



Institut Riset dan Publikasi Indonesia (IRPI)

MALCOM: Indonesian Journal of Machine Learning and Computer Science

Journal Homepage: <https://journal.irpi.or.id/index.php/malcom>

Vol. 6 Iss. 3 July 2026, pp: 1312-1323

ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Development of a Location-Based GeoTrackOps Mobile Application with Offline Synchronization Using Agile Scrum for Field Teams

Pengembangan Aplikasi Mobile GeoTrackOps Berbasis Lokasi dengan Sinkronisasi Offline Menggunakan Agile Scrum untuk Tim Lapangan

**Jhosua A. Mataputun^{1*}, Lilis Indrayani²,
Christian D. Suhendra³, Iban Getarjati⁴**

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Papua, Indonesia

⁴PT Geosain Delta Andalan, Indonesia

E-Mail: ¹jhosuamataputun@gmail.com, ²lilisindrayani8@gmail.com,
³c.suhendra@unipa.ac.id, ⁴getarjati96@gmail.com

Received Apr 15th 2026; Revised May 24th 2026; Accepted Jun 12th 2026; Available Online Jun 25th 2026

Corresponding Author: Jhosua A. Mataputun

Copyright © 2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

GeoTrackOps is a mobile application developed to support field team operations through location-based reporting and offline-online data synchronization. The initial implementation still had limitations in GPS/LBS stability and data synchronization when devices switched from offline to online. This study aims to enhance GeoTrackOps by improving location tracking, SQLite-based local storage, and offline-online synchronization mechanisms for field activities. The study used a Research and Development approach with Agile Scrum through requirement analysis, product backlog preparation, sprint-based implementation, and iterative evaluation. The developed features include route management, stopsite reporting, GPS/LBS tracking, local data queues, backend synchronization, and a role-based web dashboard for Admin and Basecamp Operator. Evaluation was conducted using Black-Box Testing and technical performance testing on two Android devices along a 3.2 km test route. The results showed that all 15 functional scenarios passed, covering login, role-based access, route and stopsite workflows, GPS/LBS tracking, offline storage, synchronization, and dashboard access. Technical performance testing showed an average synchronization time of 12.4 seconds, an average dashboard position latency of 6.7 seconds, a data delivery success rate of 100%, and average GPS/LBS errors of 8.3 meters in open areas and 24.1 meters in closed areas. These results indicate that GeoTrackOps can support more reliable field reporting and monitoring, especially in areas with limited network connectivity.

Keywords: Agile Scrum, Black-Box Testing, Geotrackops, Location-Based Service, Offline Synchronization

Abstrak

GeoTrackOps merupakan aplikasi mobile yang dikembangkan untuk mendukung operasional tim lapangan melalui pelaporan berbasis lokasi dan sinkronisasi data offline-online. Implementasi awal masih memiliki keterbatasan pada kestabilan GPS/LBS dan sinkronisasi data ketika perangkat berpindah dari kondisi offline ke online. Penelitian ini bertujuan untuk menyempurnakan GeoTrackOps melalui peningkatan pelacakan lokasi, penyimpanan lokal berbasis SQLite, dan mekanisme sinkronisasi offline-online untuk aktivitas lapangan. Penelitian ini menggunakan pendekatan Research and Development dengan Agile Scrum melalui analisis kebutuhan, penyusunan product backlog, implementasi berbasis sprint, dan evaluasi iteratif. Fitur yang dikembangkan meliputi manajemen route, pelaporan stopsite, GPS/LBS tracking, antrean data lokal, sinkronisasi ke backend server, serta dashboard web berbasis peran untuk Admin dan Basecamp Operator. Evaluasi dilakukan menggunakan Black-Box Testing dan pengujian performa teknis pada dua perangkat Android di rute uji sepanjang 3,2 km. Hasil pengujian menunjukkan bahwa seluruh 15 skenario fungsional memperoleh status lulus, mencakup login, akses berbasis peran, alur route dan stopsite, GPS/LBS tracking, penyimpanan offline, sinkronisasi, dan akses dashboard. Pengujian performa teknis menunjukkan waktu sinkronisasi rata-rata 12,4 detik, latensi posisi dashboard rata-rata 6,7 detik, tingkat keberhasilan pengiriman data 100%, serta error rata-rata GPS/LBS sebesar 8,3 meter pada area terbuka dan 24,1 meter pada area tertutup. Hasil ini menunjukkan bahwa GeoTrackOps dapat mendukung pelaporan dan monitoring lapangan secara lebih andal, terutama pada wilayah dengan keterbatasan konektivitas jaringan.

Kata Kunci: Agile Scrum, Geotrackops, Layanan Berbasis Lokasi, Pengujian Black-Box, Sinkronisasi Offline



DOI: <https://doi.org/10.57152/malcom.v6i3.2797>

1. PENDAHULUAN

Transformasi digital pada sektor operasional lapangan mendorong kebutuhan terhadap sistem kerja yang lebih cepat, terstruktur, dan terintegrasi. Proses pelaporan yang sebelumnya banyak bergantung pada pencatatan manual mulai diarahkan ke aplikasi *mobile* agar pencatatan aktivitas, dokumentasi, dan pemantauan pekerjaan dapat dilakukan secara lebih efisien. Perkembangan teknologi digital dalam manajemen operasional juga dinilai mampu mengubah pola kerja organisasi melalui pemanfaatan data yang lebih terhubung [1]. Selain itu, aplikasi *mobile* banyak digunakan sebagai media pengumpulan data karena dapat mendukung pencatatan berbasis perangkat, penggunaan sensor, serta integrasi dengan sistem penyimpanan data terpusat [2].

Meskipun demikian, penerapan aplikasi *mobile* pada kegiatan lapangan masih menghadapi kendala, terutama ketika petugas bekerja pada area dengan koneksi internet yang tidak stabil. Kondisi tersebut dapat menyebabkan keterlambatan pengiriman laporan, duplikasi pencatatan, dan ketidaksesuaian antara data pada perangkat pengguna dan server. Oleh karena itu, aplikasi lapangan memerlukan dukungan mode *offline* dan mekanisme sinkronisasi data agar pencatatan tetap dapat dilakukan ketika jaringan terbatas dan data dapat dikirimkan kembali saat koneksi tersedia [3].

Pendekatan *Location-Based Service* (LBS) menjadi relevan dalam sistem operasional lapangan karena memungkinkan aplikasi memanfaatkan informasi lokasi pengguna sebagai bagian dari proses pelaporan dan pemantauan. Data lokasi dapat memperjelas konteks aktivitas, mendukung keterlacakan pekerjaan, serta membantu pihak pengelola dalam memonitor progres tim lapangan. Kajian mengenai LBS menunjukkan bahwa teknologi ini telah digunakan pada berbagai kebutuhan, seperti navigasi, logistik, sistem industri, dan layanan berbasis perangkat *mobile* [4].

Beberapa penelitian sebelumnya menunjukkan bahwa aplikasi *mobile*, layanan berbasis lokasi, dan sinkronisasi data memiliki peran penting dalam mendukung aktivitas berbasis perangkat. Harris et al. (2022) mengembangkan platform *mobile* untuk mendukung pengumpulan data berbasis perangkat dan integrasi data secara terpusat [2]. Wang & Wang (2024) menjelaskan bahwa mode *offline* dan sinkronisasi data diperlukan agar aplikasi *mobile* tetap dapat digunakan ketika koneksi jaringan tidak stabil [3]. Sementara itu, Kung et al. (2023) menunjukkan bahwa *Location-Based Service* dapat digunakan untuk mendukung kebutuhan navigasi, pemantauan, dan layanan berbasis posisi pengguna [4].

