

Perancangan Sistem Monitoring Operasional Alat Berat di Pertambangan PT. Pamapersada Nusantara berbasis Android

Design of an Android-based Heavy Equipment Operational Monitoring System for Mining Operations at PT Pamapersada Nusantara

¹Aldi Muhammad Taufiq*, ²Siti Hertina, ³Hendra Gunawan

^{1,3}Program Studi Teknik Informatika, STMIK IM

²Program Studi Sistem Informasi, STMIK IM

^{1,2,3}Jl. Belitung No. 7 Bandung 40113 Jawa Barat – Indonesia

*e-mail: Aldipun39@gmail.com

(received: 3 May 2025, revised: 19 May 2025, accepted: 19 May 2025)

Abstrak

PT Pamapersada Nusantara (PAMA) sebagai perusahaan di bidang pertambangan menghadapi tantangan dalam pemantauan kinerja alat berat yang masih dilakukan secara manual, sehingga rentan terhadap kesalahan pencatatan dan mengganggu kelancaran proses produksi. Permasalahan yang dihadapi meliputi ketidaktepatan pencatatan aktivitas alat berat *Shovel* dan *Haul Truck*, ketidaksesuaian status operator selama proses produksi, serta tidak tersedianya informasi lokasi operasional secara *real-time*. Penelitian ini bertujuan untuk meningkatkan efisiensi dalam identifikasi status alat berat, seperti *standby*, *delay*, dan *down*, melalui pelaporan langsung oleh operator. Metode pengembangan yang digunakan adalah *Agile Scrum*, dengan pembagian proses kerja ke dalam sprint untuk memungkinkan evaluasi dan perbaikan berkelanjutan berdasarkan kebutuhan pengguna. Hasil dari penelitian ini adalah aplikasi berbasis Android bernama *MineTrack*, yang mampu memantau aktivitas operasional alat berat secara *real-time* dengan memanfaatkan teknologi GPS untuk pelacakan lokasi serta *SQLite* untuk pengelolaan data. Kesimpulannya, aplikasi *MineTrack* meningkatkan akurasi pemantauan operasional alat berat, mendukung pengambilan keputusan yang lebih efektif dalam manajemen alat berat di industri pertambangan, serta berkontribusi pada peningkatan produktivitas dan pengurangan risiko kesalahan operasional.

Kata kunci: sistem monitoring, alat berat, android, GPS, *agile scrum*

Abstract

PT Pamapersada Nusantara (PAMA), a company operating in the mining sector, faces challenges in monitoring the performance of heavy equipment, which is still conducted manually. This method is prone to recording errors and disrupts the smooth flow of production processes. The issues include inaccurate logging of activities for Shovel and Haul Truck units, discrepancies in operator status during production, and the lack of real-time operational location information. This study aims to improve efficiency in identifying the status of heavy equipment—such as standby, delay, and down—through direct reporting by operators. The development methodology employed is Agile Scrum, with the work process divided into sprints to allow for continuous evaluation and refinement based on user needs. The outcome of this study is an Android-based application called MineTrack, which enables real-time monitoring of heavy equipment operations by utilizing GPS technology for location tracking and SQLite for data management. In conclusion, MineTrack enhances the accuracy of operational monitoring, supports more effective decision-making in heavy equipment management within the mining industry, and contributes to increased productivity while reducing the risk of operational errors.

Keywords: monitoring system, heavy equipment, android, GPS, *agile scrum*

1 Pendahuluan

Industri pertambangan merupakan sektor dengan intensitas penggunaan alat berat yang tinggi, seperti *shovel* dan *haul truck*, dalam kegiatan penggalian, pengangkutan, dan penimbunan material.

<http://sistemasi.ftik.unisi.ac.id>

Operasional di area tambang terbuka kerap menghadapi tantangan berat, seperti medan sulit, kondisi cuaca ekstrim, serta resiko kelongsoran akibat tanah timbunan yang tidak padat dan curah hujan tinggi [1]. Keandalan alat berat menjadi faktor utama dalam menjaga kelangsungan dan produktivitas proses produksi di sektor ini.

Saat ini, aktivitas monitoring alat berat di banyak area pertambangan masih mengandalkan pencatatan manual atau inspeksi visual. Pendekatan ini dinilai kurang efisien karena memerlukan waktu yang lama[2] serta sering gagal mendeteksi permasalahan pada alat secara dini, sehingga berpotensi meningkatkan *downtime* dan menurunkan produktivitas [3]

Penelitian sebelumnya menunjukkan bahwa efektivitas penggunaan alat berat dapat ditingkatkan dengan penerapan sistem monitoring kondisi secara *real-time*[4]. Selain itu, penerapan sistem monitoring terpusat berbasis digital di sektor pertambangan terbukti mampu meningkatkan efektivitas pengawasan, mempercepat pengambilan keputusan operasional, serta meningkatkan produktivitas alat berat[5].

