



**PENGEMBANGAN FITUR LOCATION TRACKER
DENGAN ALGORITMA A SEBAGAI ALTERNATIF
ROUTING PADA APLIKASI VENTOUR MOBILE:
STUDI PERBANDINGAN DENGAN GOOGLE MAPS
ROUTES API**

SKRIPSI

TEUKU MUHAMMAD ERLANGGA PRATAMA 2207412047

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN
KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026



**PENGEMBANGAN FITUR LOCATION TRACKER
DENGAN ALGORITMA A SEBAGAI ALTERNATIF
ROUTING PADA APLIKASI VENTOUR MOBILE:
STUDI PERBANDINGAN DENGAN GOOGLE MAPS
ROUTES API**

SKRIPSI

**Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk
Memperoleh Diploma Empat Politeknik**

TEUKU MUHAMMAD ERLANGGA PRATAMA 2207412047

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN
KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2026

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Teuku Muhammad Erlangga Pratama

NIM : 2207412047

Jurusan/Program Studi : Teknik Informatika dan Komputer / Teknik
Informatika

Judul skripsi : *Pengembangan Fitur Location Tracker dengan
Algoritma A** sebagai Alternatif Routing pada Aplikasi
Ventour Mobile: Studi Perbandingan dengan *Google
Maps Routes API*

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara - cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk - bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 21 Juli 2026

Yang membuat pernyataan



Teuku Muhammad Erlangga Pratama

NIM. 2207412047



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mengutip sumber:
 - a. Pengutipan untuk tujuan pendidikan, penelitian, penerbitan karya ilmiah, penerbitan laporan, penerbitan buku atau jurnal untuk tujuan pendidikan.
 - b. Pengutipan untuk tujuan kepentingan yang wajar Politeknik Negeri Jakarta.
2. Dilarang mengutip dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta.

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Teuku Muhammad Erlangga Pratama

NIM : 2207412047

Program Studi : Teknik Informatika

Judul Skripsi : *Pengembangan Fitur Location Tracker dengan A**
sebagai Alternatif Routing pada Aplikasi Ventour Mobile:
Studi Perbandingan dengan Google Maps Routes API

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Selasa, Tanggal 21
Bulan Juli, Tahun 2026 dan dinyatakan **LULUS**.

Disahkan oleh

Pembimbing I : Bambang Warsuta, S.Kom., M.T.I.

Penguji I : Rizki Elisa Nalawati, S.T., M.T.

Penguji II : Iik Muhamad Malik Matin, S.Kom. M.T.

Penguji III : Fachroni Arbi Murad, S.Kom., M.Kom.

Mengetahui :

Ketua Jurusan Teknik Informatika dan Komputer



DR. Anna Hidayanti, S.Kom., M.Kom.

NIP. 197908032003122003

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI
UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta,

saya bertanda tangan dibawah ini :

Nama : Teuku Muhammad Erlangga Pratama

NIM : 2207412047

Jurusan/Program Studi : T.Informatika dan Komputer / Teknik Informatika

Demi pengembangan ilmu pengetahuan , menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

Pengembangan Fitur Location Tracker dengan Algoritma A* sebagai Alternatif Routing pada Aplikasi Ventour Mobile: Studi Perbandingan dengan *Google Maps Routes API*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.. Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 28 Juli 2026

Yang Menyatakan



Teuku Muhammad Erlangga Pratama

NIM 2207412047

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat-Nya, skripsi ini dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Diploma Empat di Politeknik Negeri Jakarta.

Saya menyadari bahwa proses penyusunan skripsi ini tidak akan berjalan lancar tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak sejak awal masa perkuliahan. Oleh karena itu, saya ingin menyampaikan terima kasih yang tulus kepada:

- a. Bapak Bambang Warsuta, S.Kom., M.T.I., selaku dosen pembimbing yang telah meluangkan waktu, tenaga, dan pikiran untuk mengarahkan saya selama penyusunan skripsi ini.
- b. Bapak Hadyan, selaku pembimbing lapangan yang telah banyak membantu dan memfasilitasi saya dalam memperoleh data-data yang diperlukan.
- c. Bapak Teuku Badrussaman dan Ibu Endang Sulistyorini, selaku orang tua tercinta yang tiada hentinya memberikan dukungan moral, materi, serta doa yang tulus kepada saya.
- d. Teman-teman seperjuangan, yang telah banyak bertukar pikiran dan membantu saya dalam melewati proses ini.
- e. Adinda Marcha selaku pasangan yang senantiasa memberikan *support* kepada penulis.

Akhir kata, semoga Tuhan Yang Maha Esa membalas segala kebaikan semua pihak yang telah membantu. Saya berharap skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan dan penelitian ke depan.

PENGEMBANGAN FITUR *LOCATION TRACKER* PADA APLIKASI *VENTOUR MOBILE*

ABSTRAK

PT Ventura Semesta Wisata (Ventour) merupakan perusahaan layanan pariwisata umrah dan haji dengan jaringan cabang di berbagai kota di Indonesia. Dalam rangka mendukung digitalisasi layanan dan mitigasi risiko operasional di lapangan, penelitian ini mengembangkan fitur Location Tracker pada aplikasi Ventour Mobile berbasis Flutter, yang sebelumnya hanya tersedia pada platform web. Fitur ini dirancang dengan integrasi hak akses (role-based access) yang disesuaikan dengan data paket perjalanan, sehingga setiap Tour Leader hanya dapat memantau jamaah pada pakatnya masing-masing. Selain pelacakan posisi tersinkronisasi, penelitian ini juga mengimplementasikan algoritma A (A-Star) dengan heuristik Haversine dan data jaringan jalan dari OpenStreetMap (Overpass API) untuk menghasilkan rekomendasi rute alternatif, guna mengatasi keterbatasan penggunaan layanan routing berbayar. Pengembangan sistem dilakukan menggunakan metode Waterfall, dengan pendekatan location state based melalui pola manajemen status BLoC untuk menjaga konsistensi data posisi jamaah secara reaktif. Pengujian sistem dilakukan dengan metode black-box testing yang menghasilkan tingkat keberhasilan fungsional sebesar 100%, serta pengujian akurasi lokasi menggunakan perhitungan error Haversine dan waktu respons sistem. Evaluasi usability menggunakan System Usability Scale (SUS) menghasilkan skor 74,6%, termasuk kategori "Good". Hasil penelitian menunjukkan bahwa fitur yang dikembangkan telah memenuhi aspek fungsionalitas, akurasi, dan kenyamanan penggunaan bagi Tour Leader dalam mengoptimalkan pengawasan jamaah secara digital berbasis mobile.*

Kata Kunci: *Location Tracker, Flutter, BLoC, Algoritma A*, Role-Based Access, System Usability Scale, Black-Box Testing.*

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME	ii
LEMBAR PENGESAHAN	Kesalahan! Bookmark tidak ditentukan.
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	iv
KATA PENGANTAR	v
ABSTRAK	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan dan Manfaat	3
1.5 Sistematika Penulisan	4
1.5.1 BAB I PENDAHULUAN	4
1.5.2 BAB II TINJAUAN PUSTAKA	5
1.5.3 BAB III METODOLOGI PENELITIAN	5
1.5.4 BAB IV HASIL DAN PEMBAHASAN	5
1.5.5 BAB V PENUTUP	5
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian terdahulu	6
2.2 Waterfall	9
2.3 GPS	10

2.5 Dart	10
2.6 Flutter	10
2.7 PHP	11
2.8 Code Igniter	11
2.9 Rest API	12
2.10 Google Maps API	13
2.11 State management	13
2.12 BLoC State management System	14
2.13 A* Algorithm	15
2.14 User Acceptance Testing (UAT)	16
2.15 Black-box Testing	16
2.16 System Usability Scale Testing	17
BAB III METODE PENELITIAN	18
3.1 Rancangan Penelitian	18
3.4.1 Teknik Analisis Data	20
3.2 Tahapan Penelitian	28
3.2.1 Analisis Kebutuhan	30
3.2.2 Implementasi Sistem	30
3.2.3 Pengujian Sistem	30
3.2.4 Penulisan Laporan	30
3.3 Objek Penelitian	31
BAB IV HASIL DAN PEMBAHASAN	32
4.1 Analisis Kebutuhan	32
4.2 Perancangan Sistem	33
4.2.1 Use case	33
4.2.2 Activity Diagram mendapat notifikasi	34

4.2.3 <i>Activity Diagram</i> Membaca Koordinat <i>Tour Leader</i> & jamaah	35
4.2.4 <i>Activity Diagram</i> melacak jamaah	36
4.2.5 Rancangan Tampilan Fitur <i>Location tracker</i>	37
4.3 Implementasi Sistem	38
4.3.1 Implementasi Fitur <i>Location tracker</i>	38
4.3.2 <i>A* Algorithm Code</i>	40
4.3.3 GMaps Fetching	42
4.4 Pengujian Sistem	43
4.4.1 Deskripsi Pengujian	43
4.4.2 Prosedur Pengujian	44
4.4.3 Data Hasil Pengujian	44
4.4.4 Evaluasi Pengujian	48
BAB V KESIMPULAN DAN SARAN	59
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA	61
DAFTAR RIWAYAT HIDUP	64
LAMPIRAN	65

DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan Penelitian Terdahulu	6
Tabel 3.1 Daftar Pertanyaan Wawancara.....	18
Tabel 3.2 Skenario Blackbox-Testing.....	20
Tabel 3. 3 Pengujian Respon Time	23
Tabel 3. 4 Perbandingan Mode Rute A* dan Google	24
Tabel 3.5 Bobot Nilai Skala likert	24
Tabel 3.6 Pertanyaan User Acceptance Testing.....	25
Tabel 3.7 Pertanyaan System Usability Scale.....	26
Tabel 4.1 Hasil <i>Blackbox-Testing</i>	44
Tabel 4. 2 Pengujian waktu respon	46
Tabel 4. 3 Perbandingan mode rute Google Maps dan A*	46
Tabel 4. 4 Pengujian User Acceptance Testing	47
Tabel 4.5 Hasil System Usability Scale (SUS)	48
Tabel 4.6 Bobot Penilaian Evaluasi Fungsional Sistem.....	54
Tabel 4.7 Bobot Penilaian Evaluasi Kinerja Sistem	54
Tabel 4.8 Bobot Evaluasi Pengalaman & Tampilan Antarmuka Sistem	55
Tabel 4.9 Bobot Evaluasi Efisiensi & Produktivitas.....	55
Tabel 4.10 Keterangan dari persentase skor	55
Tabel 4.11 Hasil perhitungan Evaluasi Fungsional Sistem.....	56
Tabel 4.12 Hasil perhitungan Evaluasi Kinerja Sistem	56
Tabel 4.13 Hasil perhitungan Evaluasi Pengalaman & Tampilan Antarmuka Sistem.....	56
Tabel 4.14 Hasil perhitungan Evaluasi Efisiensi & Produktivitas.....	57
Tabel 4.15 Hasil Perhitungan SUS	57

DAFTAR GAMBAR

Gambar 2. 1 Metode Pengembangan Software Waterfall.....	9
Gambar 2.2 Ilustrasi Cara Kerja BLoC.....	14
Gambar 3.1 Tahapan Penelitian	29
Gambar 4.1 Use case Location tracker	33
Gambar 4.2 Activity Diagram Masuk dengan Akun Terdaftar.....	34
Gambar 4.3 Activity Diagram membaca koordinat Tour Leader & jamaah.....	35
Gambar 4.4 Activity Diagram melacak Jamaah	36
Gambar 4.5 Mockup Fitur Opsi Location tracker (1)	37
Gambar 4.6 Mockup Fitur Opsi Location tracker (2)	37
Gambar 4.7 Mockup Fitur Opsi Location tracker (3)	37
Gambar 4.8 Mockup Location tracker	37
Gambar 4.9 Implementasi Fitur Opsi Location tracker	38
Gambar 4.10 Implementasi Fitur Opsi Location tracker (2).....	38
Gambar 4.11 Implementasi Fitur Opsi Location tracker (3).....	39
Gambar 4.12 Implementasi Halaman Location tracker	39
Gambar 4. 13 A* Algorithm Snippet Code.....	40
Gambar 4. 14 Snippet Code GMaps Routes Fetching	42
Gambar 4.13 Skala System Usability Scale.....	58

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT Ventura Semesta Wisata (Ventour) merupakan perusahaan yang bergerak di bidang layanan pariwisata dengan fokus kepada ibadah umroh dan haji dengan jaringan cabang di berbagai kota di Indonesia. Dalam upaya digitalisasi layanan, perusahaan telah menghadirkan aplikasi *Ventour mobile*. Saat ini, aplikasi tersebut berperan sebagai pusat informasi utama bagi jamaah, menyediakan fitur mulai dari penawaran paket perjalanan, informasi layanan perusahaan, hingga fitur pendukung ibadah seperti Al-Qur'an digital, kumpulan doa, dzikir, dan jadwal shalat.

Dengan aplikasi yang bertujuan untuk meningkatkan layanan perjalanan, pada aspek ini perusahaan kerap menghadapi tantangan besar pada aspek pengawasan jamaah, khususnya saat sesi aktivitas bebas di luar jadwal utama. Untuk mengatasi risiko jamaah terpisah dari rombongan, Ventour saat ini menerapkan penggunaan perangkat keras (*hardware*) berupa *Mi tag*. Alat ini berfungsi sebagai penanda lokasi untuk membantu *Tour Leader* memastikan keberadaan jamaah dan mempermudah pengumpulan kembali jamaah di titik kumpul yang telah ditentukan. Namun, penggunaan *Mi tag* memiliki kendala teknis signifikan.

Berdasarkan informasi melalui wawancara dengan Supervisor IT Ventour, perangkat ini tidak mampu memperbarui lokasi secara *tersinkronisasi*, dengan jeda pembaruan (*latency*) yang bahkan bisa mencapai 30 menit. Keterlambatan data lokasi ini menimbulkan dampak operasional yang serius: *Tour Leader* kesulitan melacak posisi aktual jamaah yang tersesat, yang berujung pada terhambatnya mobilisasi rombongan ke agenda perjalanan selanjutnya. Situasi ini sering kali menciptakan "drama" pencarian jamaah. Melihat permasalahan tersebut, penulis merancang pengembangan fitur pada aplikasi *Ventour mobile* yang sudah dimiliki jamaah saat ini. Pengembangan fitur *Location tracker* di dalam aplikasi diusulkan sebagai solusi alternatif sekaligus pendukung (*support*

system) bagi penggunaan *Mi tag*. Dengan memanfaatkan GPS pada *smartphone* yang *tersinkronisasi*, fitur ini diharapkan dapat menutupi kelemahan latensi pada perangkat keras, dengan integrasi data diri *Tour Leader* maupun jamaah yang sudah dimiliki oleh *ventour* sehingga meminimalkan gangguan pada jadwal perjalanan. Selain itu, untuk mendukung kebutuhan navigasi rute perjalanan, penelitian ini mengintegrasikan algoritma A* (*A-Star*) sebagai alternatif perhitungan rute berbasis data peta terbuka *OpenStreetMap*, yang dibandingkan dengan layanan *Google Maps Routes API*. Pendekatan ini bertujuan untuk mengevaluasi sejauh mana algoritma pencarian jalur terpendek dapat diterapkan secara lokal pada perangkat *mobile*, sekaligus memberikan opsi navigasi yang tidak sepenuhnya bergantung pada layanan berbayar pihak ketiga, sehingga sistem dapat tetap memberikan estimasi rute yang memadai dalam berbagai kondisi akses layanan.

1.2 Perumusan Masalah

1. Bagaimana rancang bangun fitur *Location tracker* untuk pemantauan lokasi jamaah pada aplikasi *Ventour mobile*?
2. Bagaimana perbandingan estimasi durasi (waktu tempuh) yang dihasilkan oleh algoritma A* dengan durasi yang diberikan oleh *Google Maps API* untuk rute menemukan jamaah?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang ditentukan, berikut batasan masalah dari perancangan sistem:

1. Penelitian dibatasi pada perancangan dan pengembangan fitur *Location tracker* pada aplikasi *Ventour mobile*, dengan integrasi data yang sudah ada mulai dari data *Tour Leader* hingga jamaah yang menjadi satu kelompok pada saat perjalanan berlangsung.
2. Fitur pelacak lokasi (*location tracker*) dimulai setelah jamaah berhasil melakukan autentikasi masuk (*Login*) menggunakan akun yang telah terdaftar pada sistem.

3. Fitur *location tracker* yang dikembangkan mengakses perubahan lokasi jamaah setelah *Login* dan melakukan pergantian koordinat secara sinkron dengan mengambil data koordinat *latitude* dan *longitudinal* dalam *database*.
4. Implementasi mode rute menggunakan algoritma A* (A-star) dalam penelitian ini ditambahkan sebagai perbandingan terhadap metode *fetching* melalui *Google Maps* Routes API serta dilakukan di sisi klien, dan pengujian efektivitasnya dibatasi hanya pada jarak tempuh ≤ 20 km antara *Tour Leader* dan jamaah; performa dan akurasi algoritma di luar jarak tersebut tidak menjadi bagian dari cakupan pengujian penelitian ini.
5. Penerapan algoritma A* (A-Star) dalam pencarian rute dilakukan berdasarkan data graf jaringan jalan yang bersumber dari *Overpass* API (*OpenStreetMap*), sehingga akurasi rute bergantung pada kelengkapan data jalan pada wilayah tersebut.
6. Pada proses pengujian perbandingan antara Routes API dan algoritma A* tersebut, ditemukan kesalahan pemanggilan pada mode rute berjalan kaki (*walking*) melalui Directions API yang menyebabkan estimasi waktu tempuh tidak akurat. Mengingat temuan ini diperoleh setelah masa kerja praktik (*internship*) penulis di PT Ventura Semesta Wisata telah berakhir, penulis tidak lagi memiliki akses untuk melakukan perbaikan langsung terhadap sistem produksi. Oleh karena itu, perbaikan terhadap isu ini didokumentasikan sebagai keterbatasan penelitian sekaligus rekomendasi teknis bagi pihak pengembang internal Ventour untuk ditindaklanjuti.

1.4 Tujuan dan Manfaat

Penelitian ini bertujuan untuk merancang dan membangun fitur Location Tracker pada aplikasi Ventour Mobile yang mampu memantau lokasi jamaah secara real-time, sehingga *Tour Leader* dapat menemukan dan memantau keberadaan jamaah dengan lebih mudah dan efisien selama perjalanan berlangsung. Selain itu, penelitian ini juga bertujuan untuk membandingkan

estimasi durasi (waktu tempuh) yang dihasilkan oleh algoritma A* berbasis data OpenStreetMap dengan durasi yang diberikan oleh *Google Maps* API, sebagai evaluasi terhadap potensi algoritma A* sebagai alternatif maupun pelengkap dalam penentuan rute menuju titik lokasi jamaah. Penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun praktis. Secara teoritis, hasil perbandingan antara algoritma A* dan *Google Maps* API yang diperoleh dalam penelitian ini dapat menjadi referensi bagi penelitian selanjutnya yang mengkaji efektivitas algoritma pencarian jalur terpendek sebagai alternatif layanan peta berbayar dalam pengembangan aplikasi *mobile*.

Secara praktis, penelitian ini memberikan manfaat bagi beberapa pihak yang terlibat secara langsung. Bagi perusahaan pengelola perjalanan (Ventour), fitur *Location Tracker* ini membantu *Tour Leader* dalam memantau posisi seluruh jamaah secara tersinkronisasi melalui aplikasi terpadu, sehingga meminimalkan risiko jamaah terpisah atau tersesat selama perjalanan dan mengurangi gangguan pada jadwal perjalanan yang telah direncanakan. Bagi jamaah, keberadaan fitur ini memberikan rasa aman karena posisi mereka dapat dipantau dan ditemukan dengan lebih cepat oleh *Tour Leader* apabila terjadi kondisi darurat atau pemisahan dari rombongan.

1.5 Sistematika Penulisan

Sistematika penulisan disusun untuk memberi gambaran secara umum mengenai isi dari skripsi ini. Sistematika penyusunan proposal skripsi dijelaskan sebagai berikut:

1.5.1 BAB I PENDAHULUAN

Bab I merupakan bagian pendahuluan dalam penelitian ini yang menguraikan latar belakang penelitian, perumusan masalah dari latar belakang, serta batasan-batasan masalah yang diterapkan dalam penelitian. Selain itu, bab ini juga menjelaskan tujuan yang ingin dicapai serta manfaat dari pelaksanaan penelitian, serta menyajikan sistematika penulisan dalam penyusunan laporan skripsi ini.

1.5.2 BAB II TINJAUAN PUSTAKA

Bab II merupakan tinjauan pustaka dari penelitian ini yang menjelaskan istilah-istilah teknis pada penelitian serta penjelasan tentang teknologi yang penulis gunakan selama pengembangan sistem.

1.5.3 BAB III METODOLOGI PENELITIAN

Bab III merupakan uraian dari metode yang digunakan selama melakukan penelitian ini, uraian tersebut meliputi rancangan penelitian, objek dari penelitian, *framework* yang penulis gunakan dan teknik penulisan, Teknik dari pengumpulan dan analisis data, jadwal pelaksanaan dari penelitian ini, dan rincian biaya dari penelitian ini.

1.5.4 BAB IV HASIL DAN PEMBAHASAN

Bab ini menguraikan hasil utama dari penelitian serta menjawab rumusan masalah, yakni cara pengembangan fitur *Location tracker* dengan sistem pengawasan, serta fungsi navigasi (*Routing*) pada aplikasi Ventour Mobile. Pembahasan dimulai dari tahap perancangan desain antarmuka pengguna (UI/UX) dan pembuatan *mockup* tampilan, hingga tahap implementasi pada aplikasi. Fokus utama fitur ini merupakan sebagai alat bantu *Tour Leader* dalam menemukan jamaah. Serta langkah preventif bagi jamaah, dengan memberikan notifikasi otomatis ketika terdeteksi di luar jangkauan *Tour Leader*.

1.5.5 BAB V PENUTUP

Bab kelima merupakan bagian penutup dari laporan penelitian yang menyajikan kesimpulan dari seluruh proses pengembangan fitur yang telah dilakukan. Selain itu, bab ini juga memuat saran-saran konstruktif sebagai landasan untuk pengembangan aplikasi Ventour Mobile di masa yang akan datang.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian terdahulu

Sistem dibuat dengan dasar beberapa penelitian terdahulu yang relevan dan dapat menerapkan teori dan manfaat yang sama pada aplikasi *mobile* yang sedang pada penelitian saat ini. Perbedaan dan relevansi penelitian terdahulu tercantum pada tabel 2.1.

Tabel 2.1 Tabel Perbandingan Penelitian Terdahulu

No	Nama Peneliti	Judul	Hasil Penelitian	Persamaan	Perbedaan
1.	Muhammad Fakhrol Ulum (Ulum and Pasuruan, 2025)	Pengembangan Aplikasi Mobile Tracking dengan Integrasi GPS Realtime	Aplikasi Mobile Tracking berbasis GPS	Pengembangan aplikasi Mobile Tracking	Penerapan aplikasi, Penelitian ini berorientasi umum. Sedangkan penelitian penulis memiliki penerapan pada layanan perjalanan
2.	(Randi Ramadhani and Sisephaputra, 2026)	Design of Lost Pilgrim Monitoring Application with Android LBS	Aplikasi <i>mobile</i> dan <i>website</i> dengan LBC (<i>Location Based Services</i>)	Pengembangan aplikasi <i>mobile</i> dengan tujuan <i>monitoring</i> jamaah tersesat.	Implementasi penelitian terdahulu fitur komunikasi dua arah dan tombol darurat, namun sistem tersebut bersifat reaktif dan bergantung pada pengguna saat terjadi masalah. Penelitian ini mengisi gap dengan menghadirkan sistem pengawasan proaktif melalui fitur <i>Geofencing Notification</i> , secara otomatis memberi peringatan sebelum jamaah terpisah. Selain itu, penggunaan <i>Routes API</i> memberikan akurasi estimasi rute navigasi yang adaptif.