Berdasarkan kajian tersebut, masih terdapat keterbatasan penelitian yang membahas integrasi pelaporan *route* dan *stopsite*, GPS/LBS *tracking*, penyimpanan lokal SQLite, sinkronisasi *offline-online*, serta *dashboard* berbasis peran dalam satu aplikasi operasional lapangan. Oleh karena itu, penelitian ini memiliki kebaruan pada pengembangan GeoTrackOps menggunakan *Agile Scrum* untuk menyempurnakan fungsi pelaporan berbasis lokasi dan sinkronisasi data pada kondisi jaringan terbatas. Integrasi aplikasi *mobile* untuk Field User, *backend* server, penyimpanan lokal, dan *dashboard web* untuk Admin serta Basecamp Operator menjadi kontribusi utama dalam mendukung pelaporan dan *monitoring* aktivitas lapangan secara lebih andal.

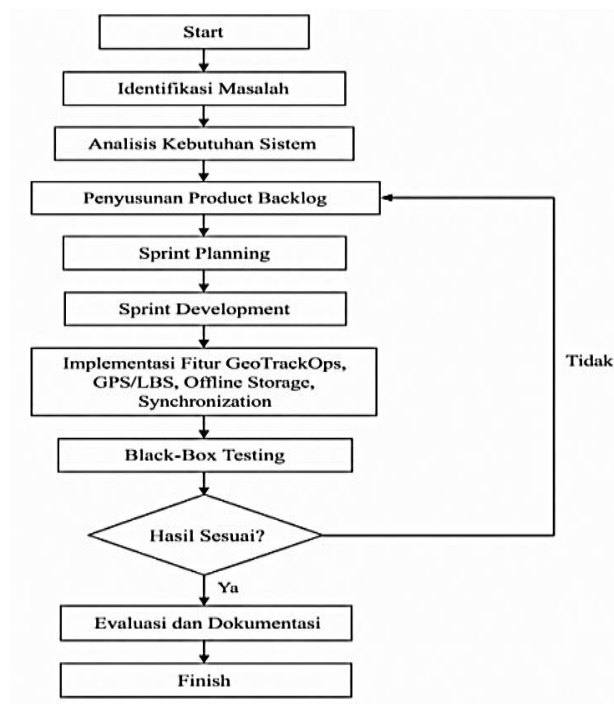
Berdasarkan implementasi awal aplikasi GeoTrackOps, masih ditemukan kendala pada fitur utama sistem. Fitur GPS belum sepenuhnya stabil, seperti kesalahan pembacaan lokasi, keterlambatan pembaruan koordinat, dan tingkat akurasi yang belum konsisten saat digunakan di lapangan. Permasalahan akurasi pada *smartphone* dapat dipengaruhi oleh keterbatasan sensor, kualitas sinyal, serta karakteristik pengukuran GNSS pada perangkat *mobile* [5]. Selain itu, latensi pada aplikasi GNSS berbasis *mobile* juga dapat memengaruhi ketepatan pembaruan posisi ketika data harus diproses dan dikirim melalui jaringan [6]. Pada sisi lain, mekanisme sinkronisasi *offline-online* pada GeoTrackOps juga belum berjalan optimal karena data yang disimpan saat *offline* masih berpotensi gagal tersinkronisasi ketika koneksi internet kembali tersedia.

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada pengembangan aplikasi *mobile* GeoTrackOps berbasis lokasi dengan dukungan sinkronisasi *offline-online* untuk tim lapangan. Pengembangan diarahkan pada peningkatan kestabilan pengambilan lokasi, penyimpanan data lokal saat perangkat berada dalam kondisi *offline*, serta penyempurnaan proses sinkronisasi data ketika perangkat kembali *online*. Aplikasi ini dirancang untuk mengakomodasi tiga peran pengguna, yaitu Admin, Basecamp Operator, dan Field User. Dengan pengembangan ini, GeoTrackOps diharapkan dapat mendukung proses pelaporan lapangan secara lebih andal, meningkatkan kualitas *monitoring*, dan mempercepat ketersediaan informasi operasional bagi pihak pengelola.

2. METODE PENELITIAN

Bagian ini menjelaskan tahapan penelitian dan pengembangan aplikasi GeoTrackOps. Penelitian menggunakan pendekatan *Research and Development* (R&D) karena berfokus pada penyempurnaan produk aplikasi *mobile* yang telah tersedia sebelumnya. Pendekatan R&D sesuai digunakan pada penelitian yang menghasilkan atau menyempurnakan produk melalui tahapan analisis kebutuhan, pengembangan, implementasi, dan evaluasi [7]. Proses pengembangan dilakukan menggunakan *Agile Scrum* karena metode ini mendukung pengembangan perangkat lunak secara iteratif melalui penyusunan *backlog*, perencanaan *sprint*, implementasi fitur, inspeksi, dan perbaikan berkelanjutan [8], [9]. Secara teknis, pengembangan diarahkan pada integrasi *Location-Based Service* (LBS), penyimpanan lokal berbasis SQLite, mekanisme

sinkronisasi *offline-online*, serta arsitektur *client-server* untuk mendukung pencatatan dan monitoring aktivitas lapangan. Evaluasi fungsi sistem dilakukan menggunakan *Black-Box Testing*, yaitu pengujian yang berfokus pada kesesuaian masukan dan keluaran sistem berdasarkan skenario uji tanpa meninjau struktur internal kode program [10]. Alur penelitian divisualisasikan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Tahapan Penelitian

Rangkaian penelitian pada Gambar 1 dilakukan secara bertahap untuk memastikan proses pengembangan berjalan secara sistematis. Tahap pertama adalah identifikasi masalah, yaitu menelaah kendala pada aplikasi GeoTrackOps awal, terutama terkait kestabilan GPS/LBS dan sinkronisasi *offline-online*. Tahap kedua adalah analisis kebutuhan sistem berdasarkan tiga peran pengguna, yaitu Admin, Basecamp Operator, dan Field User. Hasil analisis kebutuhan kemudian digunakan untuk menyusun *product backlog* yang memuat prioritas perbaikan fitur.

Tahap berikutnya adalah *sprint planning*, yaitu menentukan *backlog* yang akan dikerjakan pada setiap iterasi pengembangan. Setelah itu, dilakukan *sprint development* untuk mengimplementasikan fitur utama, meliputi pencatatan aktivitas lapangan, pemantauan lokasi, penyimpanan lokal menggunakan SQLite, dan sinkronisasi data ke *backend*. Fitur yang telah dikembangkan diuji menggunakan *Black-Box Testing* untuk memastikan fungsi berjalan sesuai dengan hasil yang diharapkan. Apabila hasil pengujian belum sesuai, fitur dikembalikan ke *product backlog* untuk diperbaiki pada iterasi berikutnya. Jika seluruh skenario sudah sesuai, proses dilanjutkan ke tahap evaluasi dan dokumentasi hasil penelitian.

