fungsional (<i>Black Box Testing</i>).....	138
4.13.2.	Pengujian Skenario <i>Input-Output</i>	139
4.13.3.	Pengujian API	140
4.13.4.	Pengujian Akurasi Rute.....	142
1.	Penyesuaian Parameter Pengujian.....	143
2.	Perancangan <i>Script</i> Python Independen.....	143
1)	<i>Script</i> untuk pengambilan data terhubung dengan aplikasi.....	143
2)	Graf dibangun sebagai <i>adjacency list</i>	143
3)	Algoritma Dijkstra diimplementasikan menggunakan <i>heapq (priority queue)</i>	144
4)	Rekonstruksi Akhir.....	144
3.	Penyesuaian dan Perbaikan Kode	144
1)	Penambahan Arah.....	146
2)	Perubahan dari basis <i>trip</i> ke basis koridor.....	146
3)	Penonaktifan sementara bonus negative.	146
4.	Hasil Verifikasi dan Perbandingan	146
5.	Analisis dan Pembahasan.....	147
1)	Kesamaan Jumlah Transfer (100% Akurat).	147
2)	Selisih Jarak yang Minimal pada Skenario Kompleks.....	147
3)	Perbedaan pada Rute Pendek.	148
6.	Analisis Fenomena <i>Over-transit</i> dan Usulan Optimasi Graf Menggunakan <i>Transfer Penalty</i>	148
1)	Alternatif Pencarian 1 (Cari 1):.....	148
2)	Alternatif Pencarian 2 (Cari 2):.....	149
4.13.5.	Pengujian Performa Graf dan Analisis Kompleksitas Waktu.....	150
1.	Kompleksitas Teoritis Waktu Dijkstra	150
2.	Waktu Eksekusi Aktual dari Log Pencarian.....	151
3.	Hubungan antara Jumlah <i>Node</i> dan Waktu Eksekusi.....	151
4.	Penggunaan Memori	154
4.13.6.	Pengujian Beban Infrastruktur (<i>Load Testing</i>).....	157
4.13.7.	Pemantauan Performa Aplikasi Menggunakan <i>Railway Metrics</i> ...	165
4.13.8.	Contoh Hasil Pencarian Rute	167

4.13.9.	Contoh Data Log Pencarian dari <i>Database</i>	168
4.13.10.	<i>Usability Testing</i>	168
1.	Perancangan Instrumen dan Struktur Kuesioner <i>Usability Testing</i>	169
2.	Analisis Kuantitatif Hasil Kuesioner Google Form Responden	171
3.	Analisis Kuantitatif Hasil Kuesioner Google Form Responden.	173
4.	Analisis Kuantitatif Hasil Kuesioner Google Form Responden.	174
4.14.	Antarmuka Pengguna (<i>User Interface</i>)	174
4.14.1.	Halaman Registrasi dan <i>Login</i>	174
4.14.2.	Halaman Peta Rute	175
4.14.3.	Halaman Algoritma	176
4.14.4.	Halaman Data Halte	176
4.14.5.	Halaman Data Koridor	177
4.14.6.	Halaman Log Pencarian	178
4.15.	Pembahasan.....	178
4.15.1.	Pencapaian Tujuan Penelitian	178
4.15.2.	Keutamaan Keberhasilan Pengujian	179
4.15.3.	Keterbatasan Pengujian	180
4.16.	Analisis Perbandingan Rute Berdasarkan Wilayah.....	186
4.16.1.	Lokasi Uji dan Koordinat.....	186
4.16.2.	Hasil Pengujian Halte ke Halte	188
4.16.3.	Hasil Pengujian POI ke POI.....	189
4.16.4.	Analisis Perbandingan.....	190
4.16.5.	Perbandingan Kompleksitas.....	192
4.16.6.	Analisis Variasi Hasil Berdasarkan Topologi Graf	194
4.17.	<i>Deployment Pipeline</i>	195
BAB 5	KESIMPULAN DAN SARAN	201
5.1.	Kesimpulan	201
5.2.	Saran.....	204
DAFTAR PUSTAKA		206
DAFTAR RIWAYAT HIDUP		211
LAMPIRAN		212