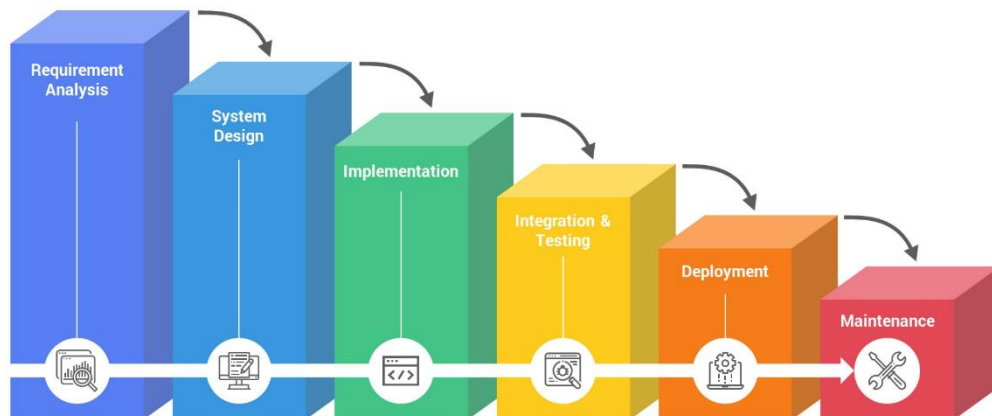
No	Nama Peneliti	Judul	Hasil Penelitian	Persamaan	Perbedaan
3.	Safinatun Najah (Safinatun Najah et al., 2025)	Rancang Bangun Aplikasi Mobile dengan GPS untuk Presensi WFO dan WFS	Aplikasi Mobile untuk presensi	Pengembangan aplikasi Mobile dengan fitur <i>location</i> untuk validasi	Penelitian terdahulu mengeksplorasi pemanfaatan teknologi lokasi untuk kebutuhan validasi presensi. Sedangkan penggunaan penelitian penulis untuk memitigasi risiko jamaah yang terpisah dari rombongan
4.	Andre Setiawan, Agung Tri Prastowo, Dedi Darwis (Setiawan et al., 2022)	Sistem <i>Monitoring</i> Keberadaan Posisi Mobil berbasis Gps dan Penyadap Suara Menggunakan <i>Smartphone</i>	Sistem integrasi IOT dengan Arduino Uno R3 dengan fitur untuk mengakses lokasi mobil, serta aksi telepon sesuai dengan situasi dan <i>command</i> dari pesan telegram	Pengembangan aplikasi dengan fitur <i>Location tracker</i>	Perbedaan mendasar terletak pada subjek pemantauan; penelitian sebelumnya difokuskan pada pelacakan kendaraan (mobil), sedangkan penelitian penulis digunakan untuk memantau keberadaan jamaah.
5.	Ahmad Shofi Baharrudin, Sunu Jatmika (Shofi Baharuddin and Jatmika, 2024)	Aplikasi <i>Monitoring</i> Lokasi Anak Menggunakan <i>Location Based Service</i> Dengan History Tracking Lokasi Berbasis Android	Aplikasi berbasis android dengan fitur pelacakan lokasi <i>tersinkronisasi</i>	Pengembangan sebuah aplikasi dengan fitur <i>Location tracker</i>	Latar belakang masalah yang berbeda meskipun penelitian memiliki kegunaan serta pendekatan serupa.

Kesimpulan yang diperoleh berdasarkan uraian tabel 2.1, penelitian penulis merupakan pengembangan fitur aplikasi *Location tracker* dengan mengintegrasikan data mengenai perbedaan *role* pengguna dan data-data pendukung lainnya menjadikannya sebuah sistem yang terpadu. Mayoritas penelitian terdahulu (Ulum and Pasuruan, 2025), (Setiawan et al., 2022), (Shofi Baharuddin and Jatmika, 2024), (Safinatun Najah et al., 2025) memiliki perbedaan pada ranah implementasi dengan menggunakan layanan *location*. Sementara itu, pada penelitian dengan studi kasus serupa (Randi Ramadhani and Sisephaputra, 2026) penelitian mengisi *gap* pendekatan memunculkan notifikasi dari reaktif ke proaktif. Sistem terdahulu bergantung pada inisiatif pengguna (*panic button*), penelitian ini mengedepankan fungsi pencegahan melalui fitur *Geofencing Notification* yang memberikan peringatan otomatis sebelum jamaah terpisah terlalu jauh setelah jamaah melakukan login dan berada pada aplikasi Ventour Mobile. Selain itu, penggunaan *Routes API* memberikan keunggulan teknis dalam penyediaan navigasi yang adaptif terhadap kondisi lalu lintas, yang membedakan penelitian ini dari metode LBS.

Dengan perbedaan serta persamaan yang telah dijelaskan. Implementasi rancangan dalam penelitian ini bertujuan untuk menyelesaikan permasalahan dari jamaah yang hilang setelah sesi *free time*. Algoritma pengembangan fitur *Location tracker* menggunakan *state-based location tracking* yaitu pengaksesan lokasi *smartphone* masing-masing jamaah hanya ketika kondisinya terpencar setelah sesi bebas melalui aplikasi *Ventour mobile*. Dengan catatan, jamaah harus membuka aplikasi serta *Login* akun jamaah. Aplikasi ini terintegrasi dengan *database* milik ventour, hal ini memungkinkan sistem menggunakan informasi dari peran (*role*), data diri dan paket perjalanan yang sedang berlangsung.

2.2 Waterfall

Waterfall SDLC Model



Gambar 2. 1 Metode Pengembangan Software Waterfall

Metode *Waterfall* dikenal sebagai pendekatan pengembangan sistem yang bersifat linier dan sekuensial, di mana setiap tahapan pengembangan dilaksanakan secara berurutan dan harus diselesaikan sepenuhnya sebelum melanjutkan ke tahapan berikutnya. Sebagaimana dikemukakan dalam literatur, metode *Waterfall* menawarkan kerangka kerja yang terstruktur melalui tahapan-tahapan utama yang meliputi analisis kebutuhan (*requirement analysis*), perancangan (*design*), implementasi (*implementation*), pengujian (*testing*), penerapan (*deployment*) dan pemeliharaan (*maintenance*), yang dapat diterapkan dalam pengembangan berbagai jenis sistem perangkat lunak (Sallu et al., 2023).

Dalam konteks penelitian ini, metode *Waterfall* dipilih sebagai pendekatan pengembangan fitur *Location Tracker* pada aplikasi *Ventour Mobile* dengan mempertimbangkan karakteristik proyek yang sesuai dengan kekuatan utama metode tersebut. Kebutuhan sistem telah terdefinisi secara jelas sejak awal pengembangan mencakup kebutuhan pelacakan lokasi jamaah secara *real-time*, integrasi algoritma *A** sebagai alternatif *routing*, serta kebutuhan antarmuka pengguna untuk *Tour Leader* sehingga tidak terdapat perubahan kebutuhan yang signifikan selama masa pengembangan berlangsung. Kondisi ini selaras dengan karakteristik utama metode *Waterfall* yang bekerja paling optimal

ketika ruang lingkup dan spesifikasi sistem telah dipahami secara menyeluruh sebelum proses implementasi dimulai, karena setiap perubahan kebutuhan di tengah pengembangan berpotensi mengganggu keseluruhan alur tahapan yang bersifat sekuensial. Tahapan pengembangan yang digunakan pada penelitian ini, hanya sampai tahap pengujian karena masih bersifat pengembangan serta penerapan serta pemeliharaan ditanggung atau menjadi hak pihak perusahaan.

2.3 GPS

Global Positioning System (GPS) merupakan sistem navigasi berbasis satelit dengan ketersediaan 24 jam sehari. Dengan ketersediaan 24 jam dalam sehari, layanan ini dapat digunakan untuk berbagai keperluan seperti estimasi waktu tempuh dengan memantau situasi lalu lintas, menentukan rute perjalanan, maupun mencari tempat khusus sepanjang perjalanan (Topete et al., 2024). Penggunaan GPS dalam penelitian kali ini adalah sebagai dasar teknologi dari fitur *Location tracker*, fitur ini diakses melalui *Google Maps API*.

2.5 Dart

Dart merupakan sebuah bahasa pemrograman *open source* yang dikembangkan oleh Google. *Dart* mendukung pengembangan aplikasi dari sisi klien maupun *server*. Mayoritas penggunaan *dart* adalah untuk pengembangan *Android apps*, *IOS apps*, *web applications* menggunakan *framework Flutter* (Vindua et al., 2024). Penerapan *dart* dalam project ini adalah sebagai bahasa dasar yang diakses melalui *framework Flutter*.

2.6 Flutter

Flutter merupakan *cross-platform development frameworks* yang memungkinkan satu *source code* berjalan pada alat maupun platform yang berbeda (Kinari et al., 2025). *Flutter* dibangun menggunakan *Dart*. Kemampuan *Dart* untuk mendukung berbagai arsitektur, menjadikan *Flutter* sebagai *framework* untuk membangun aplikasi *cross-platform* dengan performa tercepat saat ini yang memiliki performa seperti membangun aplikasi secara

native. Dengan kemampuan ini, *Flutter* dapat mempersingkat waktu pengerjaan dan menjadi solusi pengembangan aplikasi *cross-platform* secara cepat tanpa mengorbankan performa maupun kualitas visual aplikasi .

Penggunaan *Flutter* pada penelitian ini terletak pada pengembangan aplikasi untuk multi-platform yaitu android dan IOS, sebagai solusi untuk mengakses lokasi dari jamaah maupun *Tour Leader* yang sebelumnya terbatas saat menggunakan PWA.

2.7 PHP

PHP: *Hypertext Processor* adalah bahasa pemrograman *open source* yang dirancang untuk pengembangan aplikasi berbasis web. Mengutamakan pendekatan pengembangan aplikasi dengan cepat, fleksibel dan pragmatis, PHP mendukung pengembangan *website* mulai dari blog hingga *website* paling populer di dunia (Apiag et al., 2023). Dengan kemampuan kecepatan pengembangan sistem, implementasi *Ventour mobile* menggunakan PHP sebagai bahasa pemrograman untuk membuat PWA (*Progressive Web App*) dengan kemudahan akses bagi pengguna melalui berbagai platform mulai dari Android, IOS, hingga *Website*.

2.8 CodeIgniter

CodeIgniter merupakan *framework* pengembangan aplikasi dengan *footprint* yang kecil untuk mengembangkan sistem dengan fitur yang kuat dan menyediakan *toolkit* sederhana untuk mengembangkan aplikasi dengan fitur lengkap (Qolbi et al., 2025). *CodeIgniter* memungkinkan pengembangan aplikasi berjalan lebih cepat dibanding menulis *code* dari awal tanpa bantuan *framework*, dengan menyediakan *libraries* untuk berbagai tugas umum, tampilan sederhana dan struktur logika untuk mengakses *library* ini. Dengan berbagai kelebihan dari *framework CodeIgniter*, pengembangan PWA (*Progressive Web App*) dari *Ventour mobile* menggunakan *CodeIgniter* sebagai

framework utama untuk menyediakan sistem berbasis *website* yang ramah pengguna mulai dari tampilan hingga logika untuk mengelola data.

2.9 Rest API

REST API, atau *Representational State Transfer Application Programming Interface*, merupakan standar arsitektur layanan web yang memfasilitasi komunikasi data antara klien dan *server* melalui protokol HTTP. Dalam model ini, setiap data dikelola sebagai *resources* yang dapat diakses melalui alamat unik yang disebut *endpoint*. Proses komunikasi diawali dengan pengiriman permintaan (*request*) dari sisi klien yang memuat metode HTTP tertentu, *metadata* pada bagian *header*, serta beban data (*payload*)—umumnya dalam format JSON—pada bagian *body*. Sebagai timbal balik, *server* akan memproses permintaan tersebut dan mengirimkan respons yang disertai dengan kode status HTTP. Kode status ini berfungsi sebagai indikator hasil pemrosesan, seperti kategori 2xx untuk keberhasilan atau 4xx dan 5xx untuk menandakan adanya kendala pada sisi klien maupun *server*. Melalui mekanisme yang terstruktur, REST API menjadi solusi efisien untuk mengintegrasikan berbagai sistem yang berbeda secara konsisten. Dengan pendekatan ini, REST API menjadi cara efisien untuk menghubungkan sistem berbeda dan memungkinkan komunikasi data secara terstruktur melalui protokol HTTP (Kim et al., 2022).

JSON (*JavaScript Object Notation*) merupakan format pertukaran data berbasis teks yang ringan, mudah dipahami oleh manusia, serta efisien untuk diproses oleh sistem komputer. Secara struktural, JSON memiliki format informasi melalui pasangan kunci (*key*) bertipe string dan nilai (*value*) yang fleksibel. Nilai tersebut dapat berupa tipe data dasar seperti string, angka, dan boolean, maupun struktur yang lebih kompleks seperti dalam bentuk objek atau *array* (Garg et al., 2025). Pada penelitian ini, REST API berfungsi sebagai penghubung antara aplikasi dan layanan *backend* untuk pengaksesan data dengan metode *HTTP* seperti GET, POST. Aplikasi *mobile* berperan sebagai klien yang mengirimkan *request*, kemudian *request* diterima oleh layanan

backend yaitu *server* kemudian *server* akan memberikan *response* dalam bentuk JSON lalu data akan digunakan pada aplikasi *mobile*.

2.10 Google Maps API

Google Maps merupakan pengembangan dari Google yang menyediakan layanan peta dengan fitur *update* kondisi lalu lintas secara *tersinkronisasi* yang mendukung navigasi pengguna saat berkunjung ke destinasi baru . Sedangkan, API merupakan teknologi yang dapat menghubungkan kedua perangkat lunak melalui berbagai aturan. Google menyediakan layanan API, untuk pengembang melakukan integrasi pemetaan dan fitur berbasis lokasi ke dalam web atau aplikasi . Penulis mengintegrasikan *Google Maps* API dengan tujuan fasilitas peta interaktif bagi pengguna. Layanan navigasi yang menghubungkan titik koordinat *Tour Leader* dengan Jamaah dikembangkan dengan memadukan *Directions API* dan *Routes API*. Kedua layanan tersebut memiliki karakteristik yang berbeda dalam merepresentasikan rute, terutama penyediaan estimasi kondisi lalu lintas.

2.11 State management

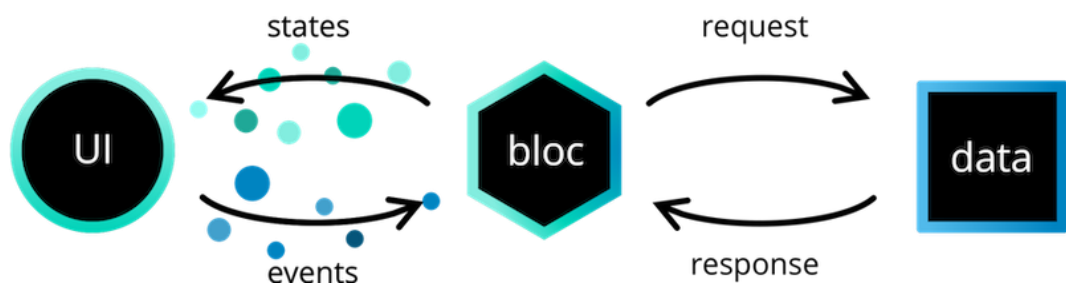
State management dalam *Flutter* adalah metode untuk mengatur data dinamis (*state*) yang menentukan tampilan antarmuka pengguna (UI). Setiap komponen UI yang bersifat dinamis memerlukan *state* tersendiri, sehingga ketika data berubah, pengembang cukup memodifikasi *state* tersebut dan memberi sinyal kepada *framework* bahwa terjadi perubahan. *Flutter* memiliki pendekatan UI deklaratif, pengembang hanya mendeskripsikan bagaimana UI seharusnya tampak pada kondisi saat ini, sementara *framework* secara otomatis menangani transisi antar *state*. *State* berisi seluruh data yang dibutuhkan untuk membangun UI pada suatu momen tertentu. Ketika *state* berubah, *widget* akan menyusun ulang deskripsinya; *framework* kemudian membandingkan perbedaan deskripsi untuk menentukan perubahan minimal yang diperlukan pada struktur *render* (Abdillah et al., 2023).

Pentingnya *state management* dalam *Flutter* terletak pada kemampuan mengelola data dinamis sehingga UI dapat merespons perubahan secara efisien. Dengan pendekatan ini, hanya *widget* yang *state*-nya berubah yang melakukan *rebuild*, sehingga beban pemrosesan berkurang dan performa aplikasi tetap optimal. Dalam penelitian ini, *state management* diterapkan untuk memisahkan logika bisnis (*logic*) dari tampilan (*view*), sehingga kode yang dihasilkan lebih *modular* dan *reusable*.

2.12 BLoC State management System

Business Logic Component (BLoC) adalah arsitektur untuk memisahkan logika bisnis dari presentasi. Penggunaan pola ini bertujuan untuk mengelola *state* dalam aplikasi dengan menggunakan *event* untuk melakukan perubahan pada *state*. Dengan menggunakan *event* untuk melakukan perubahan pada *state* dan untuk merepresentasikan status aplikasi saat ini (Zulistiyan et al., 2024). Dengan konsep dasar *event* dan *state*:

1. *Event*: Interaksi pengguna yang diterjemahkan sebagai *event* untuk diproses oleh BLoC.
2. *State*: Kondisi atau keadaan aplikasi yang dipengaruhi oleh *event*.



Gambar 2.2 Ilustrasi Cara Kerja BLoC

BLoC diterapkan karena memiliki pemisahan *class* unggul dengan memisahkan antara business logic bahkan *input* dan *output*, menggunakan arsitektur yang mirip dengan MVVM, serta menyediakan mekanisme *built-in dependency injection* untuk BLoC memungkinkan *reusable-code* dan sangat baik untuk *scalability* serta *testability*.

2.13 A* Algorithm

A* (A-Star) Search Algorithm merupakan salah satu algoritma pencarian jalur (pathfinding) dan penelusuran graf (graph traversal) yang paling banyak digunakan dan terbukti efektif dalam berbagai bidang, mulai dari navigasi robotik hingga sistem informasi geografis (Liu et al., 2022). Sebagaimana dikemukakan dalam literatur, implementasi dan analisis performa algoritma A* dalam konteks penentuan jalur terpendek masih menjadi fokus penelitian yang relevan hingga saat ini.

Secara prinsip, A* bekerja dengan menggabungkan dua komponen utama dalam menentukan prioritas eksplorasi node, yaitu $g(n)$ yang merepresentasikan biaya nyata dari titik awal ke node yang sedang dievaluasi, dan $h(n)$ yang merupakan fungsi heuristik berupa estimasi biaya dari node tersebut menuju tujuan akhir. Kombinasi keduanya membentuk fungsi evaluasi:

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

Nilai $f(n)$ inilah yang digunakan sebagai dasar prioritas eksplorasi — node dengan nilai $f(n)$ terkecil akan dieksplorasi lebih dahulu, sehingga algoritma cenderung bergerak ke arah tujuan secara lebih terarah dibandingkan algoritma Dijkstra yang mengeksplorasi semua node secara merata ke segala arah dari titik awal.

Pada penelitian ini, A* dipilih sebagai algoritma pencarian rute yang dijalankan sepenuhnya di sisi klien (*on-device*) pada perangkat mobile. Pemilihan ini didasarkan pada karakteristik utama A* yang menjadikannya lebih sesuai untuk dieksekusi di lingkungan dengan sumber daya terbatas dibandingkan algoritma sejenis. Dengan adanya fungsi heuristik yang memberikan estimasi sisa jarak ke tujuan — dalam penelitian ini menggunakan **jarak Haversine** (jarak garis lurus antara dua koordinat di permukaan bumi sebagai pendekatan *great-circle distance*) — algoritma A* dapat memprioritaskan eksplorasi node yang secara geometris lebih mendekati tujuan, sehingga jumlah node yang perlu dievaluasi menjadi jauh lebih sedikit dibandingkan pendekatan *brute-force* seperti Dijkstra pada graf berukuran

besar. Adapun data jaringan jalan yang digunakan sebagai graf masukan algoritma A* pada penelitian ini bersumber dari OpenStreetMap melalui Overpass API, di mana setiap persimpangan jalan direpresentasikan sebagai node dan setiap ruas jalan direpresentasikan sebagai *edge* berbobot yang nilainya dihitung menggunakan jarak Haversine antar node yang berdekatan.

2.14 User Acceptance Testing (UAT)

User Acceptance Testing merupakan tahap penting dalam proses pengujian perangkat lunak. UAT dilakukan untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi kebutuhan dan harapan pengguna sebenarnya sebelum diluncurkan secara resmi (Aliyah Aliyah et al., 2024). *User Acceptance Testing* dibedakan menjadi 2 bagian, yaitu *Blackbox-Testing (Error Haversine, Response Time)* dan *Beta Testing. Blackbox-Testing (Error Haversine, Response Time)* dilakukan di lingkungan pengembang (*developer*).

Tahapan berikutnya adalah *User Acceptance Testing* untuk mengidentifikasi kinerja serta manfaat dari fitur sebelum diimplementasikan. Penyusunan pertanyaan kuesioner dibuat berdasarkan 5 komponen kualitas, yaitu, fungsionalitas sistem akan mengukur fungsi fitur sudah sesuai kebutuhan. Kinerja sistem mengukur kecepatan kinerja sistem mendukung pengguna dalam waktu yang efisien. Pengalaman dan tampilan mengukur kepuasan pengalaman pengguna dalam menggunakan fitur. Efisiensi dan produktivitas mengukur peningkatan efek produktivitas dari penggunaan fitur. Serta keamanan dan keandalan mengukur tingkat kepuasan pengguna dalam menggunakan fitur (Aliyah Aliyah et al., 2024).

2.15 Black-box Testing

Blackbox testing merupakan pengujian sistem yang menitikberatkan pada validasi fungsionalitas sistem dengan fokus utama keluaran (*output*) yang dihasilkan, tanpa meninjau struktur kode internal. Validasi fitur berdasarkan kesesuaian *output* sebagai parameter utama (Wang et al., 2025). Penelitian ini menggunakan metode *black-box* untuk menguji fitur dengan fokus utama

adalah keluaran (*output*) yang dihasilkan, alasan penggunaan metode ini dikarenakan keterbatasan kemampuan pengembang dalam melakukan proses pengujian fitur.

2.16 System Usability Scale Testing

System Usability Scale (SUS) adalah salah pendekatan yang digunakan untuk mengukur tingkat belajar serta kepuasan saat menggunakan suatu sistem atau produk berdasarkan perspektif pengguna. Pendekatan ini menggunakan 10 pertanyaan dasar dengan respons skala likert, Instrumen ini terdiri dari 10 pernyataan sederhana yang dijawab dengan skala likert 5 poin, mulai dari 1 (sangat tidak setuju) hingga 5 (sangat setuju) (Arjiansa and Sutabri, 2023). Penggunaan *System Usability Scale* sebagai parameter dari kenyamanan penggunaan fitur *location tracker* menurut *Tour Leader* sebagai pengguna utama.

BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini berfokus pada pelacakan jamaah yang terpisah pasca sesi bebas di PT Ventura Semesta Wisata melalui pengembangan fitur *Location tracker*. Dengan memanfaatkan *framework Flutter*, fitur ini diintegrasikan dengan aplikasi Ventour Mobile (PWA) sebagai aplikasi migrasi. Dengan objek utama penelitian merupakan aplikasi *mobile* dengan fitur *location tracker* dengan data yang sinkron dengan peran (*role*) *Tour Leader* untuk memantau jamaah pada paket perjalanan yang aktif dalam fitur *Location tracker* pada aplikasi *Ventour mobile*. Memiliki pengguna utama *Tour Leader*, serta jamaah sebagai pengguna pasif.

Secara metodologis, penelitian ini menggunakan pendekatan metode kualitatif dan kuantitatif. Metode kualitatif dilakukan untuk pengumpulan data awal melalui teknik wawancara, serta berdasarkan observasi terhadap *platform* serupa seperti Grab Maps, Waze. Kemudian, metode kuantitatif untuk pengujian fitur aplikasi dengan metode *User Acceptance Testing* serta *SUS*. Hasil dari observasi dan wawancara bertujuan untuk meninjau fitur yang harus diimplementasikan serta bagaimana membuat antarmuka yang dapat meningkatkan pengalaman pengguna.

Pengumpulan data dilakukan dengan metode kualitatif untuk mendapatkan pemahaman mengenai kebutuhan dan situasi sebenarnya dalam merumuskan masalah. Wawancara dilakukan pada tahap pra-pengembangan (fase analisis kebutuhan & masalah). Pada tahap proposal, disusun penjelasan rancangan pertanyaan untuk wawancara.

Tabel 3.1 Daftar Pertanyaan Wawancara

No	Daftar Pertanyaan
1.	Sebagai awalan, ingin bertanya perangkat apa yang digunakan untuk memastikan lokasi jamaah saat tour berlangsung?