2.2. Pengembangan Agile Scrum

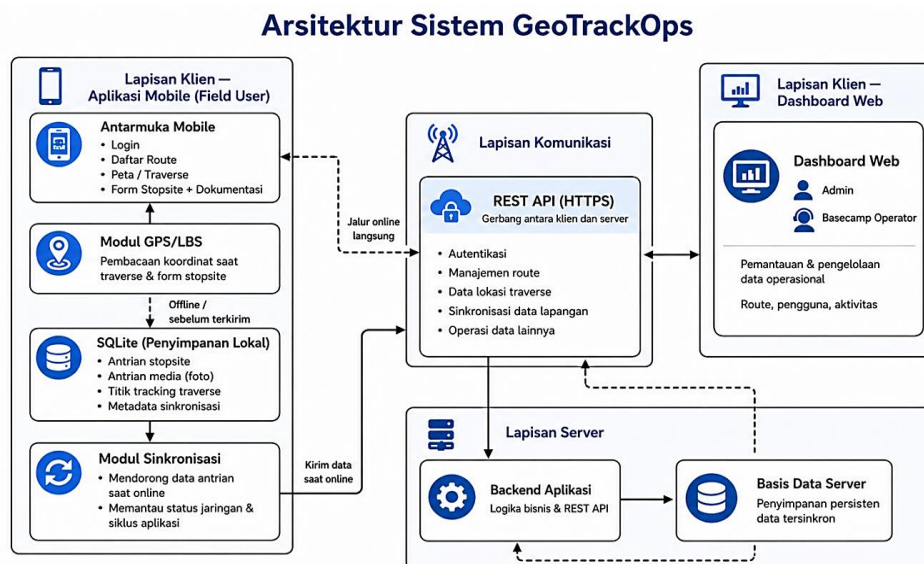
Pengembangan GeoTrackOps dilakukan menggunakan pendekatan *Agile Scrum* karena aplikasi yang dikembangkan bukan dibangun dari awal, melainkan disempurnakan dari sistem yang telah tersedia sebelumnya. Scrum dipilih karena mendukung proses pengembangan perangkat lunak secara iteratif melalui perencanaan, implementasi, inspeksi, dan adaptasi terhadap perubahan kebutuhan. Pendekatan ini juga membantu tim menjaga kualitas perangkat lunak melalui kolaborasi, transparansi, dan evaluasi berulang pada setiap siklus pengembangan [8], [9].

Pada penelitian ini, *product backlog* disusun berdasarkan temuan pada aplikasi awal, terutama pada fitur GPS/LBS, penyimpanan data lokal, dan sinkronisasi *offline-online*. Setiap *sprint* dimulai dengan pemilihan *backlog* prioritas, dilanjutkan dengan pengembangan fitur, koordinasi pengerjaan, dan peninjauan hasil. Fitur yang belum sesuai dengan kebutuhan dikembalikan ke *backlog* untuk diperbaiki pada iterasi berikutnya. Pendekatan Scrum relevan untuk pengembangan aplikasi *mobile* karena dapat membantu penyesuaian sistem terhadap kebutuhan proyek yang berubah selama proses pengembangan [11]. Evaluasi pada akhir *sprint* juga digunakan untuk memantau ketercapaian pekerjaan dan menentukan perbaikan pada siklus berikutnya [12]. Dengan demikian, penerapan Agile Scrum pada penelitian ini digunakan untuk

memastikan proses penyempurnaan fitur GPS/LBS, penyimpanan lokal, dan sinkronisasi offline-online dapat dilakukan secara bertahap, terkontrol, dan sesuai kebutuhan pengguna.

2.3. Arsitektur Sistem

Arsitektur GeoTrackOps dirancang menggunakan pola *client-server* dengan dukungan *offline-first* pada aplikasi *mobile*. Rancangan ini digunakan agar *Field User* tetap dapat mencatat aktivitas lapangan ketika koneksi internet terbatas, kemudian data yang tersimpan secara lokal dapat dikirimkan ke server saat koneksi tersedia kembali. Strategi mode *offline* dan sinkronisasi data diperlukan pada aplikasi *mobile* yang digunakan dalam kondisi jaringan tidak stabil atau terputus-putus [3]. Rancangan arsitektur sistem GeoTrackOps ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Sistem GeoTrackOps

Secara umum, sistem terdiri dari aplikasi *mobile*, layanan GPS/LBS, penyimpanan lokal, modul sinkronisasi, REST API/backend, basis data server, dan *dashboard web*. Pola ini digunakan untuk memisahkan fungsi pencatatan data pada sisi klien, pemrosesan dan penyimpanan permanen pada sisi server, serta pemantauan aktivitas melalui *dashboard web*. Dengan dukungan *offline-first*, data yang belum dapat dikirim ke server tetap disimpan pada perangkat melalui SQLite, sehingga risiko kehilangan data saat koneksi tidak stabil dapat dikurangi. Aplikasi *mobile* digunakan oleh Field User untuk memilih rute, mencatat stopsite, mengambil dokumentasi, dan mengirim data aktivitas. Layanan GPS/LBS digunakan untuk memperoleh koordinat pengguna selama aktivitas lapangan berlangsung. Pemanfaatan aplikasi mobile berbasis lokasi relevan untuk kebutuhan profesional yang melibatkan GPS, peta, survei, navigasi, dan pengumpulan data lapangan [4], [13].

Pada sisi perangkat, SQLite digunakan sebagai penyimpanan lokal untuk menampung data yang belum dapat dikirim ke server, seperti data stopsite, titik pelacakan, media, dan antrian sinkronisasi. Pendekatan penyimpanan lokal seperti ini juga diterapkan pada pengembangan aplikasi *mobile* yang memisahkan basis data lokal dan server, kemudian menyinkronkan data melalui API ketika koneksi tersedia [14].

Pada sisi server, REST API/backend menyediakan layanan autentikasi, manajemen rute, penerimaan data lokasi, sinkronisasi data lapangan, dan pengelolaan data operasional. Data yang berhasil dikirim dari aplikasi *mobile* disimpan pada basis data server dan dapat dipantau melalui *dashboard web* oleh Admin dan Basecamp Operator. Dengan arsitektur ini, GeoTrackOps dapat mendukung pencatatan lapangan berbasis lokasi, penyimpanan data saat *offline*, serta sinkronisasi data ke server secara lebih terstruktur.

2.4. Implementasi Location-Based Service

Implementasi Location-Based Service (LBS) pada GeoTrackOps digunakan untuk mencatat posisi Field User selama aktivitas lapangan dan menghubungkan data lokasi dengan proses pelaporan. LBS relevan digunakan pada aplikasi operasional karena dapat mendukung pelacakan, monitoring, navigasi, dan pengelolaan aktivitas berbasis posisi pengguna [4], [13].

Pada aplikasi *mobile*, posisi pengguna diperoleh melalui layanan *geolocation* perangkat dengan data utama berupa *latitude*, *longitude*, *accuracy*, dan *timestamp*. Data tersebut tidak disimpan sebagai koordinat yang berdiri sendiri, tetapi dikaitkan dengan konteks operasional seperti *user_id* dan *route_id*, sehingga

setiap titik lokasi dapat ditelusuri berdasarkan pengguna dan rute aktif. Informasi lokasi juga digunakan pada form *stopsite* untuk membantu pencatatan titik kerja di lapangan.

Untuk menjaga akurasi dan reliabilitas data lokasi, aplikasi melakukan validasi terhadap status pengguna, rute aktif, dan izin lokasi sebelum proses pelacakan dijalankan. Selain itu, sistem menangani kegagalan pembacaan GPS dan membatasi pengiriman data lokasi agar tidak membebani komunikasi ke *backend*. Kualitas posisi pada *smartphone* dapat dipengaruhi oleh kondisi lingkungan, keterbatasan sensor, dan karakteristik pengukuran GNSS perangkat [5]. Pada aplikasi berbasis posisi *real-time*, latensi proses akuisisi dan pengiriman data juga dapat memengaruhi kecepatan pembaruan lokasi [6].

GeoTrackOps juga mendukung pencatatan lokasi pada kondisi *offline*. Apabila koneksi internet tidak tersedia atau pengiriman ke *backend* gagal, titik lokasi disimpan sementara pada penyimpanan lokal sebagai antrean sinkronisasi. Ketika koneksi kembali tersedia, data lokasi yang tertunda dikirim ulang ke server. Strategi penyimpanan lokal dan sinkronisasi ulang diperlukan agar aplikasi *mobile* tetap andal pada kondisi jaringan yang buruk atau terputus-putus [3].