DAFTAR TABEL

Tabel 2.1. Studi Literature Review.....	11
Tabel 2.2. Perbandingan Algoritma Optimasi Rute.	23
Tabel 2.3. Jenis Graf.....	25
Tabel 2.4. Perbandingan Algoritma Optimasi Rute.	30
Tabel 3.1. Karakteristik Dataset Penelitian.	42
Tabel 3.2. Hasil Validasi GTFS Transjakarta dengan Canonical GTFS.....	45
Tabel 3.3. Parameter Bobot Jarak.	50
Tabel 3.4. Perbandingan antara Framework Website.	62
Tabel 3.5. Perbandingan antara Framework Preprocessing.	63
Tabel 3.6. Pemilihan Teknologi dalam Penelitian.....	64
Tabel 3.7. Peran Teknologi dalam Penelitian.	64
Tabel 3.8. Pengukuran Internal Database.....	65
Tabel 3.9. Kategori Pengujian Aplikasi.....	66
Tabel 3.10. Tempat dan Jadwal Penelitian.	74
Tabel 4.1. Spesifikasi Perangkat Keras.	76
Tabel 4.2. Spesifikasi Perangkat Lunak.	77
Tabel 4.3. Pustaka Pendukung.	79
Tabel 4.4. Komposisi Dataset GTFS.....	83
Tabel 4.5. Perbandingan Kondisi Sebelum dan Sesudah Pembersihan Dataset GTFS.....	85
Tabel 4.6. Perbandingan Kondisi Pembersihan Dataset GTFS.....	85
Tabel 4.7. Struktur Tabel Dataset GTFS dalam Database.....	89
Tabel 4.8. Struktur Direktori Projek.....	92
Tabel 4.9. Pemetaan Arsitektur dalam Komponen Aplikasi.	95
Tabel 4.10. Traceability Requirements.....	97
Tabel 4.11. Validasi Use Case.	99
Tabel 4.12. Radius Pencarian Otomatis Halte Terdekat.	107
Tabel 4.13. Ikon Hasil Pencarian di Peta.	108
Tabel 4.14. Strategi Pembobotan.	112
Tabel 4.15. Data Calendar GTFS Transjakarta.	123
Tabel 4.16. Jadwal Operasional Koridor Keseluruhan.....	127
Tabel 4.17. Perbandingan Pendekatan tb_calendar vs tb_trips (aktif).	129
Tabel 4.18. Data yang Disimpan dalam Log Pencarian.	135
Tabel 4.19. Pengujian Fungsional.	138
Tabel 4.20. Pengujian Input-Output.	139
Tabel 4.21. Pengujian API Integration.	140
Tabel 4.22. Pengecekan Halte berdasarkan Regional Jakarta.	143
Tabel 4.23. Penyesuaian Parameter Logika Sistem.	145
Tabel 4.24. Perbandingan Pengujian Cari Rute Manual dan Aplikasi.	147
Tabel 4.25. Perbandingan Performa Cari 1 dan Cari 2.	151
Tabel 4.26. Data Mentah Analisis Regresi (30 Sampel Acak).	152
Tabel 4.27. Ringkasan Hasil Analisis Regresi Linier.	153