PT. *United Tractors* memiliki anak perusahaan salah satunya adalah PT Pampersada Nusantara yang beroperasi di bidang kontraktor pertambangan batubara sejak tahun 1993, dengan salah satu area operasional berlokasi di Bontang, Kalimantan Timur, yang mengoperasikan alat berat secara terus-menerus selama 24 jam [6]. PAMA menghadapi tantangan dalam hal ketidaktepatan pencatatan aktivitas alat berat *shovel* dan *haul truck*, ketidaksesuaian status operator selama proses produksi, serta tidak tersedianya informasi lokasi alat berat secara *real-time*, yang berpotensi menyebabkan penurunan produktivitas, peningkatan risiko *downtime*, dan kerugian operasional.

Urgensi penelitian ini terletak pada kebutuhan untuk menggantikan sistem monitoring manual dengan sistem digital berbasis Android, yang mampu menyediakan data operasional secara cepat, akurat, dan *real-time*. Sistem ini akan didukung oleh infrastruktur jaringan *Local Area Network* (LAN) untuk menghubungkan perangkat-perangkat operasional [7]. serta memanfaatkan teknologi GPS[8] untuk pelacakan lokasi dan *SQLite* [9] untuk pengelolaan data aktivitas alat berat.

Penelitian ini bertujuan merancang aplikasi berbasis Android yang mengintegrasikan teknologi GPS, *SQLite*, dan LAN untuk mendukung pemantauan aktivitas alat berat secara *real-time*. Manfaat dari penelitian ini adalah meningkatkan akurasi pencatatan aktivitas alat berat, mempercepat pengambilan keputusan terkait operasional, mengoptimalkan produktivitas alat berat, serta mendukung pengelolaan operasional pertambangan secara lebih efektif dan efisien.

Ruang lingkup penelitian ini dibatasi pada perancangan fitur dasar aplikasi. Fitur-fitur lanjutan seperti integrasi dengan *cloud storage*, maupun analisis prediktif tidak termasuk dalam cakupan penelitian ini. Fokus utama perancangan aplikasi adalah pada kemampuan monitoring *real-time* aktivitas alat berat, penyimpanan data secara lokal menggunakan *SQLite*, serta peningkatan efisiensi dalam pelaporan harian aktivitas operasional di area pertambangan.

2 Tinjauan Literatur

Mengembangkan sistem monitoring *real-time* untuk Gas Jack *Compressor* di ladang gas dengan integrasi teknologi PLC, SCADA, dan MQTT [10] Sistem ini memungkinkan pemantauan suhu, tekanan, level, serta RPM alat berat secara digital melalui jaringan lokal sehingga mendukung percepatan pengambilan keputusan operasional. Meskipun pendekatan ini efektif dalam monitoring mesin industri berbasis *fixed location*, fokus utamanya belum menyentuh monitoring berbasis *mobile* Android di area pertambangan.

Merancang aplikasi presensi karyawan berbasis Android dan sensor GPS dengan metode *Agile Scrum* [11]. Aplikasi ini mendeteksi kehadiran karyawan secara *real-time* berdasarkan lokasi GPS untuk mencegah manipulasi absensi. Meskipun relevan dalam pemanfaatan GPS dan *mobile application*, ruang lingkungannya terbatas pada presensi personal, bukan monitoring aktivitas alat berat di medan pertambangan.

Mengembangkan sistem pelacakan kendaraan Trans Metro Bandung berbasis Android dengan memanfaatkan GPS dan *Firebase Realtime Database*[12]. Sistem ini memungkinkan pengguna memantau posisi kendaraan secara *real-time* untuk mendukung layanan transportasi publik. Namun, penerapan untuk alat berat dalam kondisi medan ekstrim seperti area tambang belum dikaji.

Mengembangkan sistem monitoring posisi komponen alat berat berbasis GPS dan IoT di PT Thiess Batakan [13]. Monitoring ini berbasis *web* dan difokuskan pada pelacakan pergerakan

komponen alat berat, bukan pada pencatatan status operasional (*standby, delay, down*) secara *real-time* berbasis Android.

Merancang sistem monitoring kondisi baterai alat berat menggunakan IoT dan aplikasi Android berbasis *Firebase* [14]. Monitoring ini berfokus pada parameter teknis baterai seperti suhu, arus, dan tegangan, tanpa memperhatikan status keseluruhan operasional alat berat.