2.5. Mekanisme Sinkronisasi Offline-Online

Mekanisme sinkronisasi pada GeoTrackOps menerapkan pendekatan *offline-first*, yaitu data lapangan tetap dapat dicatat pada perangkat *mobile* ketika koneksi internet tidak tersedia, kemudian dikirimkan ke server saat jaringan kembali tersedia. Pendekatan ini diperlukan karena aktivitas tim lapangan tidak selalu berlangsung pada area dengan kualitas jaringan yang stabil. Pada aplikasi *mobile*, mode *offline* dan strategi sinkronisasi data berperan penting untuk menjaga keandalan sistem serta mengurangi risiko kehilangan data [3], [15].

Pada GeoTrackOps, data yang dibuat saat *offline*, seperti data *stopsite*, media, dan titik tracking lokasi, disimpan terlebih dahulu pada basis data lokal SQLite. Setiap data diberi penanda status sinkronisasi, misalnya belum sinkron atau sudah sinkron, agar aplikasi dapat membedakan data yang masih berada dalam antrean lokal dan data yang telah berhasil dikirim ke server. Penggunaan basis data lokal dan API untuk memisahkan penyimpanan perangkat dan server juga diterapkan pada pengembangan aplikasi *mobile* yang membutuhkan proses sinkronisasi data [14].

Ketika koneksi internet tersedia, modul sinkronisasi mengirim data dari antrean lokal ke REST API *backend*. Pengiriman dilakukan secara bertahap, dimulai dari data utama, kemudian dilanjutkan dengan media atau data pendukung lainnya. Jika proses pengiriman gagal, data tidak dihapus dari penyimpanan lokal, tetapi tetap dipertahankan untuk dicoba kembali pada proses sinkronisasi berikutnya. Setelah data berhasil diterima oleh *backend*, status lokal diperbarui menjadi tersinkronisasi agar data tidak dikirim ulang.

Dengan mekanisme ini, GeoTrackOps tetap dapat digunakan untuk mencatat aktivitas lapangan pada kondisi jaringan terbatas. Alur sinkronisasi diterapkan secara bertahap, dimulai dari penyimpanan data ke SQLite ketika perangkat *offline*, pemberian status sinkronisasi pada setiap data, pengiriman data ke REST API *backend* ketika koneksi tersedia, percobaan ulang apabila terjadi kegagalan pengiriman, serta pembaruan status lokal setelah data berhasil diproses oleh server. Alur tersebut digunakan untuk menjaga konsistensi antara data pada perangkat *mobile* dan data pada server, sekaligus mengurangi risiko kehilangan data ketika koneksi jaringan tidak stabil.

2.6. Pengujian Black-Box

Pengujian *Black-Box Testing* dilakukan untuk mengevaluasi kesesuaian fungsi GeoTrackOps berdasarkan masukan pengguna dan keluaran yang terlihat pada antarmuka, tanpa meninjau struktur internal kode program. Metode ini digunakan karena pengujian difokuskan pada perilaku sistem dari sisi pengguna, seperti autentikasi, akses berbasis peran, alur kerja lapangan, layanan GPS/LBS, penyimpanan data saat *offline*, dan sinkronisasi data ketika perangkat kembali *online*. Pengujian *black-box* sesuai digunakan untuk memeriksa fungsionalitas aplikasi berdasarkan skenario uji dan keluaran yang diharapkan [10], [16].

Pada penelitian ini, pengujian dilakukan pada aplikasi *mobile* GeoTrackOps dan *dashboard web* sesuai peran pengguna. Field User diuji pada *alur mobile*, mulai dari *login*, pemilihan *route*, tampilan peta, pelaporan *stopsite*, pencatatan lokasi, penyimpanan *offline*, hingga sinkronisasi. Sementara itu, Admin dan Basecamp Operator diuji pada akses *dashboard* untuk memastikan fungsi *monitoring* dan pengelolaan data berjalan sesuai hak akses. Ringkasan kategori pengujian ditunjukkan pada Tabel 1.

Tabel 1. Kategori Pengujian Black-Box GeoTrackOps

No	Kategori Pengujian	Jumlah Skenario	Fokus Pengujian	Keluaran yang Diharapkan
1	Login dan autentikasi	3	Pengujian dilakukan pada kondisi kredensial valid, kredensial tidak valid, dan akun dengan status belum dapat digunakan.	Sistem hanya mengizinkan pengguna valid masuk ke aplikasi, menyimpan sesi pengguna yang berhasil masuk, serta menolak akses dan menampilkan pesan

No	Kategori Pengujian	Jumlah Skenario	Fokus Pengujian	Keluaran yang Diharapkan
2	Akses berbasis peran	2	Pengujian dilakukan untuk memastikan perbedaan hak akses antara Field User, Admin, dan Basecamp Operator pada aplikasi <i>mobile</i> dan <i>dashboard web</i> .	kesalahan pada kondisi akun yang tidak sesuai. Field User hanya dapat mengakses fitur lapangan seperti <i>route</i> , peta, dan <i>stopsite</i> , sedangkan Admin dan Basecamp Operator dapat mengakses fungsi pemantauan dan pengelolaan data melalui <i>dashboard web</i> .
3	Alur <i>route</i> dan <i>stopsite</i>	4	Pengujian mencakup pemuatan daftar <i>route</i> , proses memulai <i>traverse</i> , tampilan peta pada <i>route</i> aktif, serta pengisian dan penyimpanan <i>form stopsite</i> beserta validasinya.	<i>Route</i> dapat dimuat dan dijalankan, peta menampilkan aktivitas lapangan, serta data <i>stopsite</i> dapat tersimpan sesuai kondisi jaringan, baik langsung ke server saat <i>online</i> maupun ke antrean lokal saat <i>offline</i> . Saat <i>online</i> , titik lokasi berhasil dikirim ke <i>backend</i> ; saat <i>offline</i> , titik lokasi disimpan pada penyimpanan lokal dengan status belum tersinkronisasi untuk dikirim ulang ketika koneksi tersedia.
4	GPS/LBS tracking	2	Pengujian dilakukan pada kondisi <i>route</i> aktif dengan izin lokasi aktif, baik saat perangkat <i>online</i> maupun <i>offline</i> .	
5	Sinkronisasi <i>offline-online</i>	4	Proses sinkronisasi data: data <i>stopsite</i> , media, dan titik lokasi disimpan di SQLite saat <i>offline</i> ; saat koneksi tersedia atau sinkronisasi manual dijalankan, modul sinkronisasi mengirim data ke REST API; data yang berhasil diproses diperbarui statusnya menjadi tersinkron; data gagal tetap berada dalam antrean untuk <i>retry</i> .	Data tertunda tidak hilang saat <i>offline</i> , antrean lokal bertambah sesuai data yang belum terkirim, data berhasil dikirim ke server saat sinkronisasi berjalan, status lokal berubah menjadi tersinkronisasi, dan data gagal tetap dapat diproses ulang melalui <i>retry</i> .
Total	-	15	-	-

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengembangan dan pengujian aplikasi *mobile* GeoTrackOps setelah dilakukan penyempurnaan pada fitur utama sistem. Hasil yang dibahas meliputi implementasi aplikasi, pengujian layanan berbasis lokasi, pengujian sinkronisasi *offline-online*, serta ringkasan pengujian *black-box*. Pembahasan difokuskan pada kesesuaian hasil pengembangan terhadap kebutuhan operasional tim lapangan, terutama dalam pencatatan aktivitas berbasis lokasi, penyimpanan data saat *offline*, dan sinkronisasi data ke server ketika koneksi tersedia kembali.