Tabel 4.28. Penggunaan Memori oleh Aplikasi.	154
Tabel 4.29. Ringkasan hasil pengujian baseline 1 user.....	159
Tabel 4.30. Ringkasan hasil pengujian lokal performa infrastruktur	160
Tabel 4.31. Ringkasan hasil pengujian lokal ekstrem performa infrastruktur	161
Tabel 4.32. Ringkasan Hasil Pengujian Beban Production (Railway).....	163
Tabel 4.33. Perbandingan Kelima Skenario Uji Performa.	164
Tabel 4.34. Ringkasan hasil pengujian public dengan Railway Metrics.....	165
Tabel 4.35. Contoh Hasil Pencarian Rute dengan Aplikasi.	167
Tabel 4.36. Contoh Perbandingan Data Log Pencarian dari sebuah Rute.	168
Tabel 4.37. Contoh Perbandingan Data Log Pencarian dari sebuah Rute.	169
Tabel 4.38. Distribusi Frekuensi dan Skor Aktual Kuesioner Usability Testing.	171
Tabel 4.39. Keutamaan Pengujian Sistem berdasarkan Hasil Terlihat.....	179
Tabel 4.40. Keterbatasan Pengujian Sistem berdasarkan Hasil Terlihat.	180
Tabel 4.41. Pengecekan Koordinat Regional Jakarta dengan Tinker.	187
Tabel 4.42. Pengujian Antar Halte.	188
Tabel 4.43. Antar Lokasi POI.....	189
Tabel 4.44. Hasil Pencarian Rute Jakarta dengan Preferensi Cari 1.	190
Tabel 4.45. Hasil Pencarian Rute Jakarta dengan Preferensi Cari 2.	190
Tabel 4.46. Perbandingan Karakteristik Cari 1 dan Cari 2.	191
Tabel 4.47. Perbandingan Bobot.	192
Tabel 4.48. Perbandingan Kompleksitas.....	193
Tabel 4.49. Komponen Deployment pada Railway Cloud.....	196
Tabel 4.50. Parameter Deployment Railway.....	199
Tabel 4.51. Parameter Variabel Sistem.....	199
Tabel 4.52. Pengaturan Stabilitas Deployment.	200

DAFTAR GAMBAR

Gambar 1.1. Lima kota dengan kemacetan terparah di Indonesia menurut Data TomTom.	1
Gambar 1.2. Perbandingan Jumlah Halte Transjakarta pada tahun 2024.	2
Gambar 2.1. Persentase Laravel untuk Framework Website.	27
Gambar 3.1. Kerangka Kerja Applied Analytical Experimental Research.	37
Gambar 3.2. Tahapan Penelitian Sistem Integrasi GTFS.	38
Gambar 3.3. Alur Pembersihan Data GTFS.	44
Gambar 3.4. Alur Pembersihan Data POI.	46
Gambar 3.5. Tampilan Visualisasi Data POI Jakarta menggunakan Aplikasi QGIS.	48
Gambar 3.6. Implementasi Dijkstra.	52
Gambar 3.7. Diagram ERD.	57
Gambar 3.8. Use Case Diagram.	59
Gambar 3.9. Component Diagram.	61
Gambar 3.10. Activity Diagram Pencarian Rute & Alur Utama.	68
Gambar 3.11. Activity Diagram Middleware Layer.	69
Gambar 3.12. Activity Diagram Cache Management.	70
Gambar 3.13. Activity Diagram Geocoding dan Validasi POI.	71
Gambar 3.14. Activity Diagram Eksekusi Algoritma Dijkstra.	72
Gambar 3.15. Activity Diagram Penyimpanan Log Pencarian.	73
Gambar 4.1. Arsitektur Aplikasi dalam Laravel.	81
Gambar 4.2. Pengecekan Isi Dataset GTFS dengan Canonical.	84
Gambar 4.3. Data GTFS dan POI yang telah diolah.	88
Gambar 4.4. Class Diagram Service.	101
Gambar 4.5. Class Diagram Controller.	103
Gambar 4.6. Alur Fitur Pencarian POI.	105
Gambar 4.7. Alur Pemilihan Halte Terdekat suatu POI.	106
Gambar 4.8. Sequence Diagram Skenario 1 : Muat Halaman Peta dan Cache GTFS.	115
Gambar 4.9. Sequence Diagram Skenario 2 : Autocomplete Pencarian Halte dan POI.	116
Gambar 4.10. Sequence Diagram Skenario 3 : Pencarian Rute Dijkstra + POI + Logging.	118
Gambar 4.11. Sequence Diagram Skenario 4 : Autentikasi Pengguna (Login/Register/Logout).	120
Gambar 4.12. Siklus Status Cache.	122
Gambar 4.13. Diagram Service Calendar Jadwal Transjakarta.	124
Gambar 4.14. Pengecekan Filter Kalender dengan Tinker.	126
Gambar 4.15. Pengecekan Jumlah Service Transjakarta per Minggu dengan Tinker.	126
Gambar 4.16. Bukti Jumlah Service Transjakarta per Minggu dengan Tinker. ..	127
Gambar 4.17. Bukti Jumlah Service Transjakarta per Minggu dengan Tinker. ..	129