Tabel 1. Analisis studi literatur terkait

Studi	Teknologi Yg digunakan	Fokus Studi	Temuan Kunci
[10]	PLC, SCADA, MQTT, LAN	Monitoring Gas Compressor	Monitoring <i>real-time</i> berbasis jaringan lokal untuk mendukung pengambilan keputusan.
[11]	Android, GPS, Agile Scrum	Presensi Karyawan Mobile	Validasi kehadiran berbasis GPS untuk keakuratan presensi <i>mobile</i> .
[12]	Android, GPS, Firebase Realtime DB	Monitoring Transportasi Umum	Validasi kehadiran berbasis GPS untuk keakuratan presensi <i>mobile</i> .
[13]	GPS, IoT, Web Platform	Monitoring Komponen Alat Berat	Pelacakan posisi alat berat untuk efisiensi operasional tambang.
[14]	Android, GPS Tracker	Rental Kendaraan	Pelacakan kendaraan untuk keamanan dan pengelolaan armada rental.

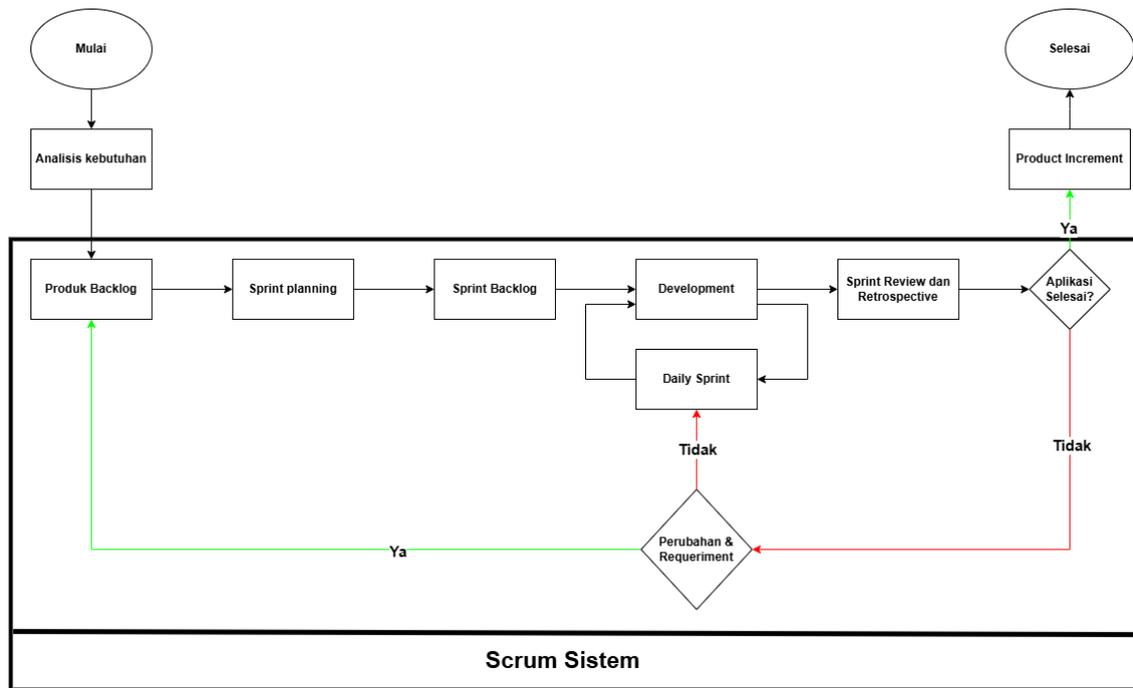
Berdasarkan Tabel 1 literatur yang telah dikaji, mayoritas penelitian sebelumnya telah memanfaatkan teknologi GPS, IoT, SCADA, dan aplikasi *mobile* untuk tujuan monitoring, baik terhadap kendaraan, presensi personal, maupun parameter teknis alat. Namun, dari semua jurnal yang telah ditemukan belum ada pembahasan khusus perancangan sistem monitoring status operasional alat berat di area pertambangan berbasis Android dengan integrasi GPS, *SQLite* untuk penyimpanan lokal, serta jaringan LAN untuk pengiriman data secara *real-time*. Oleh karena itu, penelitian ini memiliki *novelty* berupa:

1. Pengembangan aplikasi berbasis Android untuk monitoring status (*standby, delay, dan down*) alat berat secara *real-time*,
2. Pemanfaatan GPS untuk pelacakan posisi alat berat,
3. Penyimpanan data secara *offline* menggunakan *SQLite* untuk mengantisipasi kondisi area tambang dengan keterbatasan koneksi, dengan mekanisme sinkronisasi data yang dilakukan secara otomatis saat koneksi jaringan kembali tersedia
4. Penggunaan jaringan LAN lokal untuk mendukung transfer data antar perangkat secara cepat dan stabil di lingkungan pertambangan, yang diimplementasikan sebagai jalur utama pengiriman data dari perangkat monitoring ke server lokal sebelum disinkronkan ke sistem pusat saat koneksi memungkinkan.