3.1. Hasil Implementasi Aplikasi

Hasil implementasi menunjukkan bahwa GeoTrackOps telah menyediakan dua antarmuka utama, yaitu aplikasi *mobile* untuk Field User dan *dashboard web* untuk Admin serta Basecamp Operator. Aplikasi *mobile* digunakan untuk mendukung aktivitas lapangan, mulai dari *login*, pemilihan *route*, pemantauan lokasi melalui peta, pelaporan *stopsite*, hingga pengambilan dokumentasi lapangan. Pemanfaatan aplikasi *mobile* sebagai media pengumpulan data lapangan relevan karena perangkat *mobile* dapat mendukung pencatatan berbasis perangkat, penggunaan sensor, dan integrasi data ke sistem terpusat [2].

Pada alur Field User, aplikasi menampilkan daftar *route* yang dapat dipilih untuk memulai aktivitas. Setelah *route* aktif, pengguna diarahkan ke halaman peta untuk mendukung pemantauan posisi dan pelaporan *stopsite*. *Form stopsite* menyediakan input data kerja, pengambilan koordinat lokasi, dan unggah dokumentasi. Apabila koneksi internet tidak tersedia, data disimpan terlebih dahulu pada penyimpanan lokal dan diproses kembali melalui mekanisme sinkronisasi saat koneksi tersedia. Pola pemisahan data lokal dan server melalui API juga digunakan pada pengembangan aplikasi *mobile* yang membutuhkan sinkronisasi data [14].

Sementara itu, Admin dan Basecamp Operator menggunakan *dashboard web* untuk memantau progres aktivitas lapangan dan mengelola data operasional. Perbedaan antarmuka berdasarkan peran pengguna membuat fungsi sistem lebih sesuai dengan kebutuhan masing-masing pengguna. Tampilan hasil implementasi GeoTrackOps ditunjukkan pada Gambar 3.

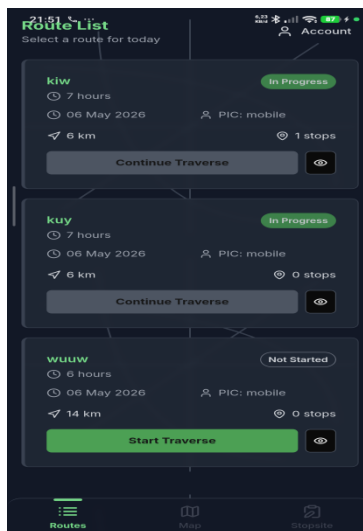
Gambar 3 menunjukkan tampilan utama GeoTrackOps berdasarkan alur penggunaan. Field User menggunakan aplikasi *mobile* untuk *login*, memilih *route*, memantau posisi pada peta, dan mencatat *stopsite*, sedangkan Admin dan Basecamp Operator menggunakan *dashboard web* untuk pemantauan dan pengelolaan data operasional.

3.2. Hasil Pengujian Location-Based Service

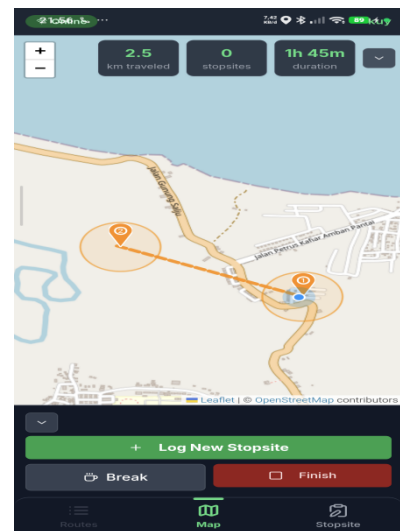
Pengujian *Location-Based Service* dilakukan untuk memastikan bahwa GeoTrackOps mampu membaca koordinat pengguna, mengirim titik lokasi saat perangkat *online*, menyimpan titik lokasi saat *offline*, dan mengisi koordinat pada *form stopsite*. Pengujian ini penting karena data lokasi digunakan untuk mendukung keterlacakan aktivitas lapangan. Pada aplikasi berbasis lokasi, kualitas pembacaan posisi dapat dipengaruhi oleh karakteristik GNSS *smartphone*, kondisi lingkungan, dan latensi pengiriman data lokasi [4], [5], [6].



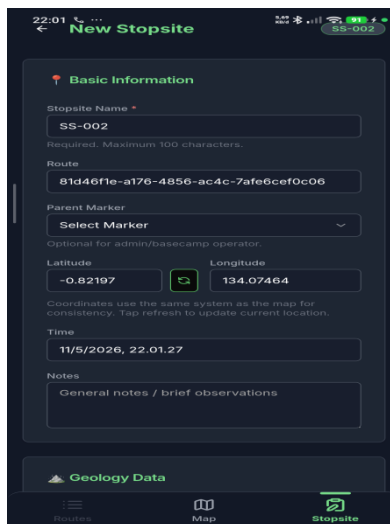
(a) Login



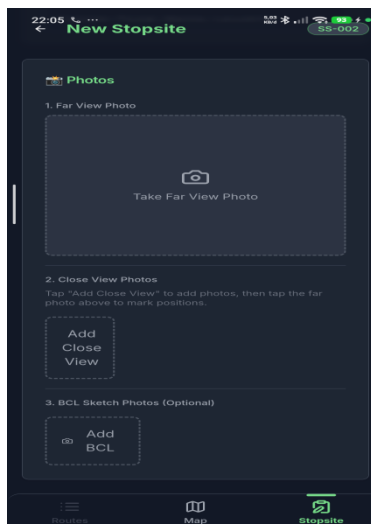
(b) Daftar route



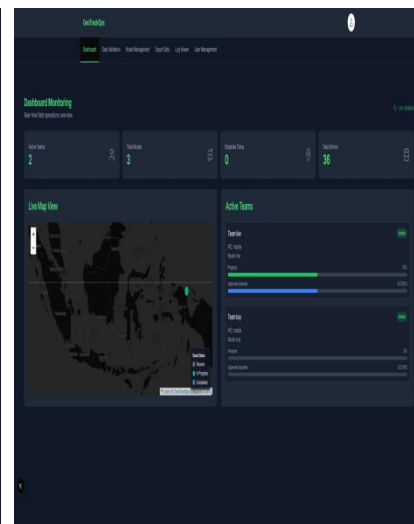
(c) Peta traverse



(d) Form informasi dasar stopsite



(e) Form dokumentasi stopsite



(f) Dashboard web

Gambar 3. Tampilan Antarmuka GeoTrackOps

Tabel 2. Hasil Pengujian Location-Based Service

No	Skenario Pengujian	Hasil Pengujian	Status
1	Izin lokasi diaktifkan dan aplikasi membaca posisi perangkat	Koordinat latitude dan longitude berhasil terbaca pada aplikasi	Lulus
2	Route aktif dan perangkat berada dalam kondisi <i>online</i>	Titik lokasi berhasil dikirim dan diterima oleh <i>backend</i>	Lulus
3	Route aktif dan perangkat berada dalam kondisi <i>offline</i>	Titik lokasi berhasil disimpan pada antrean lokal	Lulus

No	Skenario Pengujian	Hasil Pengujian	Status
4	Field User mengambil koordinat pada <i>form stopsite</i>	Koordinat berhasil mengisi <i>form stopsite</i> sesuai pembacaan perangkat	Lulus

Berdasarkan Tabel 2, fitur LBS pada GeoTrackOps telah berjalan sesuai skenario pengujian. Aplikasi mampu membaca koordinat perangkat, mengirimkan data lokasi saat koneksi tersedia, serta menyimpan titik lokasi secara lokal saat perangkat *offline*. Hasil ini menunjukkan bahwa fitur lokasi telah terintegrasi dengan aktivitas operasional seperti *route tracking* dan pelaporan *stopsite*.