No	Daftar Pertanyaan
2.	Apa ada kendala ketika penggunaan alat <i>Location tracker</i> sebelumnya hingga perlu menggantikan dengan fitur aplikasi?
3.	Apa ekspektasi dari aplikasi yang memiliki <i>Location tracker</i> dari PT Ventura Semesta Wisata?
4.	Untuk keperluan data pendukung, apa memungkinkan untuk saya meminta estimasi atau bahkan riwayat data pengeluaran operasional untuk pengadaan atau penyewaan alat <i>Location tracker</i> pak?
5.	Bolehkah saya meminta estimasi atau perhitungan penggunaan <i>Google Maps</i> api dalam penggunaan <i>Location tracker</i> ?
6.	Rencananya, kapan fitur ini digunakan apakah hanya dalam jadwal kegiatan tertentu? atau bersifat <i>tersinkronisasi</i> 24 jam?
7.	Siapa saja target pengguna yang berinteraksi langsung dengan fitur ini?
8.	Mengingat penggunaan aplikasi di luar negeri, apa ada kriteria area atau kondisi geografis tertentu saat fitur harus bekerja secara optimal?
9.	Untuk pengembangan fitur ini, apa ada arahan <i>tech stack</i> khusus yang akan digunakan?
10.	kenapa harus pake <i>Location tracker</i> buatan venom di saat banyak aplikasi <i>Location tracker</i> (Latar belakang harus kustomisasi fitur <i>location</i>)

Penelitian ini menggunakan metode analisis kualitatif pada tahap pra pengembangan serta analisis kuantitatif pada tahap pasca pengembangan. Teknik analisis kualitatif digunakan untuk mengidentifikasi kebutuhan sistem (*requirement gathering*). Data diperoleh melalui observasi lapangan dan wawancara dengan pihak operasional PT Ventura Alam Semesta terkait kendala teknis pada beberapa perjalanan sebelumnya.

Setelah sistem selesai diimplementasikan, penelitian menggunakan pendekatan kuantitatif untuk menguji kelayakan dan tingkat penerimaan sistem menggunakan teknik analisis deskriptif. Data kuantitatif diperoleh dari

responden *Tour Leader* melalui instrumen kuesioner yang dibagi menjadi dua bagian utama:

1. **Analisis Variabel UAT (Skala likert):** Data dari kuesioner *User Acceptance Test* (UAT) diolah menggunakan skala likert 1-5. Proses analisis dilakukan dengan menghitung nilai rata-rata (*mean*) pada lima variabel utama, yaitu: (1) Fungsionalitas, (2) Kinerja, (3) Pengalaman & Antarmuka, (4) Efisiensi, dan (5) Keamanan. Hasil perhitungan ini akan menentukan persentase kelayakan sistem secara fungsional.
2. **Analisis System Usability Scale (SUS):** Secara spesifik, aspek *usability* diukur menggunakan instrumen standar SUS yang terdiri dari 10 butir pernyataan. Teknik analisis data dilakukan dengan mengonversi skor jawaban responden menjadi nilai akhir berskala 0–100 berdasarkan aturan kalkulasi standar SUS. Skor ini kemudian diinterpretasikan ke dalam *Adjective Rating* untuk menentukan tingkat kemudahan penggunaan aplikasi oleh *Tour Leader*.

3.4.1 Teknik Analisis Data

Berdasarkan penjelasan di atas, untuk pengujian kelayakan sistem, penulis menggunakan *Blackbox-Testing* (*Error Haversine, Response Time*), *User Acceptance Testing* serta *System Usability Test*. Pengujian *Blackbox-Testing* (*Error Haversine, Response Time*) dilakukan dengan metode *blackbox testing* bertujuan memastikan keluaran (*output*) fitur telah sesuai. Berikut merupakan rancangan pengujian aplikasi dengan *Blackbox-Testing* terlampir pada tabel 3.2.

Tabel 3.2 Skenario *Blackbox-Testing*

No	Skenario	Informasi	Langkah	Hasil yang diharapkan
1.	Mengakses data lokasi jamaah	<i>Tour Leader</i> telah <i>Login</i> dan mengakses	klik tombol ‘ <i>Track Lokasi</i> ’	Aplikasi menampilkan Lokasi jamaah dengan <i>Marker</i> .

No	Skenario	Informasi	Langkah	Hasil yang diharapkan
		fitur <i>location tracker</i>		
2.	Menggunakan mode rute	<i>Tour Leader</i> telah <i>Login</i> dan telah berada di halaman <i>location tracker</i>	Pilih mark jamaah, opsi kendaraan lalu klik tombol 'Tunjukkan Rute'	Aplikasi menampilkan rute antara titik <i>Tour Leader</i> dengan jamaah
3.	Update lokasi jamaah saat mode rute	<i>Tour Leader</i> telah <i>Login</i> dan berada dalam mode rute.	Klik 'Track Lokasi' setelah memilih <i>Marker</i> jamaah	Aplikasi <i>mengupdate</i> lokasi jamaah
4,	Memunculkan notifikasi ketika ada jamaah di luar radius pengawasan	Notifikasi muncul dengan angka jamaah yang berada di luar radius pengawasan	Klik tombol 'Track Lokasi' ketika ada jamaah yang berada di luar area pengawasan	Aplikasi menampilkan notifikasi dengan angka jamaah yang berada di luar radius pengawasan
5.	Memunculkan notifikasi sebagai jamaah ketika berada terlalu jauh dari <i>Tour Leader</i>	Notifikasi berada di luar radius pengawasan muncul	<i>Login</i> dengan akun jamaah	Aplikasi menampilkan notifikasi berada di luar radius pengawasan

No	Skenario	Informasi	Langkah	Hasil yang diharapkan
6.	Track Jamaah saat lokasi mati	Posisi fitur lokasi mati	Ke menu <i>location tracker</i> saat fitur lokasi mati	Meminta akses lokasi
7.	Track Jamaah saat lokasi menyala	Posisi fitur lokasi menyala	Ke menu <i>location tracker</i> saat fitur lokasi menyala	<i>Marker</i> jamaah langsung tampil
8.	Track jamaah saat tidak ada internet	Kuota internet ataupun tidak terkoneksi dengan jaringan	Akses aplikasi saat tidak memiliki akses internet	Muncul halaman pemberitahuan bahwa “Anda sedang <i>offline</i> ”

Ketika skenario mendapatkan hasil yang sesuai dengan ketentuan awal maka *testing* yang dilakukan telah berhasil. Pengukuran keberhasilan *Blackbox-Testing (Error Haversine, Response Time)* dihitung dengan rumus:

$$\text{Persentase Keberhasilan} = \frac{\sum \text{Skenario berhasil}}{\sum \text{Skenario Uji}} \times 100\%$$

Selain pengujian terhadap beberapa skenario fungsional tersebut, dilakukan pula pengujian akurasi lokasi dan waktu respon sistem. Pengujian akurasi koordinat lokasi dilakukan dengan membandingkan hasil koordinat dari sistem terhadap titik referensi yang diperoleh secara manual melalui aplikasi *Google Maps* (dengan metode *tap-and-hold*). Metode pengambilan titik referensi secara manual ini berpotensi menimbulkan margin kesalahan tambahan akibat keterbatasan presisi sentuhan pada layar perangkat, sehingga hasil perbandingan akurasi koordinat dalam penelitian ini perlu dipahami dengan mempertimbangkan

toleransi kesalahan dari metode pengujian itu sendiri, bukan murni mencerminkan akurasi sistem semata.

Sementara itu, pengujian waktu respon dilakukan dengan mengukur durasi yang dibutuhkan sistem untuk menampilkan marker lokasi jamaah menggunakan *stopwatch*, terhitung dari saat data lokasi diterima hingga marker tampil pada peta. Untuk memperoleh gambaran yang lebih komprehensif mengenai performa sistem dalam kondisi jaringan yang bervariasi, pengujian ini dilakukan pada dua kondisi koneksi internet yang berbeda, yaitu jaringan seluler (mobile data) dan jaringan Wi-Fi. Pemilihan kedua kondisi jaringan ini didasarkan pada kenyataan bahwa pengguna aplikasi di lapangan — baik *Tour Leader* maupun jamaah — tidak selalu berada dalam jangkauan jaringan Wi-Fi yang stabil, sehingga pengujian pada kondisi jaringan seluler dianggap merepresentasikan skenario penggunaan nyata yang lebih realistis, sementara pengujian pada kondisi Wi-Fi berfungsi sebagai pembandingan dengan kondisi jaringan yang lebih stabil dan terkontrol.

Spesifikasi kecepatan masing-masing jaringan yang digunakan dalam pengujian ini — meliputi kecepatan upload dan download — diukur sebelum pelaksanaan pengujian dan akan disajikan pada tabel pengujian sebagai konteks pendukung hasil yang diperoleh. Perbedaan kecepatan jaringan ini diharapkan dapat menunjukkan sejauh mana kondisi konektivitas mempengaruhi waktu respon sistem secara keseluruhan, sehingga hasil pengujian dapat memberikan gambaran performa yang lebih representatif terhadap berbagai kondisi penggunaan di lapangan.

Tabel 3. 3 Pengujian Respon Time

No.	Skenario	Waktu respon (Jaringan seluler)	Waktu Respon (Wifi)
1.	Lokasi Jamaah Muncul		
2.	<i>Update</i> Posisi		
3.	Mode Rute (A*)		

No.	Skenario	Waktu respon (Jaringan seluler)	Waktu Respon (Wifi)
4.	Mode Rute (Gmaps)		

Setelah dilakukan pengujian waktu respon terhadap penampilan marker lokasi jamaah, tahap pengujian berikutnya berfokus pada perbandingan antara metode *fetching* rute melalui *Google Maps Routes* API dan metode pencarian rute menggunakan algoritma A berbasis data *OpenStreetMap*, untuk mengevaluasi selisih akurasi jarak, durasi dari kedua pendekatan tersebut.

Tabel 3. 4 Perbandingan Mode Rute A* dan Google

No	Awal	Tujuan	Mode	Jarak A* (Km)	Jarak Google (Km)	Selisih	Durasi (A*)	Durasi Google	Selisih
1.									

Tahapan selanjutnya, *User Acceptance Testing* dilakukan dengan menyebar kuesioner berdasarkan respons skala likert dengan indikator sebagai acuan responden antara lain.

Tabel 3.5 Bobot Nilai Skala likert

Indikator	Keterangan	Bobot Nilai
SS	Sangat Setuju	5
S	Setuju	4
CS	Cukup Setuju	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Pada tabel di bawah merupakan daftar pertanyaan kuesioner terdiri dari 5 variabel, yaitu : (1). Fungsionalitas sistem (2). Kinerja sistem (3). Pengalaman & tampilan antarmuka sistem dan 4). Efisiensi & produktivitas. Pada penelitian ini pengujian aspek keamanan dan

keandalan sistem tidak dilakukan dengan sebab penelitian selesai setelah masa umrah berakhir.

Tabel 3.6 Pertanyaan *User Acceptance Testing*

No	Variabel	Pertanyaan	(P)
1	Evaluasi Fungsional Sistem	Pengguna merasa fitur ini sudah sesuai kebutuhan	A1
2		Fitur <i>location tracker</i> mampu menampilkan seluruh penanda (<i>Marker</i>) lokasi jamaah yang berada di bawah tanggung jawab saya dengan jelas.	A2
3	Evaluasi Fungsional Sistem	Pilihan mode transportasi (motor, mobil, berjalan) menyesuaikan kebutuhan di lapangan	A3
4	Evaluasi Kinerja Sistem	Fitur ini selalu tersedia ketika dibutuhkan	B1
5		Rute navigasi yang ditampilkan sistem akurat dalam memandu saya menemukan jamaah yang terpisah.	B2
6		<i>Error</i> dalam sistem diatasi dengan baik	B3
7	Evaluasi Pengalaman & Tampilan Antarmuka Sistem	Tampilan mudah dimengerti	C1
8		Fitur dapat dioperasikan dengan mudah	C2
9		Penggunaan fitur mudah diingat	C3
10	Evaluasi Efisiensi & Produktivitas	Fitur mempersingkat tahapan pencarian jamaah	D1
11	Evaluasi Efisiensi & Produktivitas	Fitur mempercepat proses pengumpulan jamaah	D2

No	Variabel	Pertanyaan	(P)
		dibandingkan metode manual atau penggunaan alat pelacak sebelumnya (<i>Mi Tag</i>).	
12		Pengambilan keputusan perjalanan menjadi lebih cepat	D3

Berdasarkan data kuesioner yang diperoleh dari para responden, tahap analisis dilanjutkan dengan menghitung nilai rata-rata (*mean*) dari setiap butir pernyataan pada variabel penelitian. Perhitungan nilai rata-rata ini bertujuan untuk melihat kecenderungan jawaban responden secara umum terhadap fungsionalitas sistem. Nilai rata-rata dihitung dengan menggunakan rumus sebagai berikut:

$$Mean = \frac{\sum \text{Pertanyaan} \times \text{Bobot}}{\text{Total responden}}$$

Setelah nilai rata-rata dari setiap variabel diperoleh, langkah selanjutnya adalah melakukan perhitungan persentase skor. Perhitungan ini berfungsi untuk menentukan tingkat kelayakan atau efektivitas sistem berdasarkan kriteria persentase yang telah ditetapkan. Persentase skor dihitung dengan membandingkan total skor aktual yang diperoleh dengan skor ideal (skor maksimum), menggunakan persamaan sebagai berikut:

$$\text{Persentase} = \frac{\text{Nilai Mean}}{\text{Bobot Maksimum}} \times 100$$

Kuesioner berlanjut dengan menanyakan 10 pertanyaan sebagai *System Usability Scale (SUS)*. Dengan pertanyaan sebagai berikut:

Tabel 3.7 Pertanyaan System Usability Scale

No	Pernyataan
P1.	Saya berpikir akan menggunakan fitur ini lagi.
P2.	Saya merasa fitur ini rumit untuk digunakan.
P3.	Saya merasa fitur ini mudah untuk digunakan.

No	Pernyataan
P4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan fitur ini.
P5.	Saya merasa fitur <i>location tracker</i> ini berjalan seharusnya.
P6.	Saya merasa ada banyak hal yang tidak pas pada fitur <i>location tracker</i> .
P7.	Saya merasa orang lain akan memahami cara menggunakan fitur ini dengan cepat.
P8.	Saya merasa fitur ini membingungkan.
P9.	Saya merasa percaya diri dalam menggunakan fitur ini.
P10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Data yang diperoleh melalui instrumen kuesioner *System Usability Scale* (SUS) akan diolah untuk mendapatkan skor akhir yang merepresentasikan tingkat kegunaan aplikasi. Berdasarkan metodologi standar SUS, pengolahan data dilakukan dengan beberapa tahapan transformasi skor sebagai berikut:

1. Transformasi Skor Pernyataan Ganjil: Untuk setiap pernyataan dengan nomor urut ganjil (P1, P3, P5, P7, P9), skor yang diberikan oleh responden dikurangi dengan angka 1.
2. Transformasi Skor Pernyataan Genap: Untuk setiap pernyataan dengan nomor urut genap (P2, P4, P6, P8, P10), angka 5 dikurangi dengan skor jawaban yang diberikan oleh responden.
3. Kalkulasi Skor Individu: Hasil transformasi dari sepuluh pernyataan tersebut dijumlahkan, kemudian dikalikan dengan konstanta sebesar 2,5 untuk menghasilkan skor akhir responden dengan rentang nilai 0–100.

Setelah mendapat nilai dari perhitungan di atas, skor SUS dapat diperoleh dengan rumus perhitungan.

$$X = \frac{\sum ((X_{\text{Ganjil}} - 1) \times 2.5) + \sum ((5 - X_{\text{Genap}}) \times 2.5)}{\text{Total Responden}}$$

Dengan penjelasan:

X = rata-rata skor

X_{ganjil} = skor pertanyaan ganjil

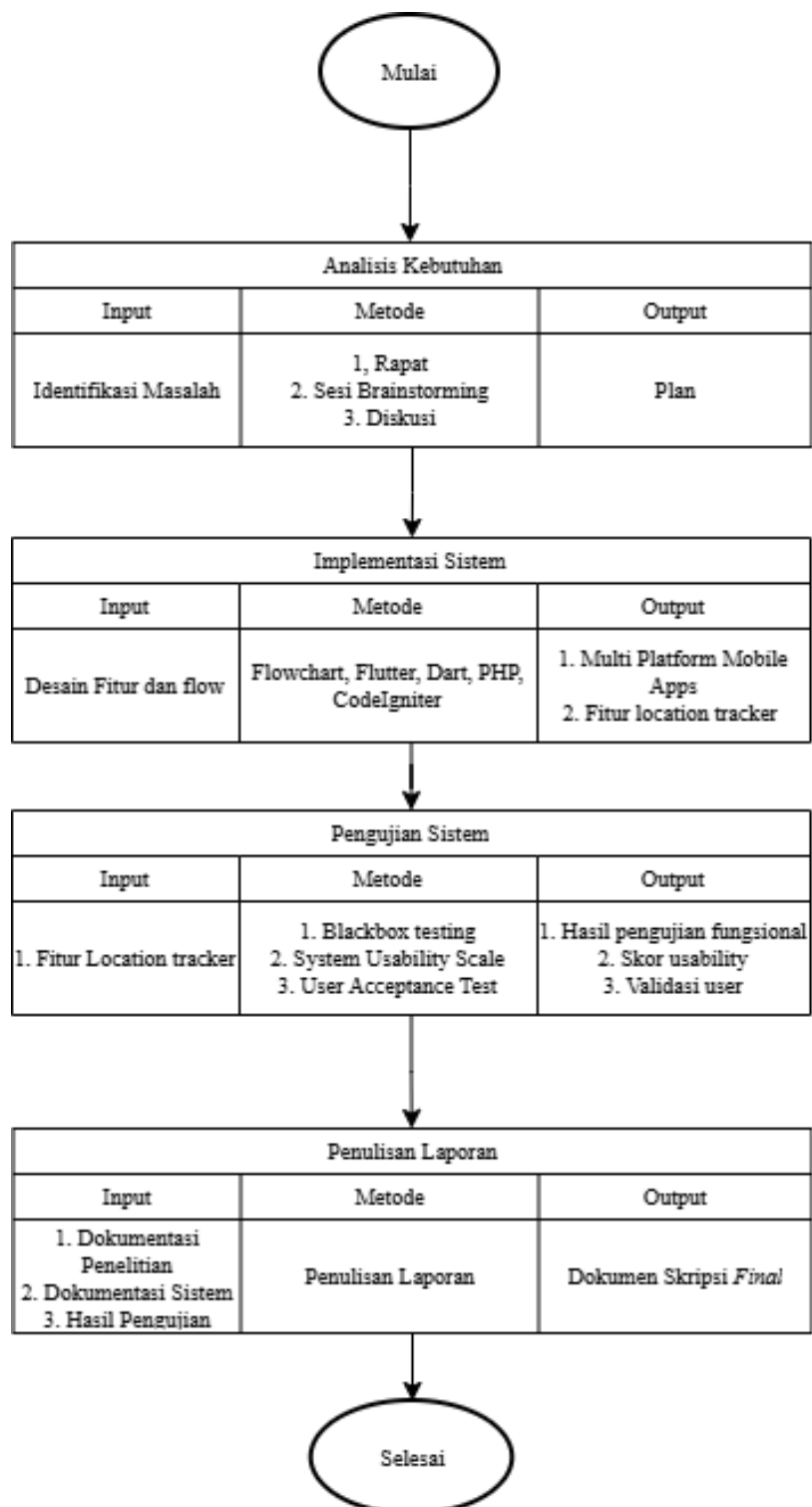
X_{genap} = skor pertanyaan genap

2.5 = Faktor pengali rumus SUS

Berdasarkan data ini fitur dikembangkan dengan metode *Waterfall*. Hasil pengujian aplikasi digunakan untuk evaluasi aplikasi. Selama proses pelaksanaan penelitian berlangsung, penelitian melalui beberapa tahapan yang dilalui untuk menyelesaikan permasalahan. Proses penelitian ini disusun dalam beberapa tahapan untuk menyelesaikan masalah operasional terkait keamanan lokasi jamaah di lapangan.

3.2 Tahapan Penelitian

Penelitian ini dilakukan dengan menggunakan metode pengembangan sistem *Waterfall*, yaitu model pengembangan yang dilakukan secara sistematis dan bertahap, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan berikutnya. Metode *Waterfall* dipilih dalam penelitian ini karena kebutuhan sistem telah terdefinisi secara jelas sejak awal pengembangan, sehingga tidak terdapat perubahan kebutuhan yang signifikan selama proses berlangsung. Kondisi ini sesuai dengan karakteristik utama model *Waterfall* yang bekerja paling optimal ketika ruang lingkup dan spesifikasi sistem sudah dipahami secara menyeluruh sebelum proses implementasi dimulai, sehingga setiap tahapan dapat direncanakan dan dieksekusi secara terstruktur tanpa risiko perubahan arah pengembangan di tengah jalan. Adapun pelaksanaan penelitian ini hanya mencakup tahapan hingga pengujian sistem (*testing*), tanpa melanjutkan ke tahapan *deployment* dan *maintenance*. Hal ini dikarenakan kedua tahapan tersebut berada di luar cakupan penelitian, mengingat aplikasi Ventour telah memiliki infrastruktur *deployment* dan proses pemeliharaan yang dikelola secara tersendiri oleh pihak pengembang. Dengan demikian, fokus penelitian diarahkan sepenuhnya pada perancangan, implementasi, dan evaluasi fitur *location tracker* hingga tahap pengujian, terlampir pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.2.1 Analisis Kebutuhan

Tahap awal pengembangan aplikasi, dimulai dari proses analisis kebutuhan yang dilakukan bersama dengan pihak Ventura Semesta Wisata. Dengan tujuan memperoleh informasi secara langsung terkait kebutuhan serta permasalahan. Analisis kebutuhan dilakukan melalui rapat, sesi *brainstorming* bersamaan dengan studi literatur lalu sesi diskusi untuk finalisasi konsep pengembangan fitur.

3.2.2 Implementasi Sistem

Pada tahap ini, rancangan sistem direalisasikan menjadi sebuah aplikasi *mobile* sesuai dari desain melalui pendekatan metode *Waterfall*. Implementasi teknisnya memanfaatkan bahasa pemrograman *dart* dengan *framework Flutter*. Selain itu, manajemen *state* aplikasi dikelola secara terstruktur menggunakan pola BLoC guna memastikan performa aplikasi yang optimal sesuai dengan kebutuhan yang telah didefinisikan sebelumnya.

3.2.3 Pengujian Sistem

Untuk memvalidasi fitur sudah sesuai dengan kebutuhan, maka pengujian sistem dilakukan, metode pengujian sistem yang digunakan pada penelitian ini adalah *black-box testing*, *user acceptance testing* dan *system usability scale*, sehingga pengujian hanya dilakukan secara fungsionalitas tanpa menguji kode program, hal ini didasari karena keterbatasan kemampuan penulis akan kemampuan *testing*.

3.2.4 Penulisan Laporan

Tahap akhir dari penelitian ini adalah penulisan laporan dari penelitian yang telah dilakukan mencakup seluruh tahapan selama proses pengembangan aplikasi, mulai dari identifikasi masalah, analisis kebutuhan, implementasi, hingga pengujian sistem. Dokumentasi dilakukan secara sistematis berdasarkan standar penulisan ilmiah.

3.3 Objek Penelitian

Objek penelitian ini berfokus pada permasalahan pelacakan jamaah yang terpisah dari rombongan pasca sesi bebas pada paket perjalanan yang diselenggarakan oleh PT Ventura Semesta Wisata, melalui pengembangan fitur *Location tracker*. Fitur *Location tracker* dikembangkan menggunakan *framework* Flutter dan diintegrasikan ke dalam aplikasi Ventour Mobile, yang merupakan hasil migrasi dari aplikasi berbasis *Progressive Web App* (PWA) sebelumnya. Objek utama dalam penelitian ini adalah aplikasi mobile dengan fitur *location tracker* yang datanya tersinkronisasi berdasarkan peran (role) pengguna, khususnya peran *Tour Leader*, untuk memantau posisi jamaah pada paket perjalanan yang sedang aktif.