Gambar 4.18. Alur Autentikasi Pengguna Aplikasi.	131
Gambar 4.19. Contoh Pencarian Lokasi Umum dengan Alamat, Koordinat, dan Gulir Manual.	134
Gambar 4.20. Contoh Pengecekan Koordinat GPS Lokasi Tempat Umum.	134
Gambar 4.21. Contoh Pengecekan Data Pencarian Log dengan Tinker.	136
Gambar 4.22. Contoh Pengecekan Data Rute dengan Tinker.	136
Gambar 4.23. Contoh Pengecekan Data Koridor dengan Tinker.	136
Gambar 4.24. Halaman Algoritma dalam kondisi kosong (belum ada pencarian).	137
Gambar 4.25. Halaman Algoritma setelah pencarian rute dan/atau pemilihan log history.	138
Gambar 4.26. Hasil pengujian baseline 1 user – Percobaan 1 (Cold start).....	158
Gambar 4.27. Hasil pengujian baseline 1 user – Percobaan 2 (Cache hit)	158
Gambar 4.28. Hasil pengujian baseline 1 user – Percobaan 3 (Cache hit)	158
Gambar 4.29. Hasil pengujian beban lokal menggunakan Postman Collection Runner	160
Gambar 4.30. Hasil pengujian beban lokal ekstrem menggunakan Postman Collection Runner	161
Gambar 4.31. Hasil pengujian beban publik menggunakan Postman Collection Runner	163
Gambar 4.32. Tampilan Registrasi pada Aplikasi.	175
Gambar 4.33. Tampilan Login pada Aplikasi.	175
Gambar 4.34. Tampilan Pencarian Rute Transjakarta pada Aplikasi.	175
Gambar 4.35. Tampilan Perhitungan Rute pada Aplikasi.	176
Gambar 4.36. Tampilan Data Halte pada Aplikasi.	176
Gambar 4.37. Tampilan Data Koridor pada Aplikasi.	177
Gambar 4.38. Tampilan Log Pencarian Admin pada Aplikasi.	178
Gambar 4.39 Pengecekan Koordinat Lokasi per Regional Jakarta dengan Tinker.	187
Gambar 4.40. Proses Deployment.....	196

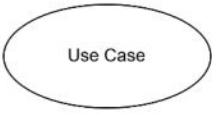
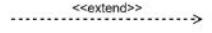
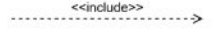
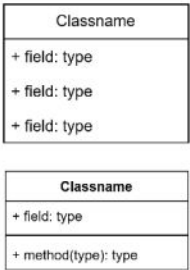
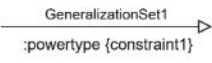
DAFTAR RUMUS

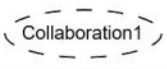
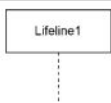
(2.1) Rumus Jarak Euclidean (Pythagoras).....	29
(2.2) Rumus Jarak Equirectangular.....	29
(2.3) Rumus Haversine.....	30
(2.4) Rumus Vincenty (Inverse Problem)	30
(3.1) Rumus Bobot Perjalanan Dijkstra	49
(3.2) Rumus Lengkap Bobot Dijkstra	50
(3.3) Rumus Haversine (Perhitungan Jarak)	50
(4.1) Kompleksitas Waktu Dijkstra.....	150
(4.2) Kompleksitas Waktu Floyd-Warshall (pembanding).....	150
(4.3) Persamaan Regresi Linier (waktu eksekusi vs node)	153

DAFTAR LAMPIRAN

Lampiran 1. A. Kode Python Pra-Pemrosesan & Verifikasi	212
Lampiran 2. B. Bukti Visual Implementasi Kode & Antarmuka	219
Lampiran 3. C. Database MySQL dengan PhyMyAdmin	228
Lampiran 4. D. Deployment Aplikasi Web	232
Lampiran 5. E. Backup Dokumentasi	237
Lampiran 6. F. Bukti Hasil Usability Testing.....	238
Lampiran 7. G. Bukti Turnitin	244