3.3. Hasil Pengujian Sinkronisasi Offline-Online

Pengujian sinkronisasi *offline-online* dilakukan untuk mengetahui perubahan kondisi data sebelum dan sesudah proses sinkronisasi. Pengujian difokuskan pada data *stopsite*, media dokumentasi, titik lokasi *tracking*, serta penanganan data yang gagal dikirim. Perbandingan ini digunakan untuk memastikan bahwa data yang dibuat saat perangkat *offline* tetap tersimpan di SQLite lokal sebagai antrean sinkronisasi, lalu dapat dikirim ke *backend* ketika koneksi tersedia kembali. Strategi mode *offline* dan sinkronisasi data pada aplikasi *mobile* diperlukan untuk menjaga reliabilitas sistem dalam kondisi jaringan yang buruk atau terputus-putus [3], [15]. Selain itu, penggunaan penyimpanan lokal dan API sebagai penghubung ke server relevan pada aplikasi *mobile* yang memisahkan data pada perangkat dan data pada server [14]. Perbandingan kondisi data sebelum dan sesudah sinkronisasi *offline-online* ditunjukkan pada Tabel 3

Tabel 3. Perbandingan Data Sebelum dan Sesudah Sinkronisasi

No	Objek Pengujian	Sebelum Sinkronisasi	Proses Sinkronisasi	Sesudah Sinkronisasi
1	Data <i>stopsite</i>	Field User menyimpan data <i>stopsite</i> saat perangkat <i>offline</i> . Data masuk ke SQLite lokal dengan status belum tersinkronisasi.	Saat koneksi tersedia, modul sinkronisasi mengirim data <i>stopsite</i> dari antrean lokal ke REST API <i>backend</i> .	Data <i>stopsite</i> berhasil diterima server, status lokal berubah menjadi tersinkronisasi, dan antrean lokal berkurang.
2	Media dokumentasi	Referensi media/foto tersimpan pada perangkat dan belum diunggah ke server.	Setelah data utama <i>stopsite</i> diproses, sistem mengunggah media terkait secara bertahap ke <i>backend</i> .	Media berhasil terhubung dengan data <i>stopsite</i> di server dan tidak lagi berstatus tertunda.
3	Titik lokasi <i>tracking</i>	Titik lokasi yang direkam saat <i>offline</i> tersimpan pada penyimpanan lokal dengan status belum terkirim.	Ketika perangkat kembali <i>online</i> , titik lokasi tertunda dikirim ke <i>backend</i> melalui proses sinkronisasi.	Titik lokasi berhasil tersimpan di server dan status lokal diperbarui menjadi tersinkronisasi.
4	Kegagalan dan <i>retry</i>	Sebagian data tetap berada dalam antrean lokal ketika pengiriman gagal karena gangguan koneksi atau respons server tidak berhasil.	Sistem mempertahankan data gagal dan memproses ulang pada sinkronisasi berikutnya atau melalui sinkronisasi manual.	Data tidak hilang, dapat dikirim ulang, dan status diperbarui setelah sinkronisasi berhasil.

Berdasarkan Tabel 3, proses sinkronisasi GeoTrackOps menunjukkan perubahan kondisi data dari status tertunda pada penyimpanan lokal menjadi tersinkronisasi setelah data berhasil diproses oleh server. Sebelum sinkronisasi, data *stopsite*, media dokumentasi, dan titik lokasi *tracking* masih berada pada SQLite lokal dengan status belum tersinkronisasi. Setelah koneksi tersedia dan modul sinkronisasi berjalan, data dikirim ke *backend*, status lokal diperbarui, dan data dapat digunakan pada sisi server maupun *dashboard*. Jika proses pengiriman gagal, data tidak dihapus dari antrean lokal sehingga masih dapat diproses ulang pada siklus sinkronisasi berikutnya.

3.4. Hasil Pengujian Performa Teknis

Pengujian performa teknis dilakukan untuk memperoleh data kuantitatif mengenai waktu sinkronisasi, latensi pembaruan lokasi pada *dashboard*, tingkat keberhasilan pengiriman data, dan akurasi GPS/LBS. Pengujian dilakukan menggunakan 2 perangkat *Android* pada rute uji sepanjang 3,2 km dengan kondisi jaringan 4G stabil setelah koneksi kembali tersedia. Ringkasan hasil pengujian performa teknis ditunjukkan pada Tabel 4.

Berdasarkan Tabel 4 dan Gambar 4, waktu sinkronisasi rata-rata dari data yang tersimpan saat *offline* hingga berhasil terkirim ke server adalah 12,4 detik, dengan waktu minimum 4,1 detik dan maksimum 38,7 detik. Hasil ini menunjukkan bahwa mekanisme sinkronisasi *offline-online* pada GeoTrackOps mampu mengirimkan kembali data yang tertunda setelah koneksi tersedia. Perbedaan waktu sinkronisasi dapat dipengaruhi oleh jumlah antrean data, ukuran media yang dikirimkan, serta kondisi jaringan saat proses

pengiriman berlangsung. Strategi penyimpanan lokal dan sinkronisasi ulang diperlukan pada aplikasi *mobile* agar data tetap dapat diproses pada kondisi jaringan yang tidak stabil [3], [15].

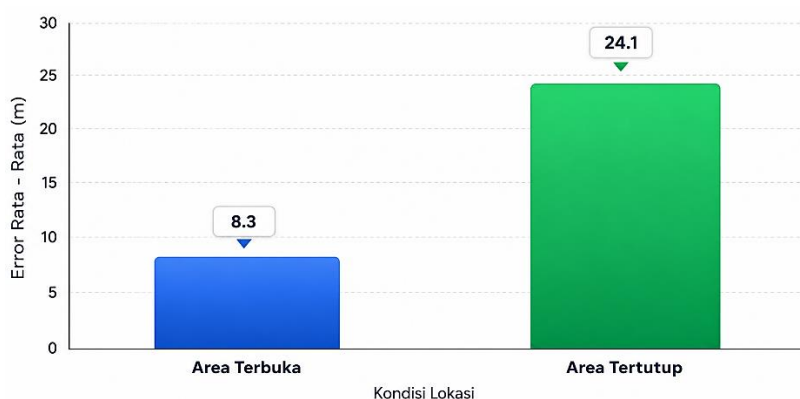
Latensi tampilan posisi pada *dashboard web* berada pada rentang 5,2–9,8 detik, dengan rata-rata 6,7 detik dan standar deviasi 1,3 detik. Nilai tersebut masih sesuai dengan mekanisme sistem karena aplikasi *mobile* mengirimkan data lokasi setiap 3 detik dan *dashboard web* melakukan *polling* setiap 5 detik. Pada aplikasi berbasis posisi *real-time*, latensi proses akuisisi, pengiriman, dan pembaruan data lokasi dapat memengaruhi kecepatan tampilan posisi pengguna [6].

Tingkat keberhasilan pengiriman data mencapai 100%, yang menunjukkan bahwa seluruh antrean data *offline* berhasil tersinkronisasi ke server setelah koneksi tersedia kembali. Hasil ini menunjukkan bahwa mekanisme penyimpanan lokal, antrean sinkronisasi, pengiriman ke *backend*, dan pembaruan status data berjalan sesuai rancangan. Penggunaan basis data lokal yang dipadukan dengan API server relevan untuk aplikasi *mobile* yang membutuhkan pemisahan data perangkat dan data server serta proses sinkronisasi ketika koneksi tersedia [14].

Gambar 4 menunjukkan bahwa error rata-rata GPS/LBS pada area terbuka sebesar 8,3 meter, sedangkan pada area tertutup meningkat menjadi 24,1 meter. Perbedaan ini menunjukkan bahwa akurasi pembacaan lokasi pada perangkat *mobile* dipengaruhi oleh kondisi lingkungan, hambatan fisik, kualitas sinyal, dan karakteristik pengukuran GNSS pada *smartphone* [5]. Dengan demikian, hasil pengujian performa teknis melengkapi pengujian fungsional *Black-Box* dengan bukti kuantitatif mengenai kinerja GeoTrackOps dalam mendukung pelaporan lapangan berbasis lokasi.