Pengguna utama dari fitur ini adalah *Tour Leader*, yang bertindak sebagai pihak yang memantau lokasi jamaah secara *tersinkronisasi* selama perjalanan berlangsung, terutama pada momen-momen rawan seperti sesi bebas, di mana jamaah berpotensi terpisah dari rombongan. Sementara itu, jamaah berperan sebagai pengguna pasif, yaitu pihak yang posisinya dipantau melalui fitur *Location tracker* tanpa perlu melakukan interaksi aktif terhadap sistem. Dengan demikian, objek penelitian ini secara spesifik mencakup proses pengembangan, implementasi, dan pengujian fitur *Location tracker* pada aplikasi Ventour Mobile sebagai solusi untuk meningkatkan keamanan dan kemudahan pemantauan jamaah selama mengikuti paket perjalanan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan awal dalam pengembangan fitur aplikasi. Analisis dilakukan guna menyesuaikan kebutuhan pengguna dengan pengembangan fitur aplikasi. Untuk menunjang kesesuaian kebutuhan, metode yang digunakan mencakup wawancara, studi literatur dan observasi terhadap aplikasi dengan fitur sejenis seperti *Google Maps*, *Waze*, *Grab maps*. Melalui wawancara, penulis mendapatkan informasi dari pengguna serta pemangku kepentingan mengenai permasalahan dan ekspektasi terhadap fitur yang dikembangkan. Dengan informasi yang telah didapat, penulis melakukan studi literatur untuk mencari referensi serta dokumentasi relevan dalam pengembangan fitur serupa. Penulis membandingkan teori yang didapat dari referensi maupun dokumentasi dengan aplikasi yang memiliki fitur sejenis, untuk memahami bagaimana fitur ini dapat membantu dan informasi saja yang dibutuhkan, termasuk dalam penyediaan informasi lalu lintas, penentuan rute, serta respons yang didapatkan. Hasil dari analisis kebutuhan, sebagai berikut

4.1.1 Kebutuhan Fungsional

1. Fitur *Location tracker*

- a. *Tersinkronisasi update location.*
- b. Penyusunan rute dari posisi *Tour Leader* kepada jamaah.

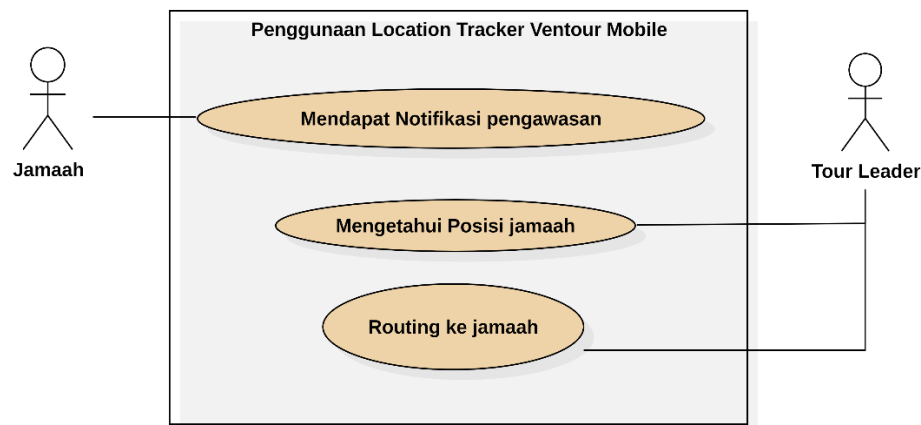
4.1.2 Kebutuhan Non-Fungsional

- a. Pengembangan aplikasi menggunakan bahasa pemrograman *dart* dengan *framework Flutter*.
- b. Penyesuaian data paket dengan data *Tour Leader* serta jamaah dengan *database* perusahaan.
- c. Menggunakan metode *Waterfall* sebagai alur proses pengembangan aplikasi.

4.2 Perancangan Sistem

Bagian ini akan menjelaskan gambaran mengenai sistem yang akan dikembangkan melalui beberapa pemodelan, seperti *Unified Modelling Language* (UML), *flowchart* dan *activity diagram*. Pemodelan akan menggambarkan alur kerja sistem, proses bisnis, serta interaksi antara pengguna dengan aplikasi secara jelas. Selain itu, dalam bagian ini membahas mengenai *Mockup* tampilan antarmuka aplikasi yang memberikan visualisasi mengenai tampilan fitur-fitur aplikasi. Dengan adanya rancangan, pengembangan sistem diharapkan sistematis, efisien dan sesuai dengan kebutuhan yang telah dianalisis.

4.2.1 Use case



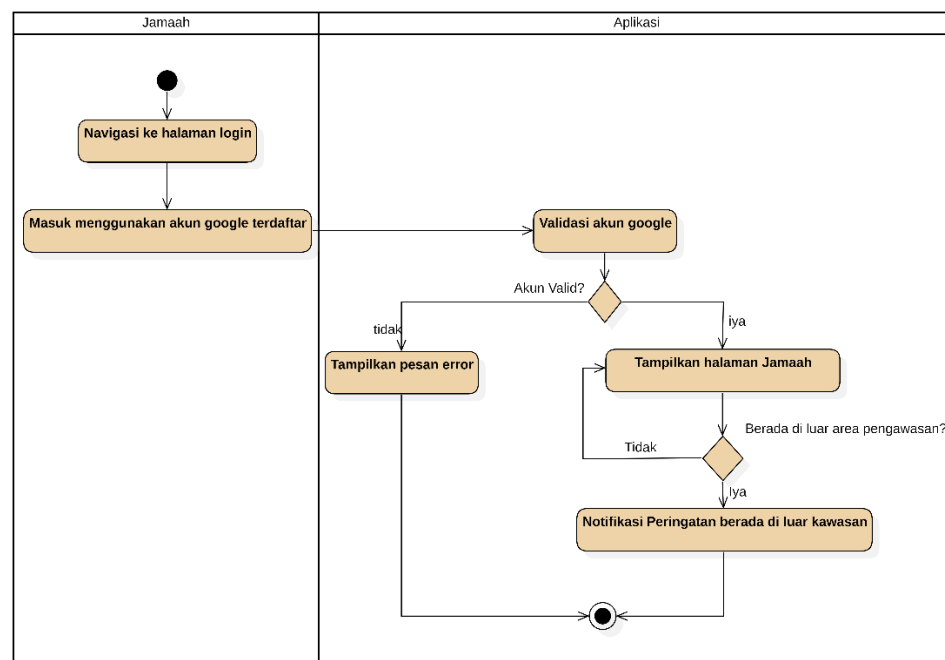
Gambar 4.1 Use case Location tracker

Gambar 4.1 menampilkan *Use case Diagram* yang memodelkan interaksi fungsional antara aktor pengguna dan sistem dalam batasan modul (*system boundary*) "Penggunaan Location tracker Ventour Mobile". Diagram ini menggambarkan ruang lingkup hak akses serta fungsionalitas khusus yang tersedia bagi dua aktor utama, yakni *Jamaah* dan *Tour Leader*. Secara rinci, fungsionalitas sistem (*use case*) bagi masing-masing aktor diuraikan sebagai berikut:

1. Aktor: *Jamaah* merupakan aktor pasif dalam konteks pemantauan aktif, namun bertindak sebagai entitas yang diawasi oleh sistem.

2. Aktor: *Tour Leader* merupakan aktor pengelola (*supervisor*) yang memiliki akses untuk menginisiasi dan memantau pergerakan Jamaah dalam suatu paket perjalanan. Dengan penjelasan *use case*, Mendapat Notifikasi Pengawasan Sistem memberikan layanan peringatan (*alert service*) kepada perangkat Jamaah. Mengetahui Posisi Jamaah Aktor *Tour Leader* memiliki hak akses untuk meminta dan menampilkan titik koordinat visual seluruh Jamaah yang berada di bawah tanggung jawabnya. *Use case: Routing* ke Jamaah Sebagai fungsi tambahan dari pemantauan lokasi, *Tour Leader* dapat memanfaatkan fitur navigasi untuk menemukan Jamaah yang terpisah.

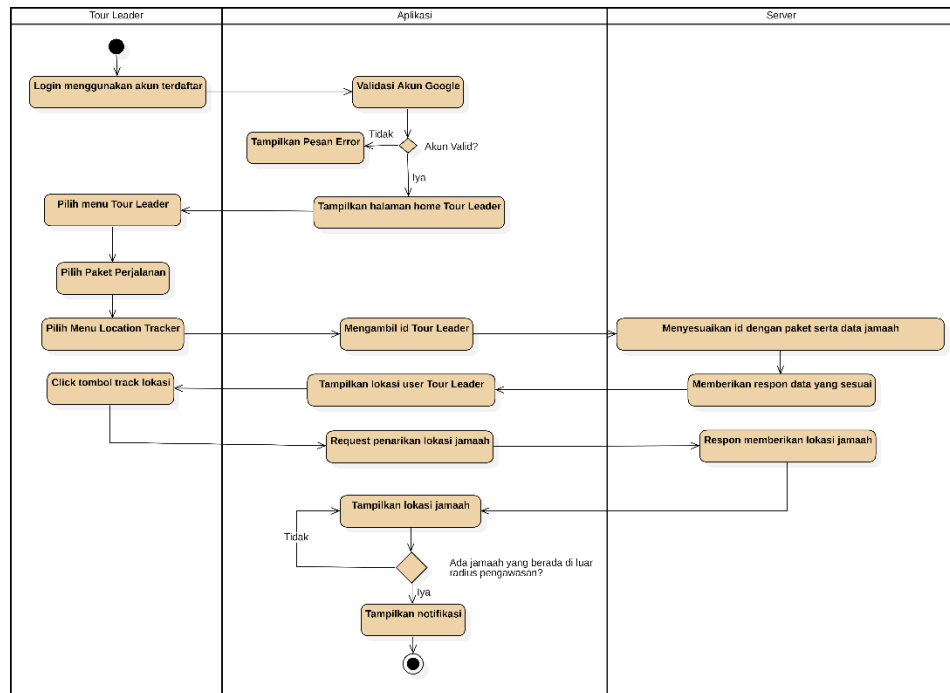
4.2.2 Activity Diagram mendapat notifikasi



Gambar 4.2 Activity Diagram Masuk dengan Akun Terdaftar

Gambar 4.2 mengilustrasikan alur aktivitas pengguna dengan peran Jamaah di dalam sistem. Proses diawali dengan autentikasi (*Login*) menggunakan akun Google yang terdaftar. Setelah akses diberikan dan fitur pemantauan berjalan di latar belakang, aplikasi dirancang untuk secara otomatis memunculkan notifikasi peringatan (*alert*) pada layar perangkat Jamaah apabila terdeteksi berada di luar batas area pengawasan.

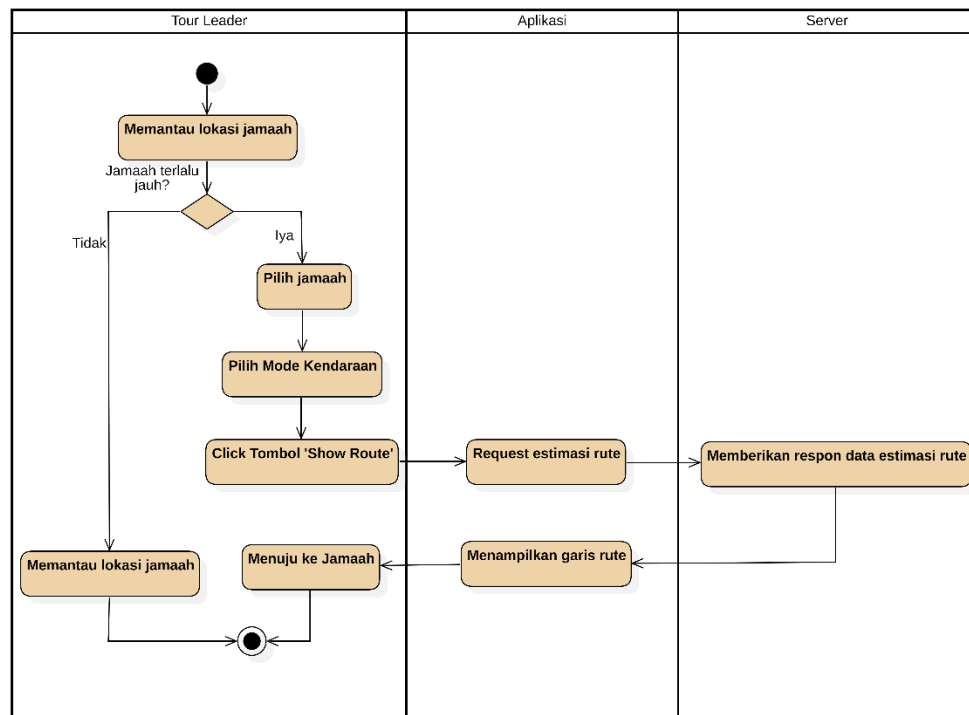
4.2.3 Activity Diagram Membaca Koordinat *Tour Leader* & jamaah



Gambar 4.3 Activity Diagram membaca koordinat *Tour Leader* & jamaah

Gambar 4.3 mengilustrasikan mekanisme interaksi sistem antara pengguna (*Tour Leader*) dengan *server* dalam proses pemanggilan data lokasi. Alur dimulai dengan langkah autentikasi, di mana *Tour Leader* wajib masuk (*Login*) ke dalam sistem dan melakukan navigasi menuju halaman utama fitur *Location tracker*. Setelah antarmuka peta termuat, pengguna menginisiasi pemantauan dengan menekan tombol pelacakan lokasi (*Track Lokasi*). Aksi ini akan mengirimkan permintaan (*request*) pembaruan data koordinat jamaah kepada *server*. Apabila validasi data di *server* berhasil, sistem akan merespons (*response*) dengan memproyeksikan titik-titik keberadaan jamaah dalam bentuk penanda (*Marker*). Pada saat yang bersamaan, aplikasi juga akan menampilkan notifikasi informatif mengenai jumlah jamaah yang berada di luar area pengawasan.

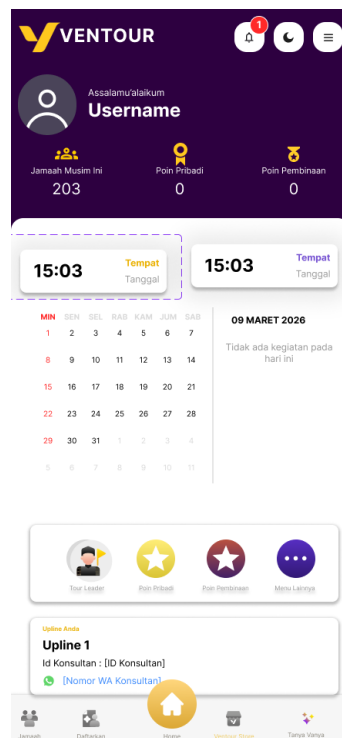
4.2.4 Activity Diagram melacak jamaah



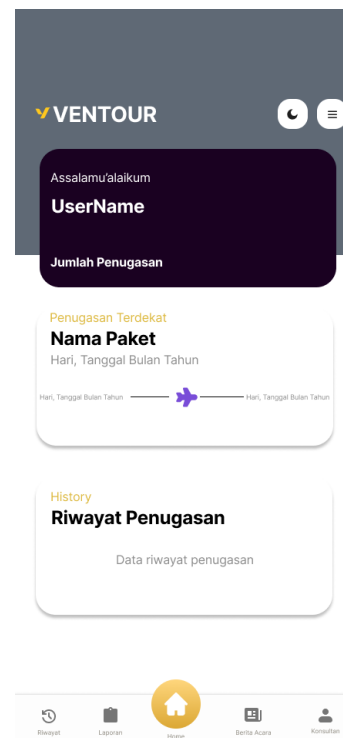
Gambar 4.4 Activity Diagram melacak Jamaah

Gambar 4.4 mendeskripsikan skenario operasional ketika aplikasi telah berhasil memetakan titik koordinat (*Marker*) jamaah. Jika terdeteksi jamaah dengan posisi berada jauh dari titik kumpul pasca sesi aktivitas bebas. Dalam kondisi tersebut, *Tour Leader* dapat memulai proses pencarian dengan memilih *Marker* jamaah yang dituju dan menentukan mode transportasi yang akan digunakan. Selanjutnya, *Tour Leader* menekan tombol 'Tunjukkan Rute' untuk mengirimkan permintaan (*request*). Kemudian *server* memproses permintaan dengan mengirim respons berupa visualisasi jalur navigasi (*polyline*) pada antarmuka peta aplikasi. Adanya informasi rute yang akurat ini memfasilitasi *Tour Leader* untuk melakukan penjemputan secara langsung dan efisien terhadap jamaah yang terpisah dari rombongan.

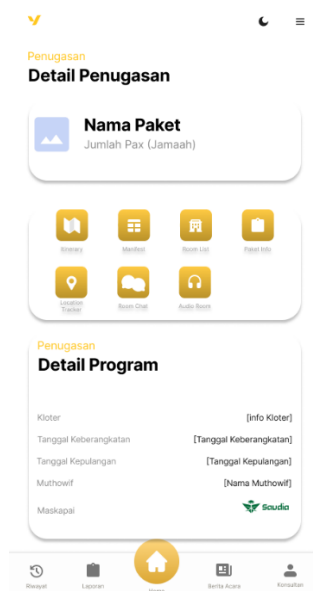
4.2.5 Rancangan Tampilan Fitur *Location tracker*



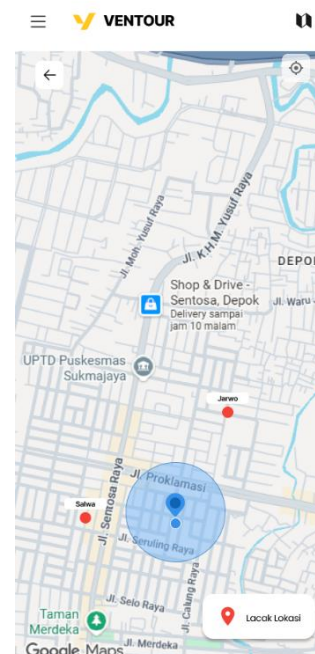
Gambar 4.5 Mockup Fitur Opsi *Location tracker* (1)



Gambar 4.6 Mockup Fitur Opsi *Location tracker* (2)



Gambar 4.7 Mockup Fitur Opsi *Location tracker* (3)



Gambar 4.8 Mockup *Location tracker*

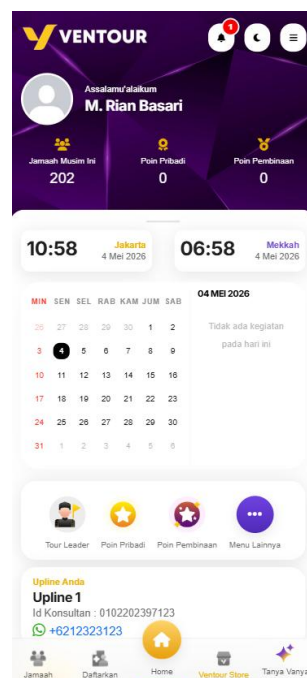
Pada gambar 4.5 hingga 4.7 merupakan rancangan desain untuk menampilkan opsi *location tracker*. Lalu pada gambar 4.8, merupakan

halaman yang menampilkan peta dengan koordinat lokasi dari *Tour Leader (user)* dan jamaah. Fitur *tracking* dimulai dengan menekan tombol lacak lokasi untuk menemukan lokasi jamaah lalu dilanjutkan dengan menekan mark jamaah ketika terlalu jauh untuk melakukan mode *routing* agar dapat menemukan jamaah yang terlalu jauh dari titik *Tour Leader*.

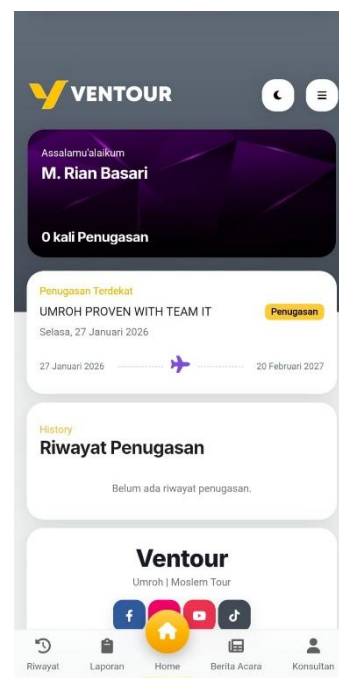
4.3 Implementasi Sistem

Setelah analisis kebutuhan, perancangan sistem dan desain tampilan aplikasi, beralih ke tahapan selanjutnya yaitu implementasi berdasarkan rancangan dan desain yang telah ditetapkan dengan *framework Flutter* dan menggunakan *Visual Studio Code* sebagai *Integrated Development Environment*, untuk membangun aplikasi android serta konfigurasi fungsi API. Berikut hasil implementasi fitur dilampirkan dalam gambar serta penjelasan sebagai berikut.

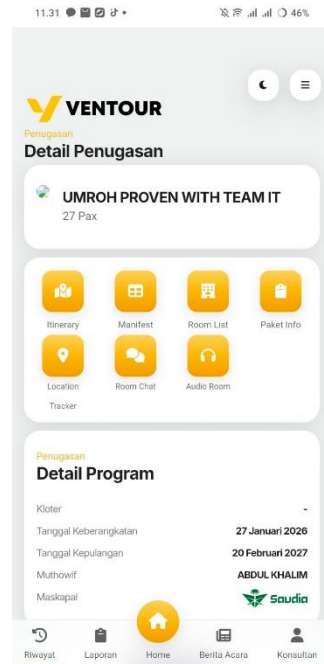
4.3.1 Implementasi Fitur *Location tracker*



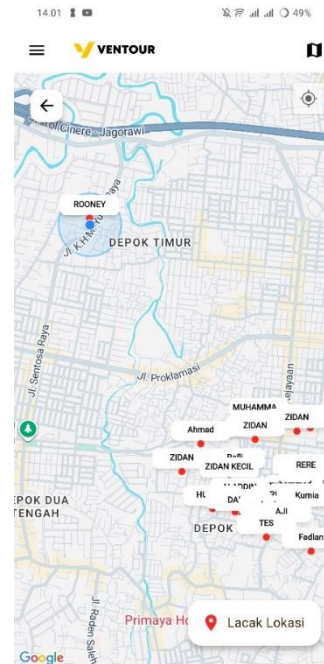
Gambar 4.9 Implementasi Fitur Opsi *Location tracker*



Gambar 4.10 Implementasi Fitur Opsi *Location tracker* (2)



Gambar 4.11 Implementasi Fitur Opsi
Location tracker (3)



Gambar 4.12 Implementasi Halaman
Location tracker

Pada Gambar 4.9 hingga Gambar 4.11 merupakan hasil implementasi dari rancangan antarmuka untuk menampilkan opsi fitur pelacak lokasi (*location tracker*). Selanjutnya, Gambar 4.12 menampilkan halaman peta digital yang memuat koordinat posisi geografis dari *Tour Leader* selaku pengguna (*user*) serta posisi para jamaah. Pada tahap implementasi ini, proses pelacakan (*tracking*) diinisiasi dengan menekan tombol "Lacak Lokasi" untuk memetakan seluruh posisi jamaah pada peta. Apabila terdapat jamaah yang terpisah terlalu jauh dari posisi *Tour Leader*, pengguna dapat menekan penanda (*Marker*) jamaah tersebut untuk mengaktifkan mode rute (*routing mode*). Fitur rute ini berfungsi untuk memandu dan mempermudah *Tour Leader* dalam menemukan jamaah yang berada di luar radius aman.

4.3.2 A* Algorithm Code

```

class PathResult {
    final List<GraphNode> path;
    final double totalDistanceMeters;
}

class PriorityQueue<T> {
    final List<T> _items = [];
    final int Function(T, T) _comparator;

    void add(T item) {
        _items.add(item);
        _items.sort(_comparator); // selalu terurut berdasarkan f(n) terkecil
    }

    T removeFirst() => _items.removeAt(0);
}

// g(n): biaya nyata dari start ke node n
final Map<String, double> gScore = {startId: 0};

// f(n) = g(n) + h(n): perkiraan total biaya
final Map<String, double> fScore = {
    startId: _heuristic(graph.nodes[startId]!, goalNode),
};

static double _heuristic(GraphNode a, GraphNode b) {
    return DistanceHelper.haversineDistance(
        LatLng(a.latitude, a.longitude),
        LatLng(b.latitude, b.longitude),
    );
}

....

while (openSet.isNotEmpty) {
    final current = openSet.removeFirst(); // ambil node dengan f(n) terkecil

    if (current == goalId) {
        return _reconstructPath(...); // jalur ditemukan
    }

    if (visited.contains(current)) continue;
    visited.add(current);

    for (final edge in neighbors) {
        final tentativeG = (gScore[current] ?? double.infinity) + edge.weight;

        if (tentativeG < (gScore[edge.toNodeId] ?? double.infinity)) {
            cameFrom[edge.toNodeId] = current;
            gScore[edge.toNodeId] = tentativeG;
            fScore[edge.toNodeId] = tentativeG + _heuristic(neighborNode, goalNode);
            openSet.add(edge.toNodeId);
        }
    }
}

static PathResult _reconstructPath(...) {
    final path = <GraphNode>[graph.nodes[current]!];
    String node = current;
    while (cameFrom.containsKey(node)) {
        node = cameFrom[node]!;
        path.add(graph.nodes[node]!);
    }
    return PathResult(
        path: path.reversed.toList(), // balik dari goal ke start → jadi start ke goal
        totalDistanceMeters: double.parse(totalDistance),
    );
}

```

Gambar 4. 13 A* Algorithm Snippet Code

Penggalan kode berikut merupakan alur utama dari implementasi algoritma A* dalam menentukan rute perjalanan sebagai alternatif

terhadap *Google Maps* API. Proses dimulai dengan inisialisasi nilai $g(n)$ dan $f(n)$ pada node awal, di mana $g(start) = 0$ karena belum ada biaya tempuh yang dikeluarkan, serta $f(start) = h(start)$ yang merepresentasikan estimasi awal jarak menuju tujuan berdasarkan fungsi heuristik Haversine.