DAFTAR SIMBOL

Simbol	Nama / Notasi	Keterangan
<i>Use Case Diagram</i>		
	<i>Aktor</i>	Entitas di luar sistem yang berinteraksi dengan <i>Use Case</i> (manusia atau sistem eksternal).
	<i>Use Case</i>	Fungsionalitas atau layanan yang disediakan sistem kepada aktor, digambarkan dengan oval.
	<i>System Boundary</i>	Batas lingkup sistem (kotak persegi panjang) yang memisahkan sistem dengan aktor eksternal.
	<i>Association</i>	Relasi komunikasi antara aktor dan <i>Use Case</i> (garis lurus tanpa panah).
	<i>Extend</i>	Relasi opsional (garis putus-putus) dari satu <i>Use Case</i> ke <i>Use Case</i> lain, menunjukkan fungsionalitas tambahan yang hanya berjalan dalam kondisi tertentu.
	<i>Include</i>	Relasi wajib (garis putus-putus) dari satu <i>Use Case</i> ke <i>Use Case</i> lain, menunjukkan fungsionalitas yang selalu dipanggil.
	<i>Generalization</i>	Relasi pewarisan (panah dengan segitiga kosong) dari aktor anak ke aktor induk.
<i>Class Diagram</i>		
	<i>Class</i>	Representasi kelas (kotak dengan 3 bagian: nama, atribut, metode).
	<i>Association</i>	Relasi struktural antar kelas (garis lurus).
	<i>Inheritance (Generalisasi)</i>	Relasi pewarisan (panah segitiga kosong) dari kelas anak ke kelas induk.

	<i>Dependency</i>	Relasi ketergantungan (panah putus-putus), biasanya untuk <i>dependency injection</i> .
	<i>Collaboration</i>	Memetakan perilaku sebuah sistem dalam konteks atau <i>Use Case</i> tertentu
	<i>Realization</i>	relasi (hubungan) antara sebuah <i>Interface</i> (antarmuka) dengan <i>Class</i> (kelas) yang mengimplementasikan detail dari kontrak tersebut
1 *	<i>Multiplicity</i>	Menunjukkan jumlah relasi. 1 = satu, * = banyak, 0..* = nol atau banyak, 1..* = satu atau banyak.
~	<i>Package</i>	Memecah sistem yang besar atau kompleks menjadi bagian-bagian yang lebih kecil dan mudah dipahami
+	<i>Public</i>	Atribut atau metode yang dapat diakses oleh semua kelas.
-	<i>Private</i>	Atribut atau metode yang hanya dapat diakses di dalam kelas itu sendiri.
#	<i>Protected</i>	Atribut atau metode yang dapat diakses oleh kelas turunan.
/	<i>Derived</i>	Konsep relasi dalam <i>Class Diagram</i> di mana sebuah kelas baru (anak) mewarisi atribut dan metode dari kelas yang sudah ada (induk).
<i>Sequence Diagram</i>		
	<i>Self-Call</i>	Pemanggilan metode di dalam objek yang sama.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Representasi objek atau aktor yang berpartisipasi dalam interaksi (kotak dengan garis putus-putus ke bawah).
	<i>Object Message</i>	Representasi pesan objek itu sendiri
	<i>Control Object</i>	Mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
	<i>Entity Object</i>	Suatu objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan kedalam suatu <i>database</i> .

	<i>Boundary Object</i>	Sebuah objek yang meenjadi penghubung sistem.
<i>Activity Diagram</i>		
	<i>Initial Node</i>	Titik awal (<i>start</i>) dari aktivitas.
	<i>Fork</i>	Memecah alur menjadi paralel
	<i>Join Node</i>	Menggabungkan alur paralel kembali
	<i>Decision Node</i>	Percabangan berdasarkan kondisi (<i>true/false</i>).
	<i>Activity / Action</i>	Aktivitas atau proses yang dieksekusi.
	<i>Swimlane</i>	Memisahkan aktivitas berdasarkan aktor atau departemen.
	<i>Activity Final Node</i>	Titik akhir (<i>end</i>) dari aktivitas.