Tabel 4. Ringkasan Pengujian Performa GeoTrackOps

Parameter Pengujian	Kondisi Pengujian	Hasil Pengujian	Keterangan
Waktu sinkronisasi	2 perangkat <i>Android</i> , rute 3,2 km, jaringan 4G stabil setelah koneksi kembali tersedia	Rata-rata 12,4 s; minimum 4,1 s; maksimum 38,7 s	Data <i>offline</i> berhasil dikirim kembali ke server setelah koneksi tersedia
Latensi posisi <i>dashboard</i>	Interval pengiriman <i>mobile</i> 3 s dan polling <i>dashboard web</i> 5 s	Rata-rata 6,7 s; rentang 5,2–9,8 s; SD 1,3 s	Waktu yang dibutuhkan hingga posisi pengguna tampil pada <i>dashboard web</i>
Tingkat keberhasilan pengiriman data	Sinkronisasi data <i>offline</i> ke server	100%	Seluruh antrean data <i>offline</i> berhasil tersinkronisasi ke server
Akurasi GPS/LBS area terbuka	Pengujian lokasi pada area terbuka	Error rata-rata 8,3 m; maksimum 18,2 m	Akurasi relatif lebih baik karena penerimaan sinyal lebih stabil
Akurasi GPS/LBS area tertutup	Pengujian lokasi pada area tertutup	Error rata-rata 24,1 m	Akurasi menurun karena hambatan lingkungan terhadap sinyal GPS/GNSS



Gambar 4. Perbandingan Error Rata-Rata GPS/LBS pada Area Terbuka dan Tertutup

3.5. Hasil Pengujian Black-Box

Pengujian *black-box* dilakukan untuk melihat kesesuaian fungsi aplikasi terhadap keluaran yang diharapkan tanpa meninjau struktur internal kode program. Pengujian ini sesuai untuk mengevaluasi perilaku aplikasi dari sisi pengguna karena fokusnya berada pada *input*, proses yang terlihat, dan *output* sistem [10], [16]. Pada Bab 2, kategori skenario uji telah disajikan pada Tabel 1. Oleh karena itu, pada bagian ini hasil pengujian diringkas berdasarkan kategori agar tidak mengulang tabel yang sama.

Tabel 5. Ringkasan Hasil Pengujian Black-Box

Kategori Pengujian	Jumlah Skenario	Lulus	Gagal	Tingkat Keberhasilan (%)
<i>Login</i> dan autentikasi	3	3	0	100%
Akses berbasis peran dan <i>dashboard</i>	2	2	0	100%
Alur <i>route</i> dan <i>stopsite</i>	4	4	0	100%
Pelacakan GPS/LBS	2	2	0	100%
Sinkronisasi <i>offline-online</i>	4	4	0	100%
Total	15	15	0	100%

Berdasarkan Tabel 5, seluruh skenario pengujian memperoleh status lulus. Hasil ini menunjukkan bahwa fungsi utama GeoTrackOps telah berjalan sesuai keluaran yang diharapkan pada skenario pengujian yang dilakukan. Fitur *login*, akses berbasis peran, alur *route* dan *stopsite*, pelacakan lokasi, penyimpanan *offline*, serta sinkronisasi data telah memenuhi kebutuhan fungsional yang ditentukan pada tahap analisis kebutuhan.

3.6. Pembahasan

Hasil implementasi dan pengujian menunjukkan bahwa GeoTrackOps mampu mendukung alur kerja operasional tim lapangan melalui integrasi aplikasi *mobile*, layanan lokasi, penyimpanan lokal, sinkronisasi *offline-online*, dan *dashboard web* berbasis peran. Field User dapat melakukan pencatatan *route*, *stopsite*, dokumentasi, dan pelacakan lokasi melalui aplikasi *mobile*, sedangkan Admin dan Basecamp Operator dapat memantau aktivitas melalui *dashboard web*. Hasil ini sejalan dengan Harris et al. [2] yang menunjukkan bahwa aplikasi *mobile* dapat mendukung pengumpulan data berbasis perangkat dan integrasi data ke sistem terpusat. Namun, GeoTrackOps memiliki perbedaan karena tidak hanya berfokus pada pengumpulan data, tetapi juga menggabungkan alur *route-stopsite*, GPS/LBS tracking, penyimpanan lokal SQLite, sinkronisasi *offline-online*, dan *dashboard* berbasis peran dalam satu sistem operasional lapangan.

Pada fitur *Location-Based Service*, aplikasi berhasil membaca koordinat perangkat, mengirimkan data lokasi saat *online*, serta menyimpan titik lokasi saat *offline*. Hal ini menunjukkan bahwa data lokasi tidak hanya berfungsi sebagai informasi posisi, tetapi juga menjadi bagian dari konteks pelaporan aktivitas lapangan. Temuan ini mendukung kajian Kung et al. [4] yang menjelaskan bahwa *Location-Based Service* dapat digunakan untuk kebutuhan navigasi, pemantauan, dan layanan berbasis posisi pengguna. Meskipun demikian, performa GPS/LBS tetap dipengaruhi oleh kondisi lingkungan. Hasil pengujian menunjukkan error rata-rata sebesar 8,3 meter pada area terbuka dan meningkat menjadi 24,1 meter pada area tertutup. Perbedaan ini menunjukkan bahwa hambatan fisik, kualitas sinyal, dan karakteristik GNSS pada *smartphone* berpengaruh terhadap akurasi lokasi, sebagaimana dijelaskan oleh Suzuki [5].

Dari sisi pembaruan lokasi, latensi posisi *dashboard* berada pada rata-rata 6,7 detik dengan rentang 5,2–9,8 detik. Nilai ini dipengaruhi oleh interval pengiriman lokasi dari aplikasi *mobile*, proses komunikasi ke server, dan polling pada *dashboard web*. Kondisi tersebut sejalan dengan Hernández Olcina et al. [6] yang menjelaskan bahwa latensi pada aplikasi GNSS real-time berbasis *mobile* dapat memengaruhi kecepatan pembaruan posisi. Dengan demikian, hasil pengujian menunjukkan bahwa GeoTrackOps telah mampu menampilkan posisi pengguna secara relatif konsisten, meskipun kecepatan pembaruan tetap bergantung pada mekanisme pengiriman data dan kualitas jaringan.

Pada mekanisme sinkronisasi *offline-online*, data yang dibuat saat perangkat *offline* berhasil disimpan dalam antrian lokal dan dikirimkan kembali ketika koneksi tersedia. Waktu sinkronisasi rata-rata sebesar 12,4 detik dan tingkat keberhasilan pengiriman data sebesar 100% menunjukkan bahwa mekanisme penyimpanan lokal, antrian sinkronisasi, pengiriman ke *backend*, dan pembaruan status data telah berjalan sesuai rancangan. Hasil ini sejalan dengan Wang dan Wang [3] serta Pothineni [15] yang menekankan pentingnya mode *offline* dan sinkronisasi data untuk menjaga reliabilitas aplikasi *mobile* pada kondisi jaringan tidak stabil. Penggunaan SQLite dan REST API pada GeoTrackOps juga mendukung pola pemisahan data lokal dan server sebagaimana diterapkan pada aplikasi *mobile* yang membutuhkan sinkronisasi data [14].