Selanjutnya, algoritma memasuki fase *loop* eksplorasi dengan mengambil node yang memiliki nilai $f(n)$ terkecil dari *openSet* menggunakan struktur data *Priority Queue*. Ketika node yang sedang dievaluasi merupakan node tujuan (*goal*), algoritma menghentikan eksplorasi dan melakukan rekonstruksi jalur melalui fungsi `_reconstructPath`, yang menelusuri balik *map* `cameFrom` dari node tujuan menuju node asal untuk menghasilkan urutan node yang membentuk rute optimal. Apabila node yang dievaluasi belum merupakan tujuan, node tersebut ditandai sebagai *visited* untuk mencegah evaluasi ulang yang dapat menyebabkan perulangan tak terbatas. Kemudian, untuk setiap node tetangga yang terhubung melalui *edge*, algoritma menghitung nilai *tentative* $g(n)$ yaitu total biaya nyata dari node awal hingga node tetangga tersebut melalui jalur yang sedang dievaluasi. Apabila nilai *tentative* $g(n)$ yang diperoleh lebih kecil dari nilai $g(n)$ node tetangga yang tersimpan sebelumnya, maka jalur menuju node tetangga diperbarui dan node tetangga ditambahkan ke dalam *openSet* untuk dieksplorasi lebih lanjut.

Proses eksplorasi ini berlanjut hingga node tujuan ditemukan atau *openSet* habis. Apabila *openSet* kosong sebelum node tujuan ditemukan, algoritma mengembalikan nilai null yang menandakan bahwa tidak ada rute yang dapat ditemukan antara titik asal dan titik tujuan dalam graf jaringan jalan yang tersedia. Secara keseluruhan, implementasi algoritma A^* dalam penelitian ini memanfaatkan kombinasi antara biaya nyata $g(n)$ dan estimasi heuristik $h(n)$ berbasis Haversine untuk memprioritaskan eksplorasi node yang secara geometris lebih mendekati tujuan, sehingga proses pencarian rute dapat dilakukan secara efisien di sisi perangkat *mobile* (*client-side*) tanpa memerlukan komputasi di sisi *server*.

4.3.3 GMaps Fetching

```
final url = Uri.parse('${AppConfig.baseUrl}/api_flutter/fetchLocation');

final Map<String, dynamic> requestBody = {
  "origin": {
    "latitude": origin.latitude,
    "longitude": origin.longitude,
  },
  "destination": {
    "latitude": destination.latitude,
    "longitude": destination.longitude,
  },
  "travelMode": travelMode,
};

final headers = {'Content-Type': 'application/json'};

final res = await http.post(
  url,
  headers: headers,
  body: jsonEncode(requestBody),
);

if (res.statusCode != 200) {
  throw Exception('Failed to fetch route info: ${res.body}');
}

final data = jsonDecode(res.body);

final distance = data['distance'] ?? 'Unknown';
final distanceMeters = data['distanceMeters'] ?? 'Unknown';
final duration = data['duration'] ?? 'Unknown';
final polyline = data['polyline'] ?? '';
final speedReadingIntervals = data['speedReadingIntervals'] ?? '[]';

return {
  'distance': distance,
  'distanceMeters': distanceMeters,
  'duration': duration,
  'polyline': polyline,
  'speedReadingIntervals': speedReadingIntervals,
};
```

Gambar 4. 14 Snippet Code GMaps Routes Fetching

4.4 Pengujian Sistem

Pengujian dilakukan untuk menemukan permasalahan atau galat pada aplikasi serta memastikan fitur yang dirancang dan dikembangkan telah berjalan sesuai dengan rancangan serta harapan. Bagian ini membahas pengujian yang dilakukan pada penelitian mulai dari deskripsi, prosedur, data hasil hingga evaluasi pengujian.

4.4.1 Deskripsi Pengujian

Setelah proses implementasi sistem berdasarkan desain yang telah dirancang, maka tahapan selanjutnya dari penelitian ini adalah pengujian sistem. Metode yang digunakan pada adalah *User Acceptance Testing*, dengan pengkategorian sebagai berikut, *Blackbox-Testing (Error (haversine, Response Time), User Acceptance Test*, serta *System Usability Scale*. Responden dalam penelitian ini difokuskan pada para *Tour Leader* sebagai pengguna utama (*end-user*) fitur *Location tracker* pada aplikasi Ventour Mobile. Pemilihan subjek penelitian didasarkan pada kriteria pengguna yang telah mengoperasikan sistem secara langsung guna memastikan data yang diperoleh bersifat objektif dan relevan. Sebanyak 7 responden terlibat dari 30 *Tour Leader* per bulan berdasarkan estimasi *Tour Leader* yang dimiliki oleh Ventour.

Responden yang dilibatkan merupakan *Tour Leader* aktif Ventour yang telah menangani beberapa kali perjalanan ibadah umrah dan haji, serta memiliki pengalaman langsung dalam melakukan pengawasan dan koordinasi jamaah di lapangan. Pengujian ini bersifat sebagai **evaluasi awal (*preliminary evaluation*)** untuk mengukur tingkat penerimaan dan kemudahan penggunaan (*usability*) fitur yang dikembangkan dari sudut pandang pengguna utama, sebelum dilakukan implementasi dan evaluasi lanjutan secara menyeluruh pada seluruh *Tour Leader* Ventour. Hasil responden digunakan sebagai data untuk mengevaluasi fungsionalitas dan tingkat kegunaan sistem dari sudut pandang operasional lapangan.

4.4.2 Prosedur Pengujian

Prosedur pengujian merupakan tahapan untuk pengembang melakukan pengujian terhadap aplikasi, sistem ataupun fitur yang telah dibuat. Proses pengujian dibagi menjadi, *Blackbox-Testing (Error Haversine, Response Time)*, *User Acceptance Testing*, *System Usability Scale (SUS)*. Diperoleh hasil perhitungan sebagai berikut.

4.4.3 Data Hasil Pengujian

1) Hasil pengujian *Blackbox-Testing*

Hasil pengujian dengan metode *blackbox testing* yang dilakukan secara bertahap untuk memastikan keluaran telah sesuai dengan kebutuhan terlampir pada tabel berikut.

Tabel 4.1 Hasil *Blackbox-Testing*

No	Skenario	Hasil yang diharapkan	Hasil Aktual	Keterangan
1.	Mengakses data lokasi jamaah	Aplikasi menampilkan Lokasi jamaah dengan <i>Marker</i>	Lokasi jamaah terlihat dengan <i>Marker</i>	Valid
2.	Menggunakan mode rute	Aplikasi menampilkan rute antara titik <i>Tour Leader</i> dengan jamaah	Muncul rute antara titik <i>Tour Leader</i> dengan jamaah	Valid
3.	Update lokasi jamaah saat mode rute	Aplikasi <i>mengupdate</i> lokasi jamaah	Lokasi jamaah <i>terupdate</i>	Valid
4,	Memunculkan notifikasi	Aplikasi menampilkan	Tampil notifikasi	Valid

No	Skenario	Hasil yang diharapkan	Hasil Aktual	Keterangan
	ketika ada jamaah di luar radius pengawasan	notifikasi dengan angka jamaah yang berada di luar radius pengawasan	dengan angka jamaah yang berada di luar radius pengawasan	
5.	Memunculkan notifikasi sebagai jamaah ketika berada terlalu jauh dari <i>Tour Leader</i>	Aplikasi menampilkan notifikasi berada di luar radius pengawasan	Tampil notifikasi berada di luar radius pengawasan	Valid
6.	Track Jamaah saat lokasi mati	Meminta akses lokasi	Meminta akses lokasi	Meminta akses lokasi
7.	Track Jamaah saat lokasi menyala	<i>Marker</i> jamaah langsung tampil	<i>Marker</i> jamaah langsung tampil	Valid
8.	Track jamaah saat tidak ada internet	Muncul halaman pemberitahuan bahwa “Anda sedang <i>offline</i> ”	Muncul halaman pemberitahuan bahwa “Anda sedang <i>offline</i> ”	Valid

2) Hasil pengujian Respon

Hasil pengujian waktu respons dengan metode *blackbox testing* yang dilakukan secara bertahap untuk memastikan keluaran telah sesuai dengan kebutuhan terlampir pada tabel berikut.

Tabel 4. 2 Pengujian waktu respon

No.	Skenario	Waktu respon (Wifi)	Waktu respon (Jaringan Seluler)
1.	Load Halaman	00.64 (Detik)	00.57 (Detik)
2.	Lokasi Jamaah Muncul	01.01 (Detik)	00.77 (Detik)
3.	Update Posisi	00.70 (Detik)	00.75 (Detik)
4.	Mode Rute (Gmaps)	01.11 (Detik)	00.82 (Detik)
5.	Mode Rute (A*)	05.36 (Detik)	05.14 (Detik)

3) Hasil perbandingan estimasi waktu dan durasi

Hasil perbandingan estimasi dan waktu durasi algoritma A* dan *Google Maps API*. Dengan metode *blackbox testing* yang dilakukan secara bertahap untuk memastikan keluaran telah sesuai dengan kebutuhan terlampir pada tabel berikut.

Tabel 4. 3 Perbandingan mode rute *Google Maps* dan A*

No	Awal	Tujuan	Mode	Jarak A* (Km)	Jarak Google (Km)	Selisih Jarak	Durasi A*	Durasi Google	Selisih Durasi
1	Perumahan Studio Alam Indah	Marker (Kamar i)	Berjalan	4.6	5.2	0.6 Km	54 Menit	16 Menit	38 Menit
2	Perumahan Studio Alam Indah	Marker (Kamar i)	Kendaraan Roda 2	4.6	5.2	0.6 Km	7 Menit	16 Menit	9 Menit

3	Perumahan Studio Alam Indah	Marker (Kamar i)	Kendaraan Roda 4	4.6	5.2	0.6 Km	9 Menit	20 Menit	11 Menit
4	Perumahan Studio Alam Indah	Marker (Eka Setyawati)	Berjalan	15.8	16.9	1.1 Km	189 Menit	37 Menit	152 Menit
5	Perumahan Studio Alam Indah	Marker (Eka Setyawati)	Kendaraan Roda 2	14.7	16.9	2.2 Km	25 Menit	35 Menit	10 Menit
6	Perumahan Studio Alam Indah	Marker (Eka Setyawati)	Kendaraan Roda 4	14.7	16.9	2.2 Km	29 Menit	43 Menit	14 Menit

4) Hasil pengujian *User Acceptance Testing*

Data yang telah diperoleh dari hasil penyebaran kuesioner kepada 7 responden *Tour Leader* diklasifikasikan berdasarkan kelompok kode pertanyaan yang mewakili setiap variabel *User Acceptance Testing* (UAT). Proses analisis dilakukan dengan menghitung nilai rata-rata (*mean*) serta akumulasi skor ke dalam bentuk persentase (%) untuk mengukur tingkat kelayakan sistem.

Tabel 4. 4 Pengujian *User Acceptance Testing*

R	A1	A2	A3	B1	B2	B3	C1	C2	C3	D1	D2	D3
1	5	5	3	5	5	5	4	5	5	5	4	5
2	4	4	5	4	5	3	5	4	3	4	4	4
3	4	5	5	4	5	4	5	5	5	5	5	5
4	4	4	4	3	4	3	4	4	4	4	5	4
5	5	4	5	5	5	5	5	4	4	4	4	3
6	5	5	5	5	5	4	5	5	5	5	5	5
7	5	4	4	5	5	4	4	5	4	4	4	5

Perhitungan bobot nilai pada data mentah ini dilakukan dengan mengalikan frekuensi jumlah jawaban responden pada setiap butir pertanyaan (A1 hingga D3) dengan nilai bobot penilaian *skala Likert* yang telah ditetapkan pada tabel 3.3. Rincian hasil perhitungan bobot, nilai rata-rata, beserta grafik interpretasi persentase untuk masing-masing variabel pengujian tersebut akan dipaparkan secara mendetail pada sub-bab selanjutnya.

5) Hasil Pengujian *System Usability Scale* (SUS)

Berikut ini merupakan hasil dari pengujian *System Usability Scale* pada fitur *Location tracker*.

Tabel 4.5 Hasil *System Usability Scale* (SUS)

R	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Skor SUS
1	4	4	4	3	3	3	4	4	4	3	90
2	3	3	3	2	3	3	3	4	3	3	75
3	3	3	4	2	3	2	4	3	4	3	77.5
4	2	3	3	3	3	3	3	3	3	1	67.5
5	1	3	1	3	3	3	1	3	1	3	55
6	3	4	4	3	4	3	3	3	3	1	77.5
7	4	2	4	2	4	3	4	3	4	2	80

4.4.4 Evaluasi Pengujian

1) Pengujian *Blackbox-Testing*

Setelah seluruh rangkaian pengujian menggunakan metode *black-box* selesai dilaksanakan, tahapan berikutnya adalah melakukan kalkulasi untuk menentukan tingkat keberhasilan pengujian tersebut. Untuk mengukur persentase dan tingkat keberhasilan dari skenario pengujian fungsional, digunakan persamaan sebagai berikut:

$$\text{Persentase Keberhasilan} = \frac{\sum \text{Skenario berhasil}}{\sum \text{Skenario Uji}} \times 100\%$$

Berdasarkan seluruh rangkaian pengujian fungsional yang telah dilaksanakan, diperoleh data bahwa dari 8 skenario uji yang dieksekusi, seluruhnya berhasil terpenuhi secara akurat tanpa adanya kegagalan sistem. Pengujian dihitung sebagai berikut:

$$\text{Persentase keberhasilan} = \frac{8}{8} \times 100 \% = 100 \%$$

Berdasarkan perhitungan di atas nilai persentase keberhasilan adalah 100%, Dengan demikian, dapat disimpulkan bahwa seluruh komponen dan menu yang diuji melalui metode *black-box testing* pada fitur *Location tracker* telah beroperasi sepenuhnya tanpa galat (*error*).

2) Pengujian *response time*

Berdasarkan hasil pengujian pada Tabel 4.2, pengujian dilakukan pada dua kondisi jaringan yang berbeda — *Wi-Fi* dengan kecepatan *download* 11,3 Mbps dan *upload* 11,3 Mbps, serta jaringan seluler dengan kecepatan *download* 98,0 Mbps dan *upload* 21,8 Mbps. Secara keseluruhan, jaringan seluler menunjukkan rata-rata waktu respon yang lebih baik dibandingkan *Wi-Fi*, yaitu 1,610 detik berbanding 1,764 detik, meskipun kecepatan *download* jaringan seluler yang digunakan dalam pengujian ini 8,7 kali lebih tinggi dari *Wi-Fi*.

Pada skenario dasar aplikasi — pemuatan halaman, kemunculan lokasi jamaah, dan pembaruan posisi — rata-rata waktu respon jaringan seluler (0,697 detik) lebih cepat dibandingkan *Wi-Fi* (0,783 detik). Perbedaan paling signifikan terlihat pada skenario kemunculan lokasi jamaah, di mana jaringan seluler (0,77 detik) lebih cepat 23,8% dibandingkan *Wi-Fi* (1,01 detik). Hal ini konsisten dengan perbedaan kapasitas *download* kedua jaringan, mengingat skenario ini melibatkan pengambilan data jamaah dari server yang cukup bergantung pada kecepatan unduh.

Terdapat satu pengecualian pada skenario pembaruan posisi (*update posisi*), di mana *Wi-Fi* (0,70 detik) justru sedikit lebih cepat dibandingkan jaringan seluler (0,75 detik) dengan selisih 0,05 detik. Perbedaan yang sangat kecil ini kemungkinan disebabkan oleh fluktuasi latensi jaringan seluler yang bersifat dinamis pada saat pengujian berlangsung serta kondisi server di saat melakukan *request*—kecepatan *download/upload* tinggi tidak selalu berkorelasi langsung dengan latensi rendah, terutama pada jaringan seluler yang latensinya dipengaruhi oleh kondisi sinyal, kepadatan pengguna di *tower* BTS terdekat, serta waktu pengujian. Selisih 0,05 detik ini berada dalam rentang variasi yang wajar dan tidak cukup signifikan untuk disimpulkan sebagai perbedaan performa yang berarti.

Pada skenario pencarian rute, jaringan seluler mengungguli *Wi-Fi* baik untuk *Google Maps Routes API* (0,82 vs 1,11 detik, selisih 26,1%) maupun algoritma *A** (5,14 vs 5,36 detik, selisih 4,1%). Menariknya, meskipun jaringan seluler memiliki kecepatan *download* yang jauh lebih tinggi, selisih performa antara kedua jaringan untuk mode *A** jauh lebih kecil (0,22 detik) dibandingkan mode *Google Maps* (0,29 detik). Hal ini dapat dijelaskan dari karakteristik proses *A** itu sendiri — sebagian besar waktu komputasinya dihabiskan untuk pemrosesan di sisi perangkat (*on-device*) berupa pembangunan graf dan eksekusi algoritma pencarian, bukan untuk transfer data jaringan. Dengan demikian, perbedaan kecepatan jaringan memberikan dampak yang lebih terbatas terhadap total waktu respon *A**, karena *bottleneck* utamanya bukan pada konektivitas melainkan pada kapasitas komputasi perangkat.

Perbedaan ini dapat dijelaskan dari sisi arsitektur kedua pendekatan. Metode *Google Maps Routes API* memanfaatkan infrastruktur server Google yang telah dioptimalkan secara global dengan data jalan yang sudah terindeks dan teknik pencarian rute tingkat lanjut (seperti *contraction hierarchies*), sehingga proses komputasi pencarian rute terjadi hampir instan di sisi server. Sebaliknya, metode *A** pada penelitian ini melibatkan beberapa tahapan tambahan yang dilakukan secara tersinkronisasi di sisi klien (*on-device*), yaitu: (1) pengambilan data

jaringan jalan dari Overpass API sesuai *bounding box* area pencarian, (2) pembentukan graf jaringan jalan dari data mentah *OpenStreetMap*, dan (3) eksekusi algoritma A* itu sendiri untuk menemukan jalur optimal. Ketiga tahapan ini belum melalui proses optimasi tingkat lanjut sebagaimana yang diterapkan pada layanan komersial seperti *Google Maps*, sehingga total waktu yang dibutuhkan menjadi lebih lama.

Meskipun demikian, waktu respon sebesar 5,36 & 05,14 detik untuk metode A* masih berada dalam **batas yang dapat diterima** (*acceptable*) untuk skenario penggunaan pada aplikasi ini, mengingat metode tersebut hanya diterapkan sebagai solusi alternatif untuk rute dengan jarak tempuh di bawah 20 km, dan dengan fokus sebagai analisis perbandingan serta berfungsi sebagai upaya mitigasi terhadap keterbatasan akses maupun biaya penggunaan *Google Maps Routes* API secara berkelanjutan.

3) Pengujian perbandingan estimasi waktu dan durasi dari A* dan *Google Maps*

Berdasarkan keenam skenario pengujian yang ada pada tabel 4.3, selisih jarak antara metode A* dan *Google Maps* berada pada rentang 0,6 km hingga 2,2 km, dengan rata-rata *error* sebesar **11,19%**. Selisih ini relatif konsisten antar mode perjalanan karena nilai jarak yang dihasilkan algoritma A* tidak dipengaruhi oleh kecepatan asumsi, melainkan murni oleh graf jaringan jalan yang berhasil diekstraksi dari data *OpenStreetMap*. Perbedaan jarak ini wajar terjadi karena perbedaan sumber data peta — *Google Maps* menggunakan data jalan miliknya sendiri yang kemungkinan lebih lengkap dan ter-update, sementara algoritma A* pada penelitian ini bergantung pada kelengkapan data kontribusi komunitas *OpenStreetMap* di area pengujian, yang dapat menghasilkan rute dengan struktur jalan yang sedikit berbeda.

Pada perbandingan durasi, ditemukan perbedaan yang signifikan antara mode berjalan dengan mode motor dan mobil. Untuk mode berjalan, *error* durasi mencapai rata-rata **324,16%**, jauh lebih tinggi dibandingkan mode motor dan mobil yang rata-rata *error*nya sebesar **43,09%**.

Perbedaan besar pada mode berjalan ini disebabkan oleh asumsi kecepatan rata-rata pejalan kaki yang ditetapkan pada sistem untuk perhitungan dengan algoritma A*, yaitu **5 km/jam**. Sementara itu, durasi yang diestimasi *Google Maps* untuk skenario serupa (16 menit untuk 5,2 km, 37 menit untuk 16,9 km) mengindikasikan bahwa *Google Maps* mengasumsikan kecepatan efektif pejalan kaki yang jauh lebih tinggi, yaitu sekitar **19-27 km/jam** — hal ini disebabkan oleh kesalahan ketika mendefinisikan method *request* ke *directions* API yang telah tertulis pada batasan masalah.

Pada mode motor dan mobil, *error* durasi yang lebih rendah (rata-rata 43,09%) menunjukkan bahwa asumsi kecepatan 30-35 km/jam pada sistem cukup mendekati estimasi *Google Maps*, meskipun belum mempertimbangkan kondisi lalu lintas tersinkronisasi sebagaimana yang dilakukan *Google Maps* melalui *traffic-aware routing*.

Hal ini disebabkan karena adanya perbedaan pembobotan serta parameter dari a* *algorithm* dan Google Maps API. Pembobotan pada Google Maps API, khususnya Routes API yang digunakan pada mode *DRIVE* dan *TWO_WHEELER* dalam penelitian ini menggunakan sistem pembobotan yang sangat kompleks dan bersifat *traffic-aware*, yang mencakup sejumlah parameter sebagai berikut: (1) jarak aktual setiap ruas jalan berdasarkan basis data peta Google, (2) kondisi lalu lintas *real-time* yang diperoleh dari sensor jalan, laporan pengguna, dan pola historis, (3) kecepatan rata-rata historis per ruas jalan yang bervariasi berdasarkan jam dan hari dalam seminggu, (4) klasifikasi jenis jalan dengan bobot berbeda untuk jalan tol, arteri, kolektor, dan jalan lokal, (5) rambu dan peraturan lalu lintas seperti larangan belok dan batas kecepatan, (6) estimasi waktu tambahan untuk setiap manuver belokan, serta (7) perbedaan karakteristik per mode transportasi. Keseluruhan parameter ini diproses oleh infrastruktur komputasi Google di sisi *server* menggunakan teknik optimasi tingkat lanjut, sehingga menghasilkan estimasi rute dan durasi yang sangat mendekati kondisi nyata di lapangan.

Sedangkan implementasi algoritma A^* pada penelitian ini menggunakan parameter pembobotan yang jauh lebih sederhana, yaitu: (1) bobot *edge* berupa jarak Haversine antar node OSM yang berdekatan sebagai nilai $g(n)$, (2) estimasi sisa jarak ke tujuan menggunakan Haversine sebagai fungsi heuristik $h(n)$, (3) asumsi kecepatan tetap per mode perjalanan (5 km/jam untuk berjalan kaki, 35 km/jam untuk kendaraan roda dua, dan 30 km/jam untuk kendaraan roda empat), (4) filter jenis jalan berdasarkan *tag highway* OpenStreetMap (*include/exclude*), serta (5) informasi arah satu arah berdasarkan *tag oneway* dari data OSM. Kesederhanaan parameter ini merupakan konsekuensi langsung dari pendekatan komputasi *client-side* yang dipilih dalam penelitian ini, di mana seluruh proses perhitungan dijalankan di perangkat *mobile* dengan sumber daya terbatas tanpa dukungan infrastruktur *server*.

Ketidaksetaraan parameter pembobotan antara kedua sistem ini memiliki implikasi langsung terhadap hasil pengujian. Selisih jarak yang diperoleh rata-rata sebesar 11,19% sebagian besar disebabkan oleh perbedaan sumber data peta (Google Maps menggunakan basis data jalannya sendiri yang lebih lengkap, sementara A^* bergantung pada kelengkapan kontribusi komunitas *OpenStreetMap* di area pengujian), bukan semata-mata karena perbedaan algoritma pencarian jalurnya.