Secara praktis, GeoTrackOps dapat membantu tim lapangan mengurangi risiko kehilangan data, keterlambatan pelaporan, dan ketidaksesuaian antara data pada perangkat dan server. Sistem ini juga memberi dukungan monitoring bagi Admin dan Basecamp Operator melalui *dashboard web*. Namun, penelitian ini masih memiliki keterbatasan karena pengujian dilakukan pada jumlah perangkat dan rute uji yang terbatas, sehingga belum mewakili seluruh variasi perangkat, versi sistem operasi, kondisi geografis, kualitas jaringan, dan gangguan koneksi ekstrem. Oleh karena itu, pengembangan berikutnya perlu diarahkan pada pengujian dengan perangkat dan kondisi jaringan yang lebih beragam, optimasi algoritma sinkronisasi, serta evaluasi akurasi lokasi pada lingkungan operasional lapangan yang lebih luas.

4. KESIMPULAN

Penelitian ini menghasilkan pengembangan lanjutan aplikasi *mobile* GeoTrackOps berbasis lokasi dengan dukungan sinkronisasi *offline-online* untuk mendukung operasional tim lapangan. Pengembangan dilakukan menggunakan pendekatan *Research and Development* dan *Agile Scrum* dengan fokus pada penyempurnaan *manajemen route*, pelaporan *stopsite*, GPS/LBS tracking, penyimpanan lokal berbasis SQLite, sinkronisasi data ke *backend* server, serta *dashboard web* untuk Admin dan Basecamp Operator.

Hasil pengujian menunjukkan bahwa seluruh 15 skenario *Black-Box* memperoleh status lulus, mencakup *login*, akses berbasis peran, alur *route* dan *stopsite*, pelacakan GPS/LBS, penyimpanan *offline*, sinkronisasi data, dan akses *dashboard*. Pengujian performa teknis menunjukkan waktu sinkronisasi rata-rata 12,4 detik, latensi posisi *dashboard* rata-rata 6,7 detik, tingkat keberhasilan pengiriman data 100%, serta error rata-rata GPS/LBS sebesar 8,3 meter pada area terbuka dan 24,1 meter pada area tertutup. Hasil tersebut menunjukkan bahwa GeoTrackOps dapat mendukung pencatatan, pelaporan, dan pemantauan aktivitas lapangan secara lebih andal, terutama pada kondisi jaringan terbatas.

Kontribusi utama penelitian ini terletak pada integrasi layanan lokasi, alur *route-stopsite*, penyimpanan lokal SQLite, sinkronisasi *offline-online*, dan *dashboard* berbasis peran dalam satu sistem operasional lapangan. Kelebihan aplikasi ini adalah tetap dapat digunakan ketika koneksi internet terbatas dan mampu menyinkronkan kembali data ke server saat jaringan tersedia. Namun, penelitian ini masih terbatas pada jumlah perangkat dan rute uji yang terbatas, sehingga belum sepenuhnya mewakili variasi perangkat, versi sistem operasi, kondisi geografis, kualitas sinyal GNSS, dan gangguan jaringan ekstrem. Penelitian selanjutnya disarankan untuk melakukan pengujian pada lingkungan operasional yang lebih luas, mengoptimalkan algoritma sinkronisasi, serta mengevaluasi akurasi lokasi dan stabilitas pengiriman data pada kondisi jaringan yang lebih beragam.

REFERENSI

- [1] S. Angelopoulos, E. Bendoly, J. C. Fransoo, K. Hoberg, C. Ou, and A. Tenhiälä, "Digital transformation in operations management: Fundamental change through agency reversal," *J. Oper. Manag.*, vol. 69, no. 6, pp. 876–889, 2023, doi: 10.1002/joom.1271.
- [2] P. A. Harris *et al.*, "MyCap: a flexible and configurable platform for mobilizing the participant voice," *JAMIA Open*, vol. 5, no. 2, p. ooac047, Jul. 2022, doi: 10.1093/jamiaopen/ooac047.
- [3] F. Wang and Y. Wang, "Optimizing Offline Mode and Data Synchronization Techniques for Literature Translation Applications on Mobile Devices," *Int. J. Interact. Mob. Technol.*, vol. 18, no. 22, pp. 115–129, Nov. 2024, doi: 10.3991/IJIM.V18I22.52451.
- [4] H.-Y. Kung, X.-G. Yue, C.-H. Chen, and F.-J. Hwang, "Architecture, Technologies, and Applications of Location-Based Services," *Mob. Inf. Syst.*, vol. 2023, p. 9796535, 2023, doi: 10.1155/2023/9796535.
- [5] T. Suzuki, "Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization," *Sensors*, vol. 23, no. 3, p. 1205, 2023, doi: 10.3390/s23031205.
- [6] J. Hernández Olcina, A. B. Anquela Julián, and Á. E. Martín Furones, "Navigating latency hurdles: an in-depth examination of a cloud-powered GNSS real-time positioning application on mobile devices," *Sci. Rep.*, vol. 14, no. 1, p. 14668, 2024, doi: 10.1038/s41598-024-65652-7.
- [7] M. Waruwu, "Metode Penelitian dan Pengembangan (R&D): Konsep, Jenis, Tahapan dan Kelebihan," *J. Ilm. Profesi Pendidik.*, vol. 9, no. 2, pp. 1220–1230, May 2024, doi: 10.29303/JIPP.V9I2.2141.
- [8] A. Alami and O. Krancher, "How Scrum adds value to achieving software quality?," *Empir. Softw. Eng.*, vol. 27, no. 7, p. 165, Dec. 2022, doi: 10.1007/S10664-022-10208-4.
- [9] W.-T. Lee and C.-H. Chen, "Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering," *Electron.*, vol. 12, no. 15, p. 3291, 2023, doi: 10.3390/electronics12153291.
- [10] A. D. Frayudha, I. R. Pande, and M. B. Juwita, "Implementation of Black Box Testing with the Application of Equivalence Partitioning Techniques in the M-Magazine Android Application at Semen Gresik High School," *Elinvo Electron. Informatics, Vocat. Educ.*, vol. 9, no. 1, pp. 134–143, 2024, doi: 10.21831/elinvo.v9i1.70382.
- [11] B. A. S. Muntai, E. Darwiyanto, and R. Nurtantyana, "The Development of a Mobile Application Using Scrum Pattern for Monitoring Project Information System," *Indones. J. Comput.*, vol. 9, no. 2, pp. 134–153, 2024, doi: 10.34818/INDOJC.2024.9.2.970.
- [12] Y. J. Pérez Castillo, S. D. Orantes Jiménez, and P. O. Letelier Torres, "Sprint Management in Agile Approach: Progress and Velocity Evaluation Applying Machine Learning," *Inf.*, vol. 15, no. 11, p. 726, 2024, doi: 10.3390/info15110726.
- [13] A. Przewięźlikowska, W. Ślusarczyk, K. Wójcik, and M. Ślusarski, "Efficient and scalable architecture for location-based mobile applications using metrika," *Autom. Constr.*, vol. 172, p. 106056, Apr. 2025, doi: 10.1016/J.AUTCON.2025.106056.

- [14] H. Kaur *et al.*, “Unified Mobile App for Streamlining Verbal Autopsy and Cause of Death Assignment in India: Design and Development Study,” *JMIR Form. Res.*, vol. 9, p. e59937, 2025, doi: 10.2196/59937.
- [15] S. H. Pothineni, “Offline-First Mobile Architecture: Enhancing Usability and Resilience in Mobile Systems,” *J. Artif. Intell. Gen. Sci.*, vol. 7, no. 1, pp. 320–326, 2024, doi: 10.60087/jaigs.v7i01.387.
- [16] A. Tyanafisya *et al.*, “Black Box Testing Using Equivalence Partitioning Technique on Bakkar Website,” *J. INSTEK (Informatika Sains dan Teknol.*, vol. 10, no. 1, pp. 230–238, May 2025, doi: 10.24252/instek.v10i1.52951.