Dengan demikian, perbandingan antara algoritma A^* dan *Google Maps* API dalam penelitian ini **tidak bersifat setara secara teknis**, melainkan merupakan perbandingan antara pendekatan *routing* berbasis komputasi lokal yang ringan dan sederhana dengan layanan *routing* komersial yang telah dioptimalkan secara global. Meskipun demikian, perbandingan ini tetap relevan dan bernilai dalam konteks penelitian, karena bertujuan untuk mengevaluasi sejauh mana pendekatan *client-side* yang tidak bergantung pada layanan berbayar dapat memberikan hasil estimasi rute yang memadai sebagai alternatif dalam skenario penggunaan aplikasi manajemen perjalanan jamaah.

4) Pengujian *User Acceptance Testing*

Setelah mendapat data responden dari *User Acceptance Testing*. Proses pengolahan data diawali dengan penghitungan skor kumulatif, tiap frekuensi jawaban pada pertanyaan dikalikan dengan nilai bobot sesuai skala likert yang ditetapkan tabel 3.3. Detail hasil kalkulasi tiap variabel dijelaskan pada bagian pembahasan berikut.

Variabel 1: Evaluasi Fungsional Sistem

Evaluasi Fungsional Sistem

Tabel 4.6 Bobot Penilaian Evaluasi Fungsional Sistem

P	SS x 5	S x 4	N x 3	TS x 2	STS x 1	Jumlah
A1	4 x 5 = 20	3 x 4 = 12	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	32
A2	3 x 5 = 15	4 x 3 = 12	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	31
A3	4 x 5 = 20	2 x 4 = 8	1 x 3 = 3	0 x 3 = 0	0 x 3 = 0	31

Variabel 2: Evaluasi Kinerja Sistem

Evaluasi Kinerja Sistem

Tabel 4.7 Bobot Penilaian Evaluasi Kinerja Sistem

P	SS x 5	S x 4	N x 3	TS x 2	STS x 1	Jumlah
B1	4 x 5 = 20	2 x 4 = 8	1 x 3 = 3	0 x 3 = 0	0 x 3 = 0	31
B2	6 x 5 = 30	1 x 4 = 4	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	34
B3	2 x 5 = 10	3 x 4 = 12	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	22

Variabel 3: Evaluasi Pengalaman & Tampilan Antarmuka Sistem

Evaluasi Pengalaman & Tampilan Antarmuka Sistem

Tabel 4.8 Bobot Evaluasi Pengalaman & Tampilan Antarmuka Sistem

P	SS x 5	S x 4	N x 3	TS x 2	STS x 1	Jumlah
C1	4 x 5 = 20	3 x 4 = 12	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	32
C2	6 x 5 = 30	1 x 4 = 4	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	34
C3	3 x 5 = 15	3 x 4 = 12	2 x 3 = 6	0 x 3 = 0	0 x 3 = 0	33

Variabel 4: Evaluasi Efisiensi & Produktivitas

Evaluasi Efisiensi & Produktivitas

Tabel 4.9 Bobot Evaluasi Efisiensi & Produktivitas

P	SS x 5	S x 4	N x 3	TS x 2	STS x 1	Jumlah
D1	3 x 5 = 15	4 x 4 = 16	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	31
D2	3 x 5 = 15	4 x 4 = 16	0 x 3 = 0	0 x 3 = 0	0 x 3 = 0	31
D3	4 x 5 = 20	2 x 4 = 8	1 x 3 = 3	0 x 3 = 0	0 x 3 = 0	31

Setelah dilakukan pembobotan skor UAT, tahap selanjutnya adalah melakukan kalkulasi persentase guna menentukan tingkat kelayakan sistem secara objektif. Penentuan kriteria interpretasi skor dilakukan berdasarkan klasifikasi berikut:

Tabel 4.10 Keterangan dari persentase skor

Persentase	Keterangan
0-20%	Sangat Kurang Baik
21-40%	Kurang Baik
41-60%	Cukup Baik

61-80%	Baik
81-100%	Sangat Baik

Kemudian, hasil akhir jumlah pada tabel 4.4 hingga 4.7 di atas, dijadikan bahan acuan untuk mencari nilai rata-rata dan persentase untuk mengukur kelayakan sistem dengan rumus persentase yang telah dijelaskan pada bab Perhitungan dilakukan sebagai berikut:

Evaluasi Fungsional Sistem

Tabel 4.11 Hasil perhitungan Evaluasi Fungsional Sistem

P	Nilai <i>Mean</i>	Persentase	Nilai Rata
A1	$32/7 = 4.5$	91.43%	89.52%
A2	$31/7 = 4.4$	88.57%	
A3	$31/7 = 4.4$	88.57%	

Dari hasil evaluasi di atas, dapat dilihat bahwa nilai rata-rata evaluasi fungsionalitas sistem yaitu 89%, semua komponen dan fitur sistem berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna.

Evaluasi Kinerja Sistem

Tabel 4.12 Hasil perhitungan Evaluasi Kinerja Sistem

P	Nilai <i>Mean</i>	Persentase	Nilai Rata
B1	$31/7 = 4.4$	88.57%	88.57%
B2	$34/7 = 4.8$	97.14%	
B3	$28/7 = 4$	80.00%	

Berdasarkan hasil evaluasi kinerja sistem yang disajikan pada tabel di atas, diperoleh nilai rata-rata persentase sebesar 88,57%. Hal ini menunjukkan bahwa secara teknis, sistem mampu menjalankan instruksi fitur *Location tracker* dengan stabil dan responsif, di mana seluruh komponen mesin aplikasi bekerja selaras dengan spesifikasi performa yang diharapkan oleh pengguna.

Evaluasi Pengalaman & Tampilan Antarmuka Sistem

Tabel 4.13 Hasil perhitungan Evaluasi Pengalaman & Tampilan Antarmuka Sistem

P	Nilai <i>Mean</i>	Persentase	Nilai Rata
---	-------------------	------------	------------

C1	$32/7 = 4.5$	91.43%	94.29%
C2	$34/7 = 4.8$	97.14%	
C3	$33/7 = 4.7$	94.29%	

Merujuk pada tabel evaluasi pengalaman dan tampilan antarmuka sistem, didapatkan skor rata-rata tertinggi sebesar 94,29%. Angka ini merepresentasikan tingkat kepuasan yang sangat tinggi dari sisi *User Interface* (UI) dan *User Experience* (UX). Hal ini membuktikan bahwa desain antarmuka aplikasi Ventour Mobile sangat intuitif, komunikatif, dan memudahkan *Tour Leader* dalam mengoperasikan peta navigasi tanpa kendala visual.

Evaluasi Efisiensi & Produktivitas

Tabel 4.14 Hasil perhitungan Evaluasi Efisiensi & Produktivitas

P	Nilai <i>Mean</i>	Persentase	Nilai Rata
D1	$31/7 = 4.4$	88.57%	88.57%
D2	$31/7 = 4.4$	88.57%	
D3	$31/7 = 4.4$	88.57%	

Pada kategori evaluasi efisiensi dan produktivitas, hasil perhitungan data menunjukkan nilai rata-rata sebesar 88,57%. Pencapaian ini mengindikasikan bahwa implementasi fitur pelacakan jamaah memberikan kontribusi nyata dalam meningkatkan produktivitas kerja di lapangan. Sistem dinilai mampu memangkas waktu pengawasan manual sehingga proses pemantauan lokasi jamaah menjadi jauh lebih efisien bagi operasional perusahaan.

3) Evaluasi Pengujian *System Usability Scale* (SUS)

Berdasarkan data kuesioner, hasil responden dihitung menggunakan rumus SUS. Hasil perhitungan dapat dilihat pada tabel 4.13.

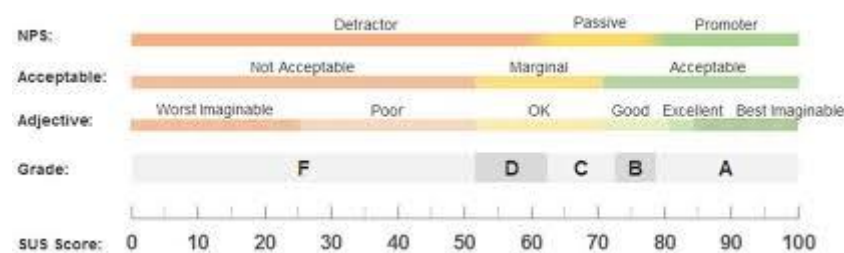
Tabel 4.15 Hasil Perhitungan SUS

R	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Skor SUS

1	4	4	4	3	3	3	4	4	4	3	90
2	3	3	3	2	3	3	3	4	3	3	75
3	3	3	4	2	3	2	4	3	4	3	77.5
4	2	3	3	3	3	3	3	3	3	1	67.5
5	1	3	1	3	3	3	1	3	1	3	55
6	3	4	4	3	4	3	3	3	3	1	77.5
7	4	2	4	2	4	3	4	3	4	2	80
Rata-rata skor											74.64

Berdasarkan pengujian *System Usability Scale* (SUS), diperoleh jumlah skor pertanyaan ganjil adalah 275 dan jumlah skor untuk pertanyaan genap adalah 247.5. Dengan data ini rata-rata dihitung dengan rumus

$$X = \frac{\sum ((X_{\text{Ganjil}} - 1) \times 2.5) + \sum ((5 - X_{\text{Genap}}) \times 2.5)}{\text{Total Responden}} = \frac{522.5}{7} = 74.64$$



Gambar 4.15 Skala *System Usability Scale*

Sumber: (RG Guntur Alam and Puji Rahayu Kurniasih, 2024)

Berdasarkan skor rata-rata SUS yang diperoleh sebesar 74,64 serta acuan skala interpretasi skor SUS, nilai tersebut menempatkan fitur *Location tracker* pada kategori Grade B dengan *Adjective Rating* 'Good'. Dari sisi tingkat penerimaan (*Acceptability*), sistem dinyatakan 'Acceptable' atau dapat diterima untuk digunakan dalam aktivitas operasional. Meskipun pada kategori NPS sistem masih berada pada level '*Passive*', skor akhir yang melampaui standar rata-rata SUS (skor 68) membuktikan bahwa antarmuka dan fungsionalitas aplikasi telah memenuhi aspek *usability* yang cukup bagi para *Tour Leader*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, pengembangan, dan pengujian yang telah dilaksanakan, dapat ditarik beberapa kesimpulan. Implementasi fitur *Location tracker* pada aplikasi *Ventour Mobile* telah berhasil direalisasikan secara penuh dari tahap perancangan hingga pengujian. Meskipun pengujian *blackbox* menunjukkan fungsionalitas fitur telah mencapai tingkat keberhasilan 100%, stabilitas koneksi internet tetap menjadi tantangan utama dalam tahap penggunaan fitur. Sedangkan, instrumen *System Usability Scale* (SUS) menghasilkan skor rata-rata sebesar 74,64, menempatkan aplikasi dalam kategori '*Good*'. Skor ini menunjukkan bahwa sistem memiliki tingkat kegunaan yang baik dan dapat diterima untuk kebutuhan operasional, meskipun pada indikator *Net Promoter Score* (NPS), pengguna masih berada pada fase mempertimbangkan untuk merekomendasikan fitur secara proaktif. Berdasarkan hasil uji fungsional, *usability* serta simulasi terbatas fitur ini berpotensi untuk memudahkan *Tour Leader* dalam pencarian atau pengumpulan kembali jamaah pasca sesi bebas serta integrasi sistem notifikasi dan pelacakan *tersinkronisasi*, meminimalkan risiko jamaah hilang atau tersesat, sekaligus menciptakan pengalaman perjalanan yang lebih interaktif serta optimal bagi seluruh peserta.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan selama proses pengembangan, berikut adalah beberapa poin saran untuk meningkatkan kualitas sistem di masa mendatang:

- 1) Implementasi Fitur *Panic Button*: Pengembangan selanjutnya diharapkan dapat meningkatkan interaktivitas dua arah dengan menambahkan fitur tombol darurat (*panic button*) di sisi jamaah. Fitur ini memungkinkan jamaah untuk mengirimkan sinyal urgensi kepada *Tour Leader* secara instan

meskipun masih berada di dalam area pengawasan, sehingga respons bantuan dapat dilakukan dengan ketika jamaah merasa tersesat.

- 2) Optimalisasi Radius Pengawasan Dinamis *Adaptive Geofencing* disarankan untuk mengganti parameter radius pengawasan yang bersifat *hardcode* (200 meter) menjadi mekanisme radius yang dinamis dan adaptif. Sistem dapat dikembangkan untuk secara otomatis menyesuaikan batas area pengawasan berdasarkan lokasi spesifik di mana penugasan *Tour Leader*, misalnya mengikuti luas fisik kompleks Masjidil Aqsa atau area bertugas lainnya.
- 3) Evaluasi dan Optimasi *State Management*, seiring dengan pertumbuhan kompleksitas dan volume fitur aplikasi, perlu dilakukan evaluasi terhadap penggunaan *state management*. Mempertimbangkan alternatif yang lebih *lightweight* atau melakukan optimasi pada pola BLoC saat ini sangat disarankan untuk mereduksi *boilerplate code* dan beban *overhead* aplikasi.
- 4) Penerapan *Automated Unit Testing* guna menjamin keandalan sistem jangka panjang. Pengujian *unit testing* pada tiap komponen akan memastikan integritas logika bisnis terjaga, mempermudah proses deteksi kesalahan sejak dini (*early bug detection*), serta meningkatkan standar kualitas dan stabilitas aplikasi secara keseluruhan.

DAFTAR PUSTAKA

- Abdillah, Mgs.M.F., Sardi, I.L., Hadikusuma, A., 2023. Analisis Performa GetX dan BLoC State Management Library Pada Flutter untuk Perangkat Lunak Berbasis Android. LOGIC: Jurnal Penelitian Informatika 1, 73.
<https://doi.org/10.25124/logic.v1i1.6479>
- Aliyah Aliyah, Nahrun Hartono, Asrul Azhari Muin, 2024. Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. Switch : Jurnal Sains dan Teknologi Informasi 3, 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Apiag, C., Cadiz, E.S., Lincopinis, D.R., 2023. A Review on PHP Programming Language.
- Arjiansa, R.R., Sutabri, T., 2023. Pengukuran Tingkat Kemudahan Pegawai Terhadap Penggunaan Layanan Aplikasi SIMRS Menggunakan Metode System Usability Scale (SUS) Pada Rumah Sakit Umum Daerah Sekayu. Indonesian Journal of Multidisciplinary on Social and Technology 1, 115–120. <https://doi.org/10.31004/ijmst.v1i2.132>
- Garg, I., 239403, M.N., Reddy, N., Matriculate, W., 2025. Study on JSON, its Uses and Applications in Engineering Organizations.
- Kim, M., Xin, Q., Sinha, S., Orso, A., 2022. Automated test generation for REST APIs: no time to rest yet, in: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2022. Association for Computing Machinery, New York, NY, USA, pp. 289–301. <https://doi.org/10.1145/3533767.3534401>
- Kinari, S.A., Funabiki, N., Aung, S.T., Kyaw, H.H.S., 2025. A Guided Self-Study Platform of Integrating Documentation, Code, Visual Output, and Exercise for Flutter Cross-Platform Mobile Programming. Computers 14. <https://doi.org/10.3390/computers14100417>
- Liu, L., Wang, B., Xu, H., 2022. Research on Path-Planning Algorithm Integrating Optimization A-Star Algorithm and Artificial Potential Field Method. Electronics (Switzerland) 11. <https://doi.org/10.3390/electronics11223660>
- Qolbi, W.F., Taufiq, M., Muhammad, T., 2025. Rancang Bangun Sistem Informasi Arsip Surat Berbasis Web dengan Menggunakan Framework CodeIgniter. Computer Journal 3, 108–120. <https://doi.org/10.58477/cj.v3i2.326>
- Randi Ramadhani, D., Sisephaputra, B., 2026. Journal of Emerging Information System and Business Intelligence DESIGN OF LOST PILGRIM

- MONITORING APPLICATION WITH ANDROID LBS. *Journal of Emerging Information System and Business Intelligence* 7, 134–143.
- RG Guntur Alam, Puji Rahayu Kurniasih, 2024. PENGGUNAAN METODE SYSTEM USABILITY SCALE (SUS) PADA APLIKASI SIMAMURAT.
- Safinatun Najah, N., Frendiana, V., Budi Utami, dan, Studi Broadband Multimedia, P., Teknik Elektro, J., Negeri Jakarta, P., GA Siwabessy, J.D., 2025. Rancang Bangun Aplikasi Mobile dengan GPS untuk Presensi WFO dan WFS, Prosiding Seminar Nasional Teknik Elektro.
- Setiawan, A., Tri Prastowo, A., Darwis, D., Pagar Alam No, J.Z., Ratu, L., Lampung, B., 2022. SISTEM MONITORING KEBERADAAN POSISI MOBIL BERBASIS GPS DAN PENYADAP SUARA MENGGUNAKAN SMARTPHONE. *Jurnal Teknik dan Sistem Komputer (JTIKOM)* 3, 2022.
- Shankar Srinivas Lingolu, M., Kumar Dobbala, M., 2022. A Comprehensive Review of Progressive Web Apps: Bridging the Gap Between Web and Native Experiences. *International Journal of Science and Research (IJSR)* 11, 1326–1334. <https://doi.org/10.21275/sr24517172948>
- Shofi Baharuddin, A., Jatmika, S., 2024. Aplikasi Monitoring Lokasi Anak Menggunakan Location Based Service Dengan History Tracking Lokasi Berbasis Android. <https://doi.org/https://doi.org/10.32815/jiskomsia.v2i2.74>
- Topete, A., He, C., Protzko, J., Schooler, J., Hegarty, M., 2024. How is GPS used? Understanding navigation system use and its relation to spatial ability. *Cogn. Res. Princ. Implic.* 9. <https://doi.org/10.1186/s41235-024-00545-x>
- Ulum, M.F., Pasuruan, I.Y., 2025. Pengembangan Aplikasi Mobile Tracking dengan Integrasi GPS Realtime, *Jurnal Sistem Informasi*. Pasuruan.
- Vindua, R., Handayani, D., Ekrinifda, A., 2024. Implementation of Dart Programming Language in Mobile-Based DRs Snack Sales Application Design. *Architecture and High Performance Computing* 6. <https://doi.org/10.47709/cnape.v6i3.4203>
- Wang, H., Zhang, Y., Fu, X., 2025. Innovative Applications of Black Box Testing: A New Strategy for Teaching Assessment in Higher Education. *IEEE Access* 13, 90514–90526. <https://doi.org/10.1109/ACCESS.2025.3568825>
- Zulistiyan, M., Adrian, M., Wibowo, Y., 2024. Performance Analysis of BLoC and GetX State Management Library on Flutter. *Journal of Information System Research (JOSH)* 5, 583–591. <https://doi.org/10.47065/josh.v5i2.4698>

DAFTAR RIWAYAT HIDUP

Teuku Muhammad Erlangga Pratama



Teuku Muhammad Erlangga Pratama lahir di Jakarta pada tanggal 04 Oktober 2004. Penulis merupakan anak pertama dari dua bersaudara. Menyelesaikan pendidikan dasar di SDN SuKmajaya 5 Depok pada tahun 2016, kemudian melanjutkan pendidikan menengah pertama di SMPN 6 Depok dan lulus pada tahun 2019. Pendidikan menengah atas diselesaikan di SMAN 3 Depok pada tahun 2022. Saat ini, penulis sedang menempuh pendidikan Diploma IV pada Program Studi Teknik Informatika, Jurusan Teknik Informatika dan Komputer di Politeknik Negeri Jakarta.

LAMPIRAN

Lampiran 1 Bukti Wawancara dengan Supervisor tim IT Ventour

Lokasi: Kantor Ventour Pusat

Waktu: 4 Februari 2026



Lampiran 2 Tabel Hasil Wawancara dengan Pihak Ventour

Lampiran Tabel 1 Hasil Sesi Wawancara

No	Pertanyaan	Jawaban
1.	Sebagai awalan, ingin bertanya perangkat apa yang digunakan untuk memastikan lokasi jamaah saat tour berlangsung?	Awalnya tidak ada alat untuk melacak jamaah ketika hilang, kemudian menggunakan <i>Mi tag</i> (perangkat Bluetooth yang terhubung jika berada di sekitar hp).

Lampiran 2 Tabel Hasil Wawancara dengan Pihak Ventour (lanjutan)

No	Pertanyaan	Jawaban
2.	Apa ada kendala ketika penggunaan alat <i>Location tracker</i> sebelumnya hingga perlu menggantikan dengan fitur aplikasi?	Kendala yang terjadi adalah perbaruan lokasi tidak <i>tersinkronisasi</i> , delay bahkan hingga setengah jam. Oleh karena itu, diperlukan fitur <i>location tracking</i> yang <i>tersinkronisasi</i> (memungkinkan double proteksi).
3.	Apa ekspektasi dari aplikasi yang memiliki <i>Location tracker</i> dari PT Ventura Semesta Wisata?	Ekspektasi penggunaan <i>Location tracker</i> adalah ketika jamaah hilang langsung dilacak dan dijemput oleh <i>Tour Leader</i> . Serta memperlancar rangkaian perjalanan sesuai dengan paket yang sedang berlangsung.
4.	Untuk keperluan data pendukung, apa memungkinkan untuk saya meminta estimasi atau bahkan riwayat data pengeluaran operasional untuk pengadaan atau penyewaan alat <i>Location tracker</i> pak?	Estimasi biaya operasional, kami membeli <i>Mi tag</i> dengan kisaran harga Rp.200.000. dipinjamkan ke jamaah masing-masing mendapat 1 dengan tiap perjalanan bersama kisaran 50 jamaah.
5.	Bolehkah saya meminta estimasi atau perhitungan penggunaan <i>Google Maps</i> api dalam penggunaan <i>Location tracker</i> ?	Berdasarkan dokumentasi dan informasi yang saya dapat, pembayaran dimulai jika sudah melewati seribu <i>request</i> , akan terhitung premium. Dengan estimasi saat ini, 1 juta rupiah per bulan dengan catatan penggunaan hanya dilakukan ketika kasuss jamaah hilang setelah sesi bebas (<i>free time</i>).

Lampiran 2 Tabel Hasil Wawancara dengan Pihak Ventour (lanjutan)

No	Pertanyaan	Jawaban
6.	Rencananya, kapan fitur ini digunakan apakah hanya dalam jadwal kegiatan tertentu? atau bersifat <i>tersinkronisasi</i> 24 jam?	Fitur digunakan ketika jamaah hilang, dengan tujuan mengurangi problematika setelah sesi <i>free time</i> .
7.	Siapa saja target pengguna yang berinteraksi langsung dengan fitur ini?	Target pengguna dari fitur ini adalah Jamaah untuk mengirimkan koordinat lokasi mereka. Sedangkan, <i>Tour Leader</i> dan muthowif memegang akses pemantauan fitur lokasi ini.
8.	Mengingat penggunaan aplikasi di luar negeri, apa ada kriteria area atau kondisi geografis tertentu saat fitur harus bekerja secara optimal?	Hal ini menjadi tantangan karena harus terkoneksi dengan kuota internet. Namun, Ventour sudah menawarkan solusi sebelum perjalanan, yaitu menyediakan layanan isi paket atau kartu perdana dengan paket data <i>roaming</i> .
9.	Untuk pengembangan fitur ini, apa ada arahan <i>tech stack</i> khusus yang akan digunakan?	Teknisnya pakai <i>Flutter</i> karena multi platform dan menyesuaikan <i>user</i> jadi harus multiplatform

Lampiran 2 Tabel Hasil Wawancara dengan Pihak Ventour (lanjutan)

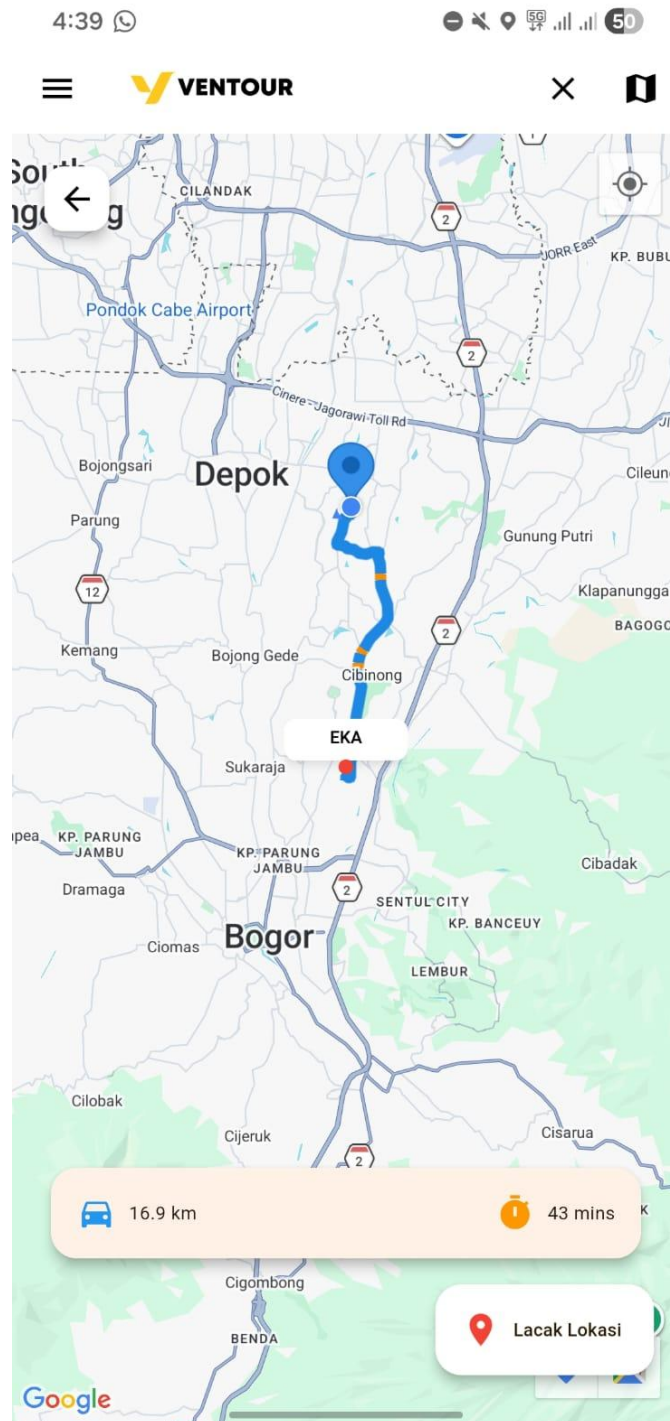
No	Pertanyaan	Jawaban
10.	Kenapa harus pake <i>Location tracker</i> buatan venom di saat banyak aplikasi <i>Location tracker</i> (informasi apa yang jadi alesan harus kustomisasi sendiri)	Masalah yang ada adalah <i>Tour Leader</i> harus mendaftarkan jamaah ke WA Group pada tiap perjalanan sehingga kurang efisien secara waktu. Dengan fitur <i>Location tracker</i> kustomisasi, aplikasi ini telah memiliki <i>database</i> tentang jamaah, serta informasi <i>Tour Leader</i> yang akan digunakan sebagai validasi satu <i>Tour Leader</i> ini memiliki jamaah siapa saja dengan menyesuaikan paket yang tengah berlangsung.

Lampiran 3

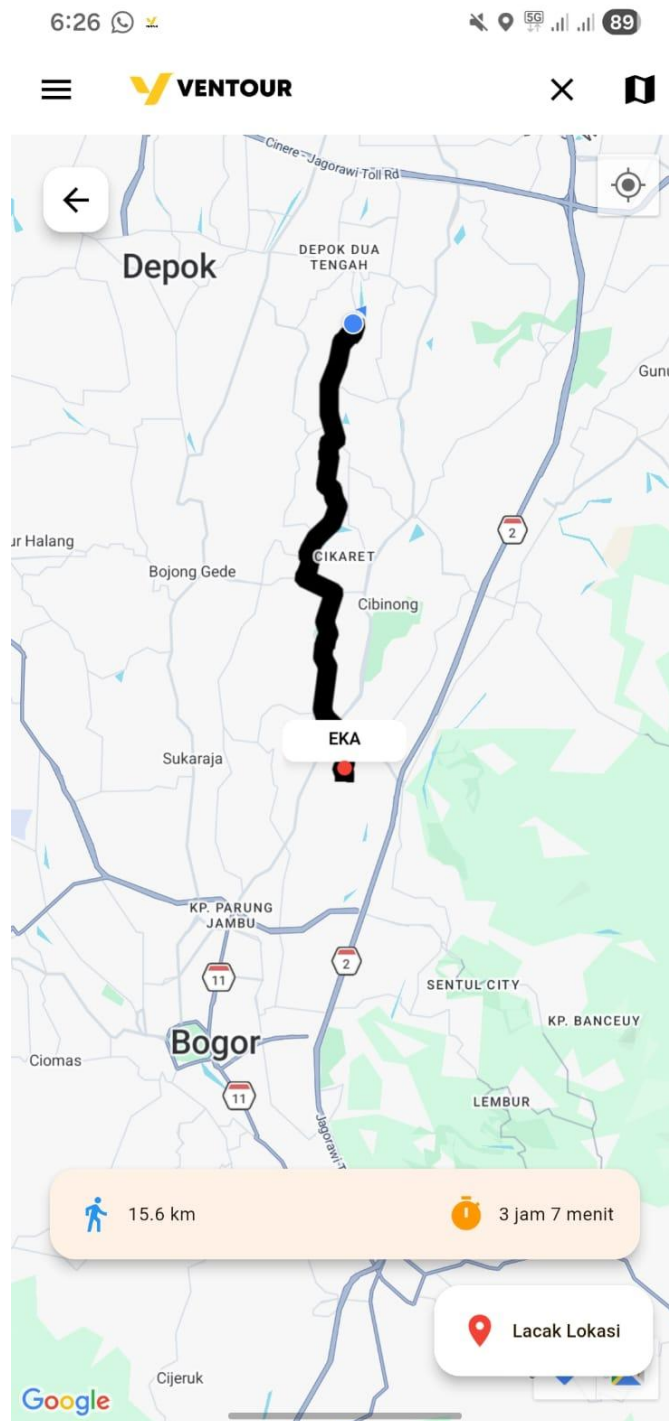
Bukti pengujian perbandingan metode A* dengan *Google Maps*

Lokasi: Perumahan Studio Alam Indah

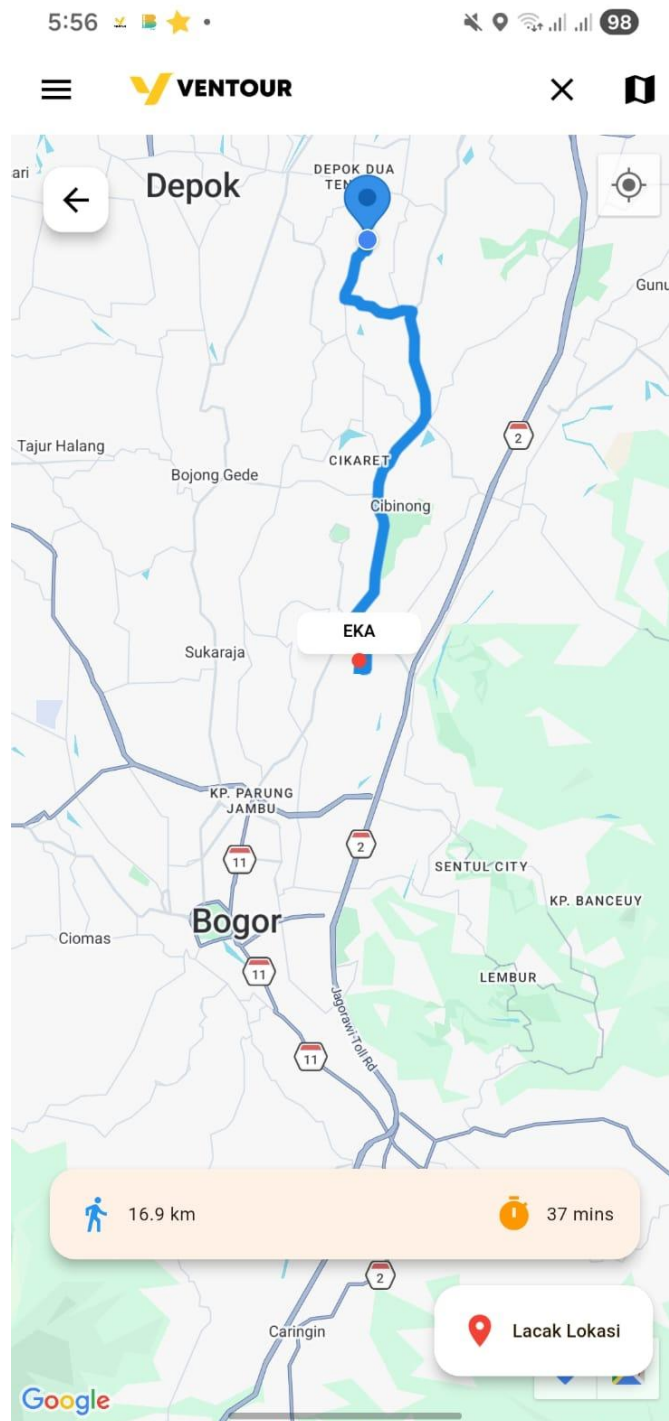
Waktu: Sabtu, 28 Juni 2026



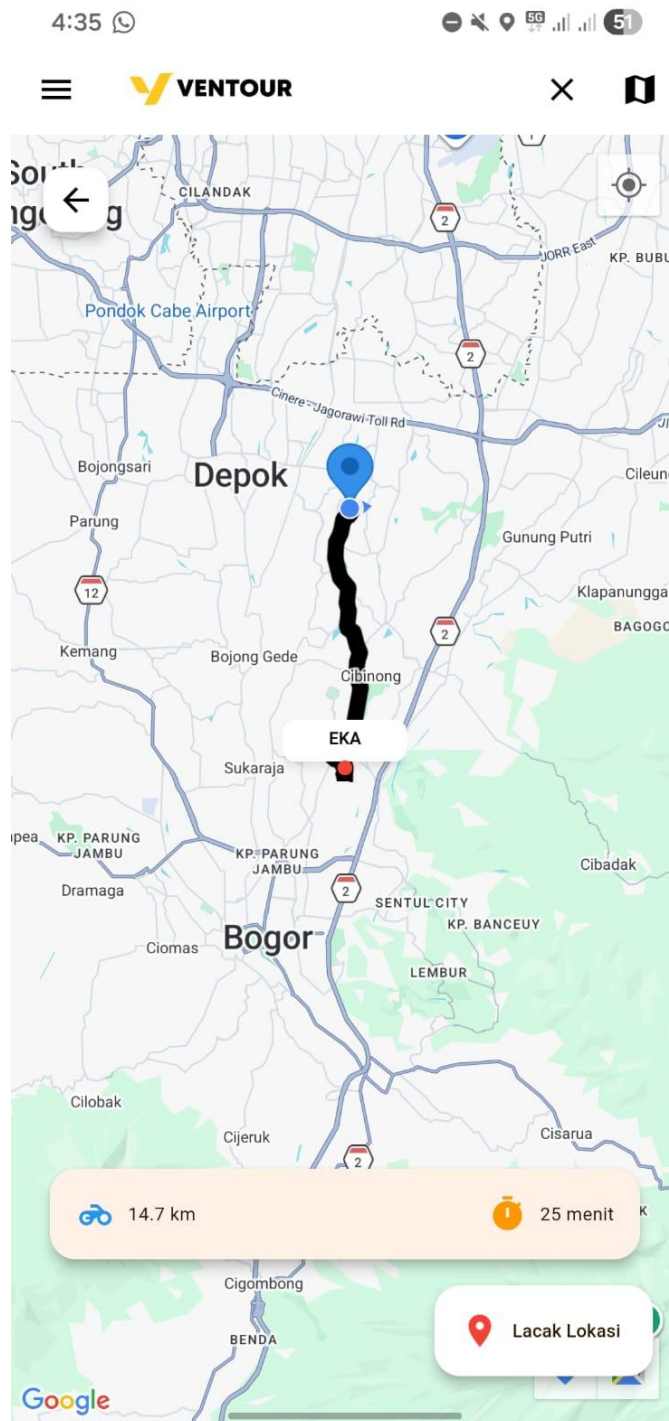
Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)

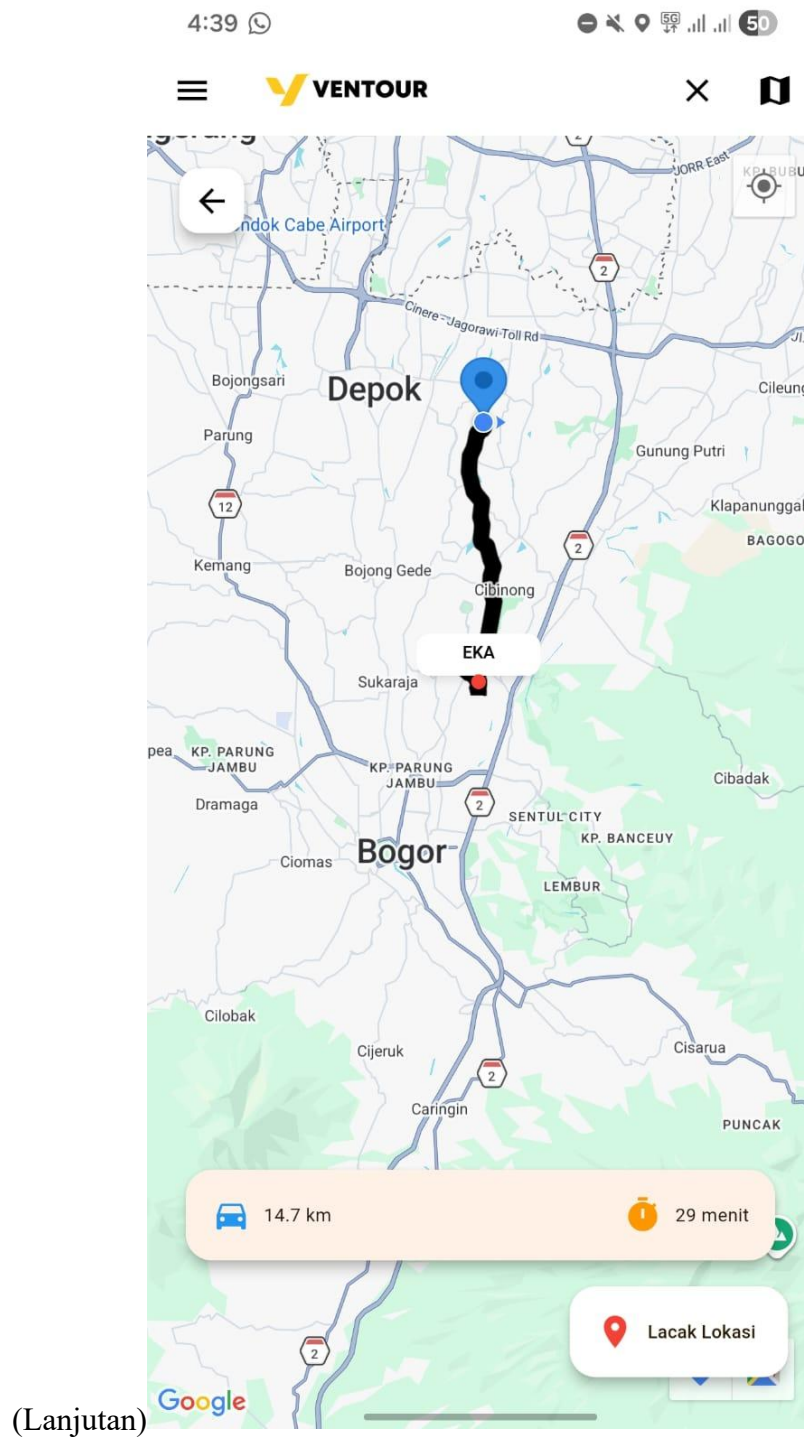


Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)

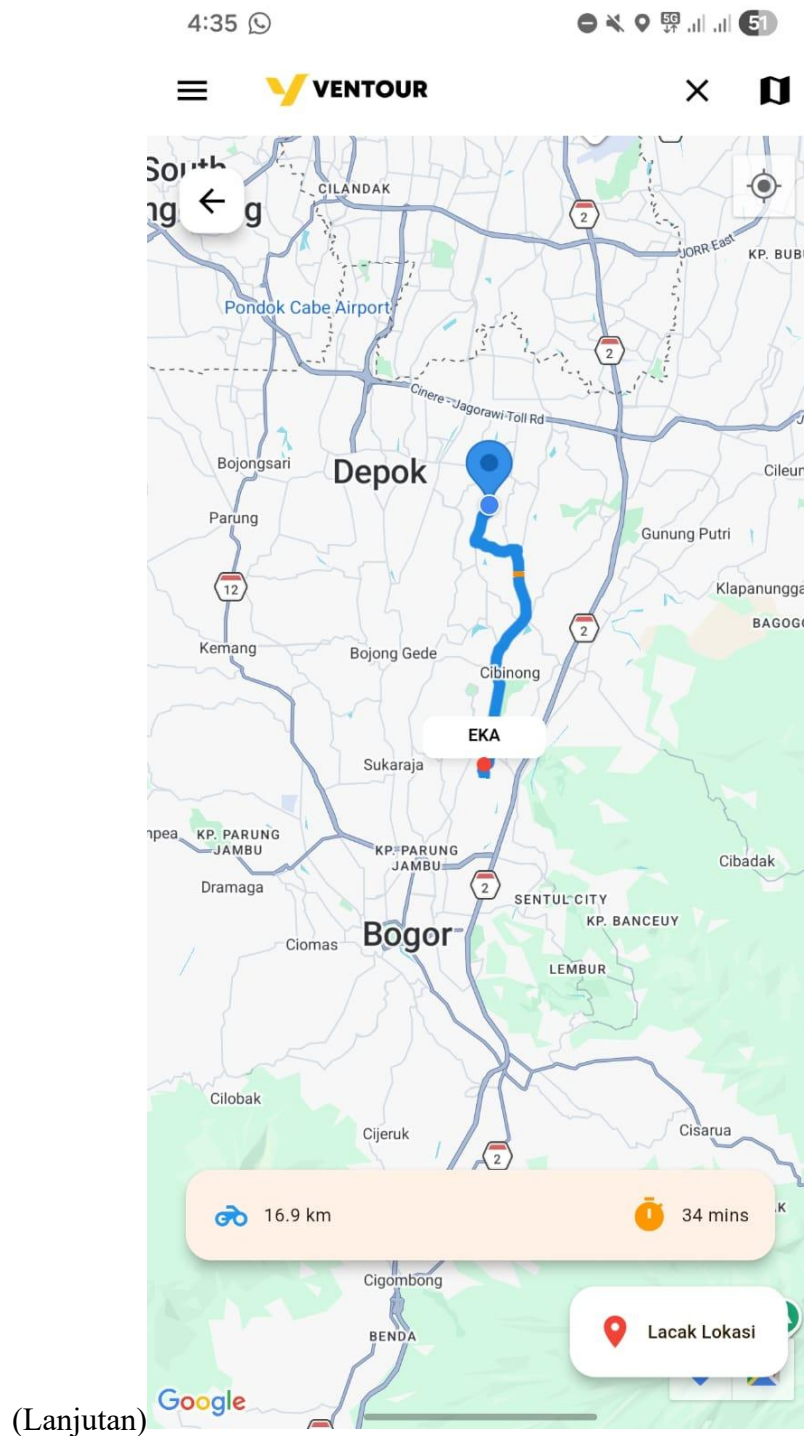


Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)

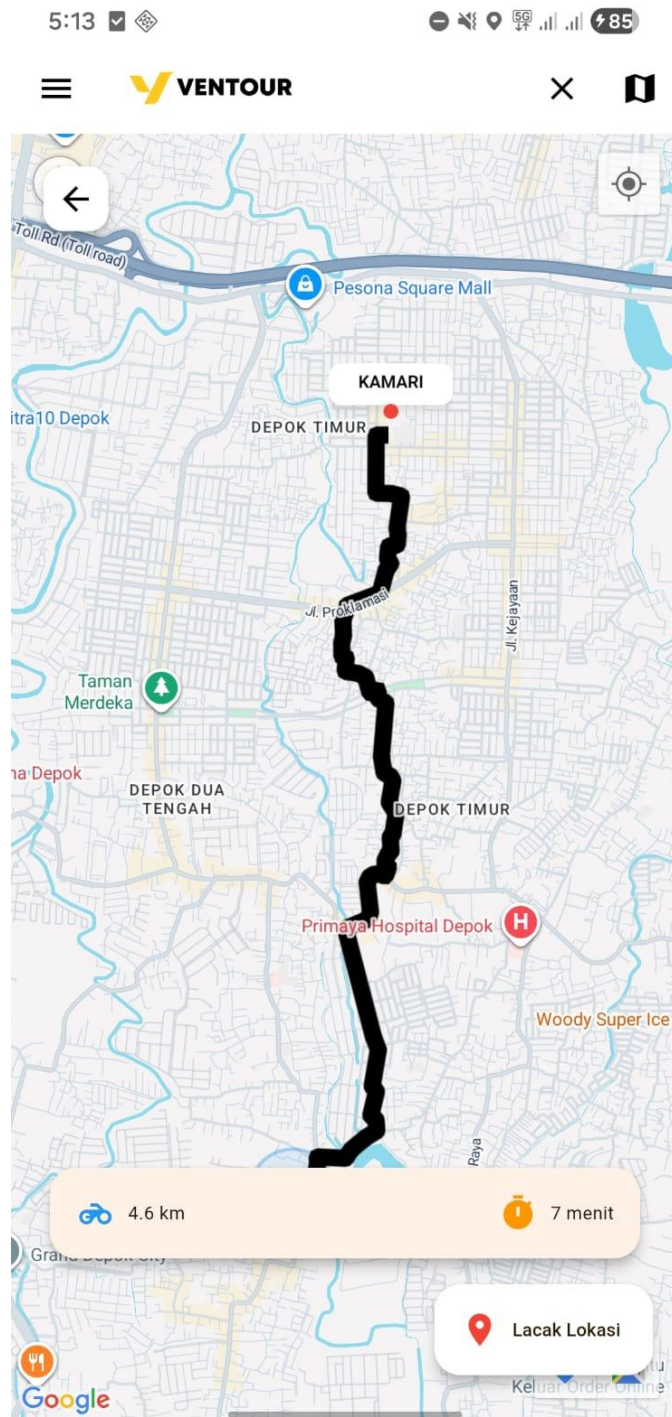


Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*

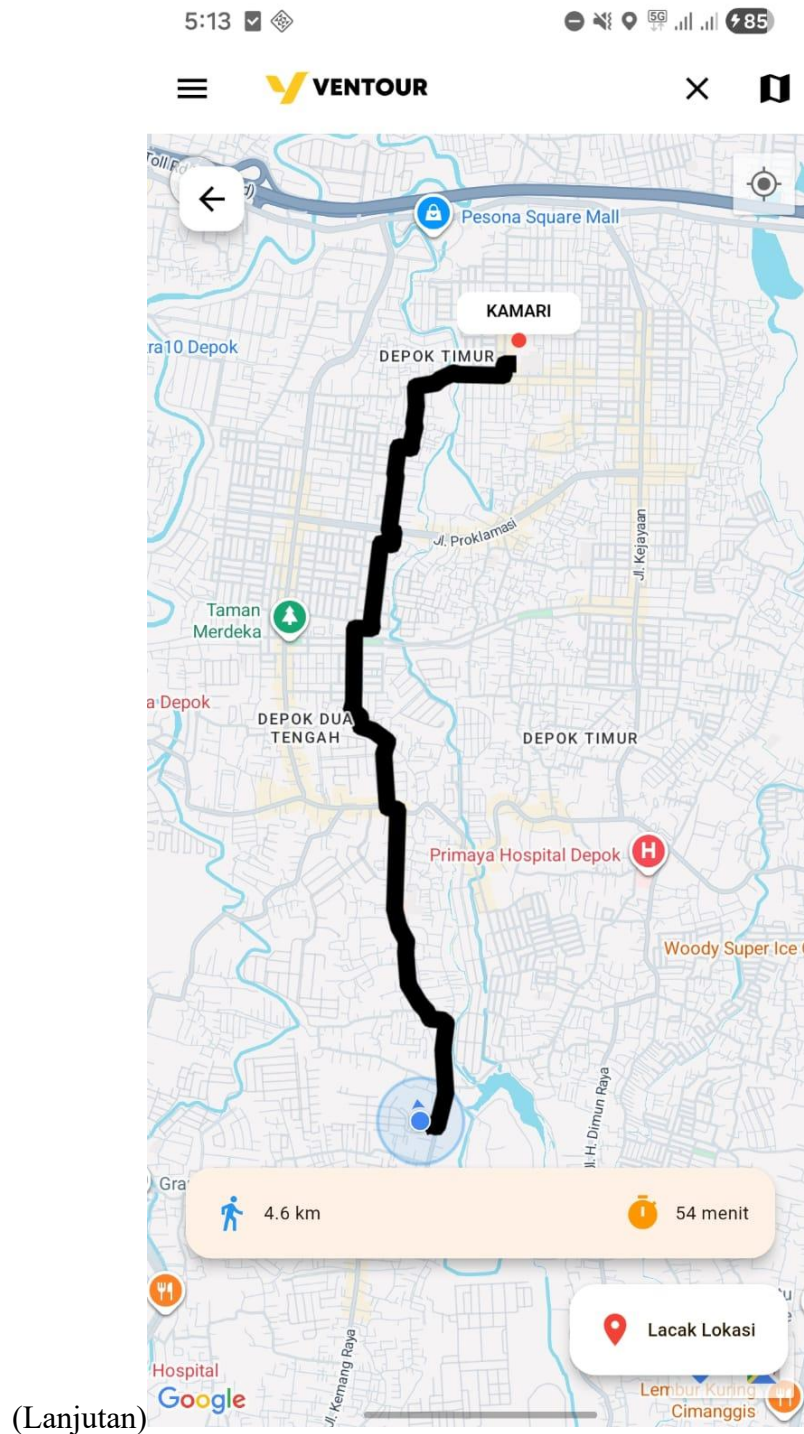
(Lanjutan)

Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*

Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)

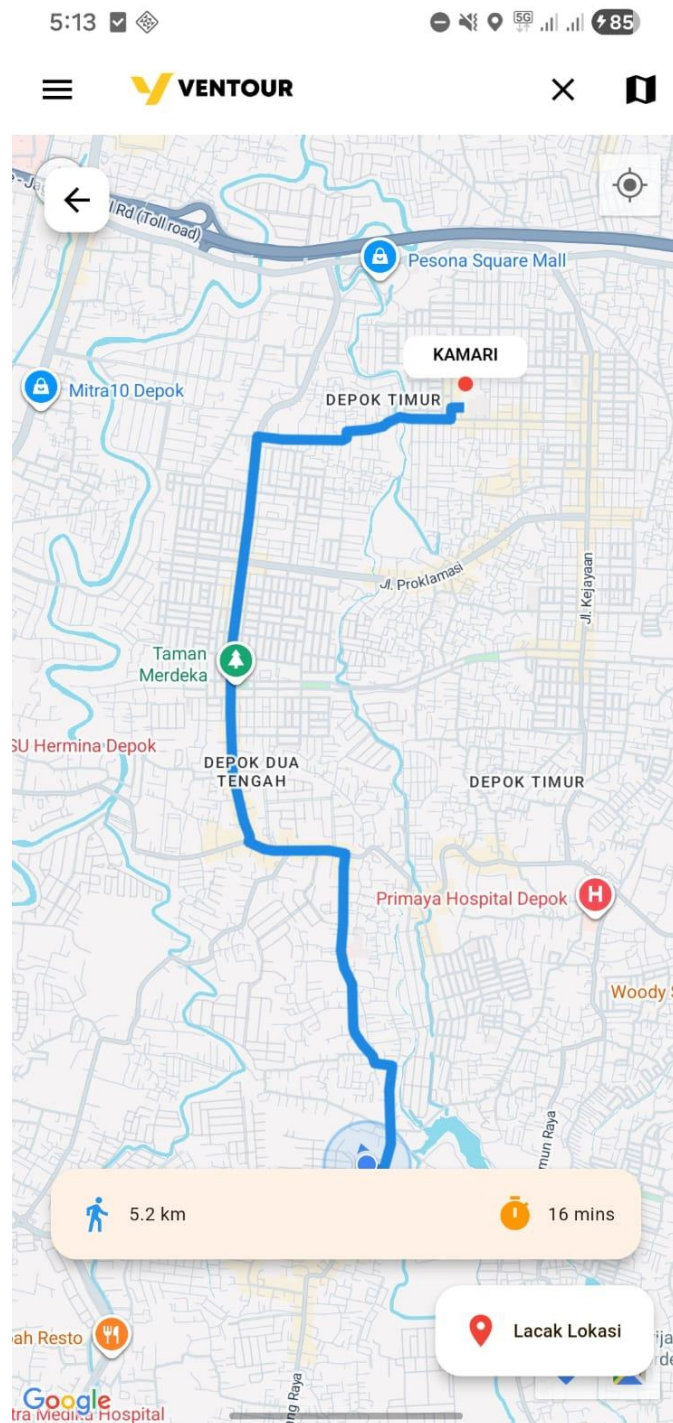


Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*

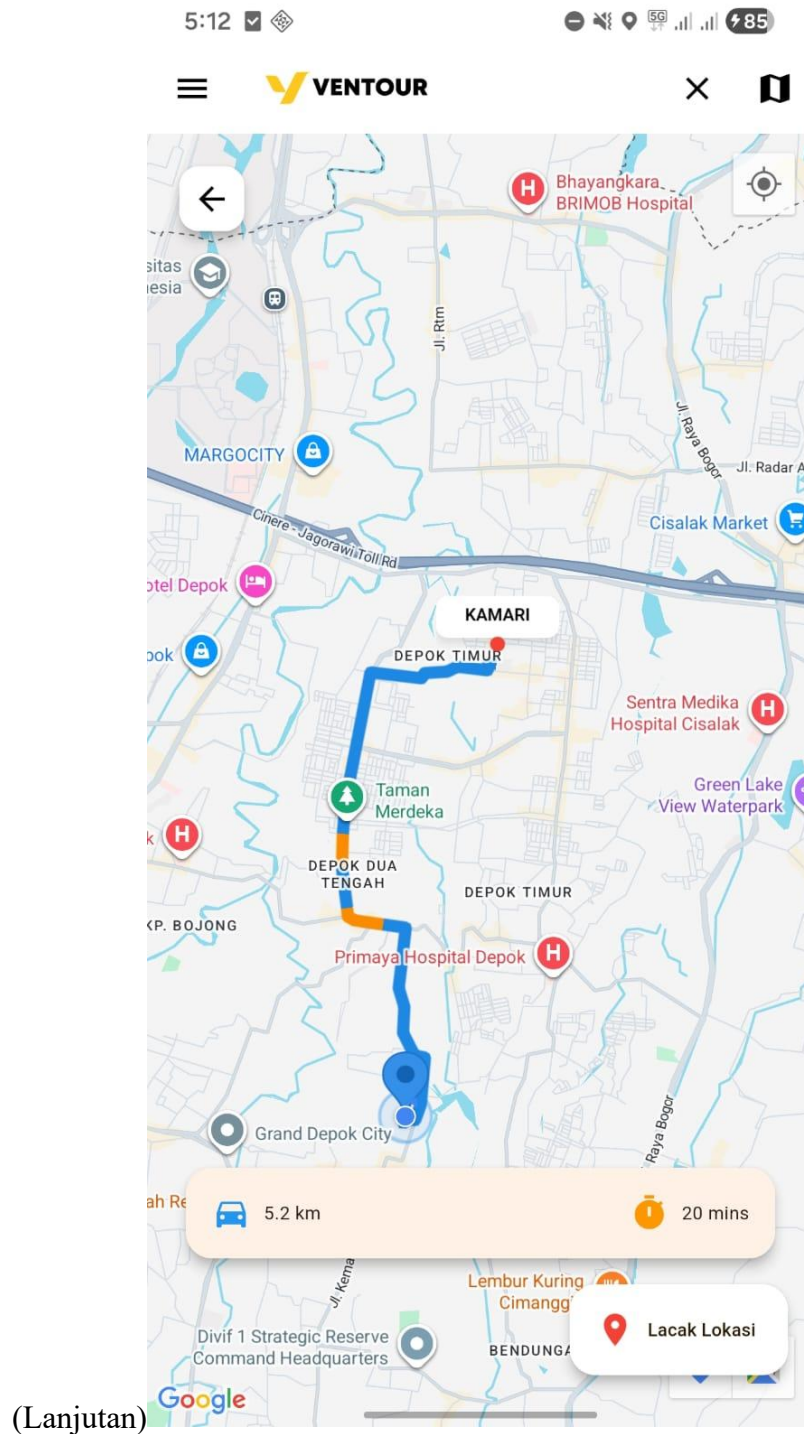


(Lanjutan)

Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)

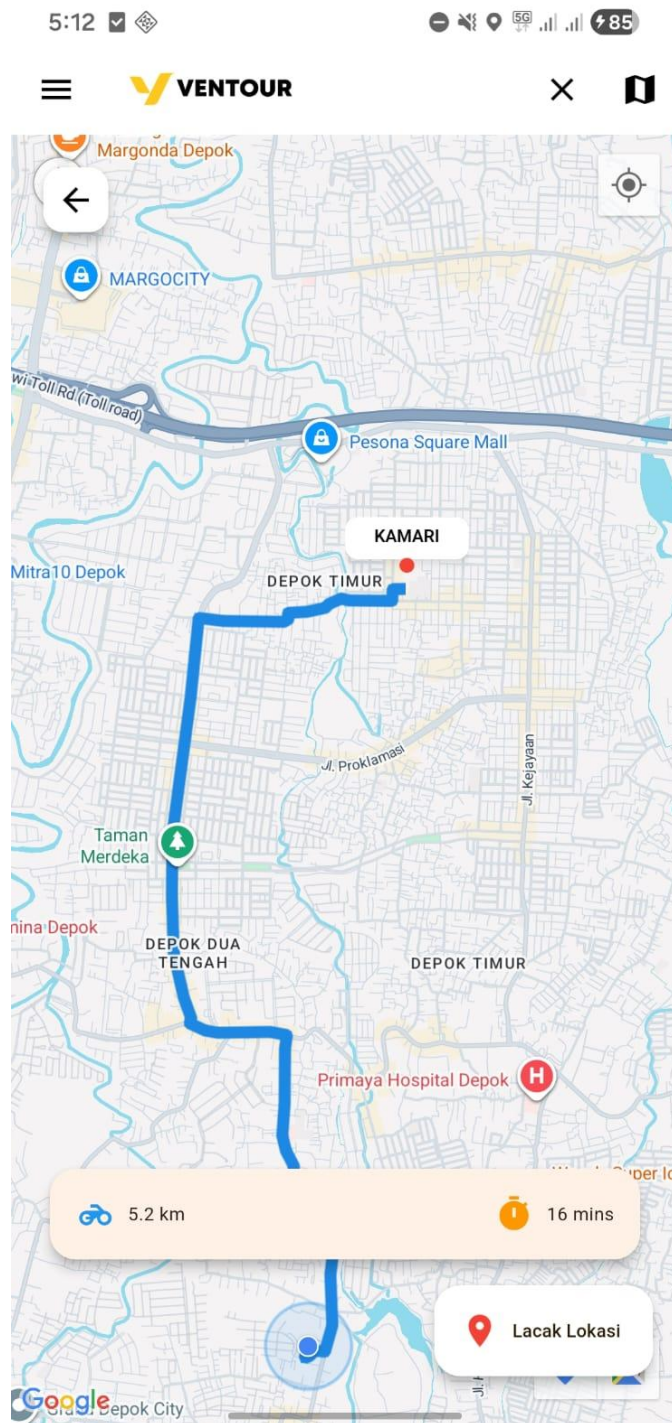


Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*

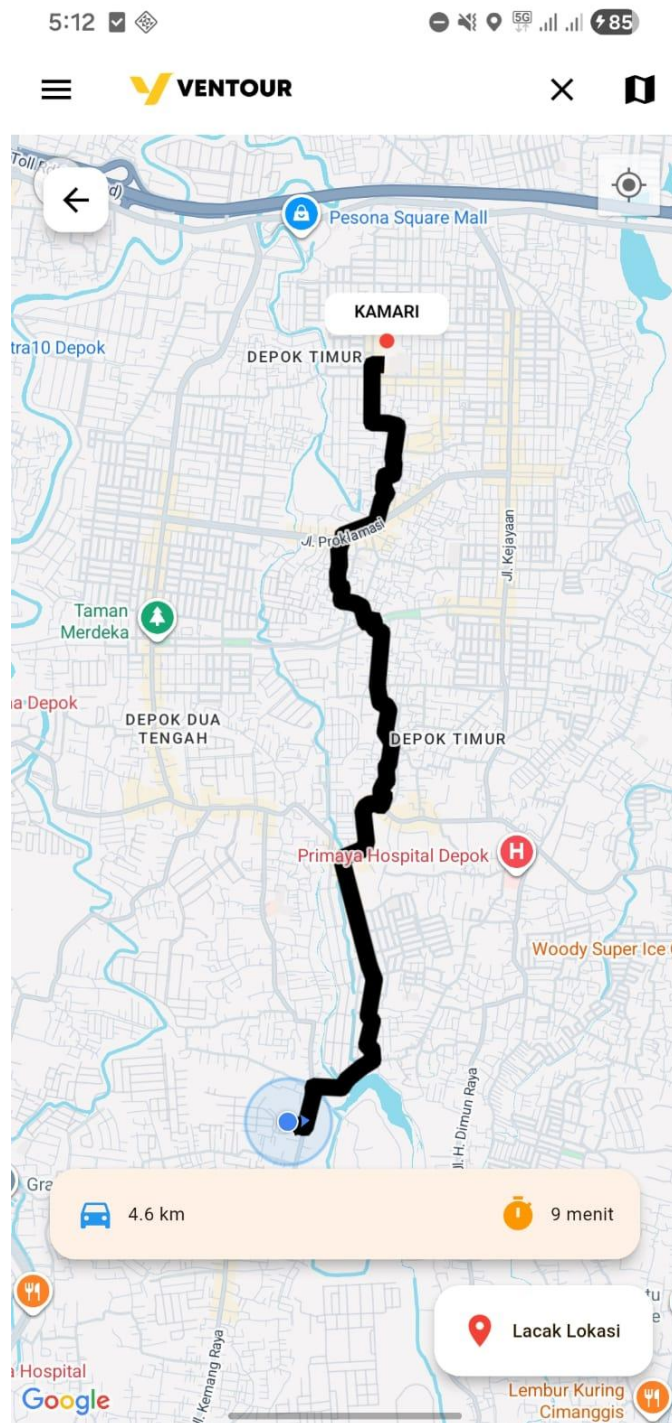


(Lanjutan)

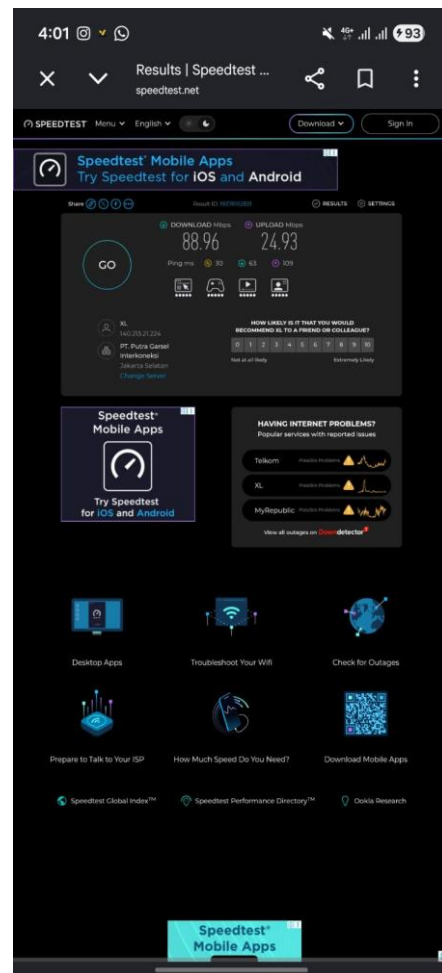
Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)



Lampiran 3 Bukti pengujian perbandingan metode A* dengan *Google Maps*
(Lanjutan)



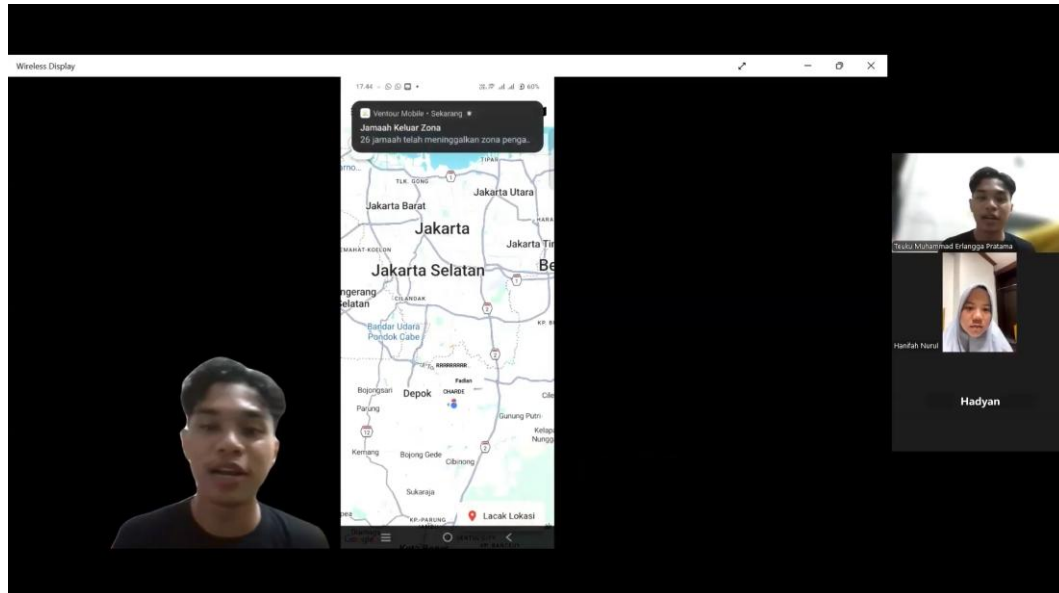
Lampiran 5 Bukti Pengujian Kecepatan Internet



Lampiran 5 Sosialisasi Fitur dengan *Tour Leader*

Lokasi: Zoom

Waktu: Sabtu, 9 Mei 2026



Lampiran 6 Hasil Responden User Acceptance Testing



Lampiran 6 Hasil Responden User Acceptance Testing (lanjutan)



Lampiran 6 Hasil Responden User Acceptance Testing (lanjutan)



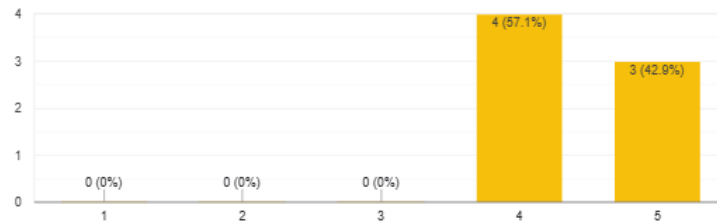
Lampiran 6 Hasil Responden User Acceptance Testing (lanjutan)

Aspek Efisien & Produktivitas

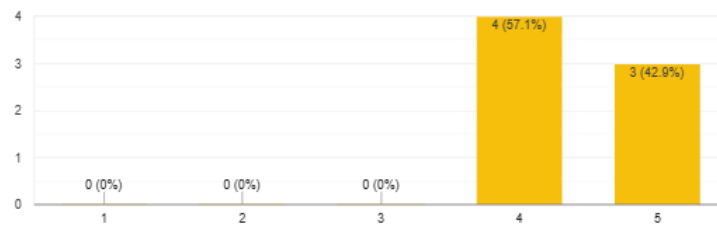
Fitur mempersingkat tahapan pencarian jamaah

[Copy chart](#)

7 responses

Fitur mempercepat proses pengumpulan jamaah dibandingkan metode manual atau penggunaan alat pelacak sebelumnya (*Mi Tag*).[Copy chart](#)

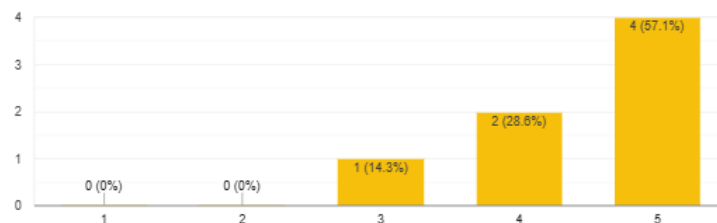
7 responses



Pengambilan keputusan perjalanan menjadi lebih cepat

[Copy chart](#)

7 responses

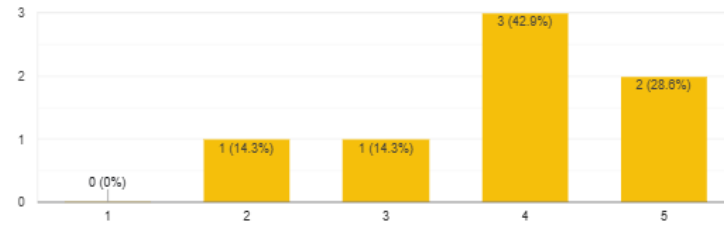


Lampiran 5 Bukti Hasil Responden SUS

Saya berpikir akan menggunakan fitur ini lagi.

[Copy chart](#)

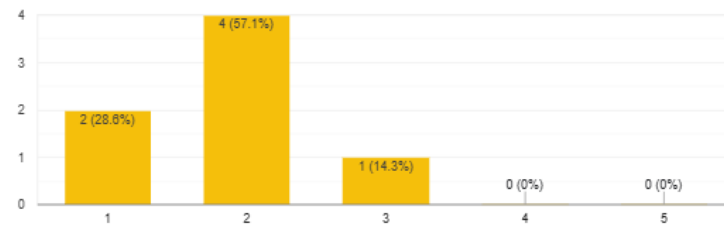
7 responses



Saya merasa fitur ini rumit untuk digunakan.

[Copy chart](#)

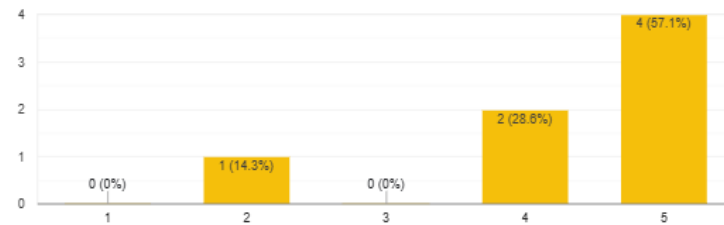
7 responses



Saya merasa fitur ini mudah untuk digunakan.

[Copy chart](#)

7 responses



Lampiran 5 Bukti Hasil Responden SUS (lanjutan)

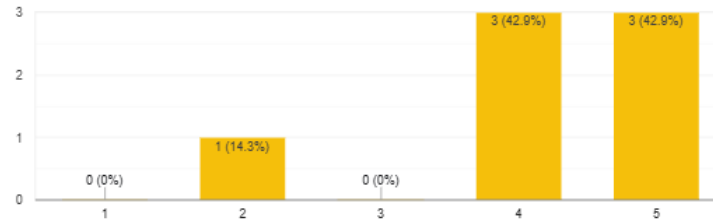


Lampiran 5 Bukti Hasil Responden SUS (lanjutan)

Saya merasa orang lain akan memahami cara menggunakan fitur ini dengan cepat.

[Copy chart](#)

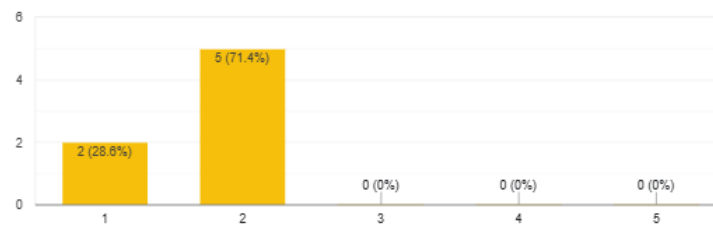
7 responses



Saya merasa fitur ini membingungkan.

[Copy chart](#)

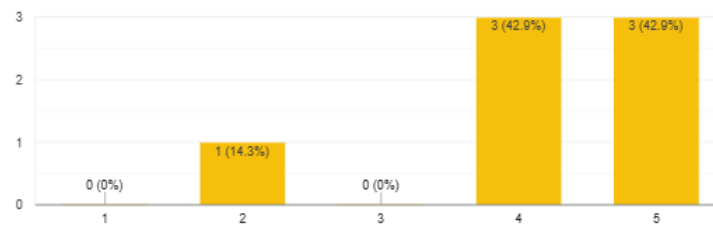
7 responses



Saya merasa percaya diri dalam menggunakan fitur ini.

[Copy chart](#)

7 responses



Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

[Copy chart](#)

7 responses