Dengan demikian, penelitian ini berkontribusi mengisi kekosongan di bidang monitoring *real-time* status operasional alat berat pada area pertambangan terbuka.

3 Metode Penelitian

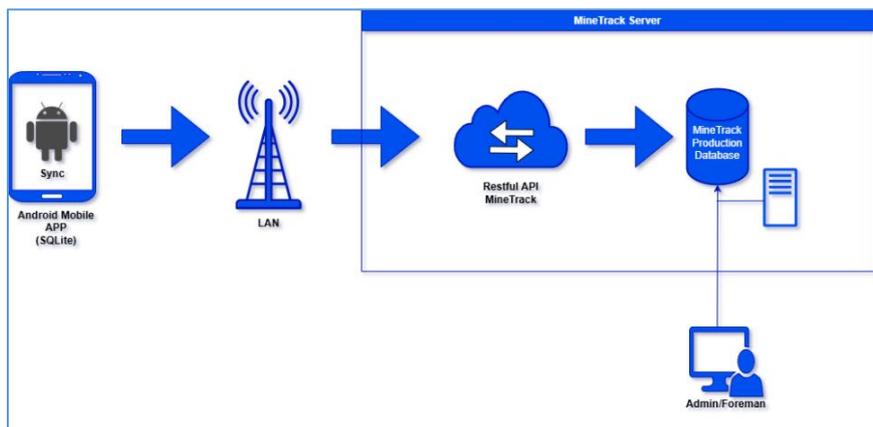
Penelitian ini menggunakan metode *Scrum*, yaitu salah satu metode dalam pengembangan perangkat lunak yang dilakukan secara bertahap (iteratif) dan fleksibel terhadap perubahan kebutuhan. Proses kerja dibagi ke dalam beberapa *sprint* untuk menyelesaikan bagian sistem secara bertahap dan menyelesaikan bagian sistem secara bertahap dan memungkinkan evaluasi pada setiap tahapannya.



Gambar 1. Alur penelitian

Sebagaimana ditampilkan pada Gambar 1, alur proses *Scrum* pada penelitian ini dimulai dari analisis kebutuhan hingga menghasilkan *product increment* melalui serangkaian tahapan, seperti *sprint planning*, *development*, dan *sprint review*, metode ini diadaptasi dari penelitian oleh Prasetyo et al. [15], yang menggunakan pendekatan *Scrum* dalam pengembangan aplikasi berbasis Android. Tahapan *Scrum* pada penelitian ini disusun dan disesuaikan berdasarkan kerangka dari penelitian tersebut. Selain itu, pendekatan serupa juga ditemukan pada penelitian oleh Ayu Ningsih et al. [16], yang menerapkan *Scrum* dalam pengembangan aplikasi berbasis *mobile*.

3.1 Arsitektur Sistem dan Mekanisme Sinkronisasi Data



Gambar 2. Arsitektur sistem

Gambar 2 menunjukkan arsitektur sistem yang dirancang terdiri dari dua komponen utama, yaitu aplikasi *mobile* berbasis Android dan server pusat yang terhubung melalui jaringan LAN. Aplikasi *mobile* digunakan oleh operator alat berat untuk mencatat dan mengelola data operasional secara langsung di lapangan. Data disimpan secara lokal di perangkat menggunakan basis data *SQLite*, yang ringan dan cocok untuk aplikasi *mobile* karena tidak membutuhkan koneksi server secara terus-menerus[17].

Komponen Sistem

Aplikasi *Mobile* (Android): Menggunakan basis data lokal *SQLite* untuk menyimpan data seperti identitas alat, aktivitas operator, posisi GPS, dan informasi ritase.

1. Server Pusat: Menyimpan data terpusat menggunakan basis data seperti *SQL Server*.
2. Jaringan LAN: Digunakan sebagai media komunikasi antara perangkat *mobile* dan server pusat saat proses sinkronisasi dilakukan.
3. *RESTful API*: Antarmuka komunikasi yang memungkinkan aplikasi *mobile* berinteraksi dengan server menggunakan protokol HTTP dan format data JSON.

Alur Data dan Sinkronisasi

Alur data dimulai dari *input* pengguna di aplikasi Android. Data tersebut disimpan di *SQLite* secara *offline*. Ketika perangkat terhubung ke jaringan LAN, sistem akan melakukan sinkronisasi ke server pusat. Proses ini mencakup:

1. Deteksi koneksi LAN: Aplikasi secara berkala memeriksa apakah koneksi LAN tersedia.
2. Trigger sinkronisasi: Sinkronisasi dilakukan secara otomatis atau manual saat koneksi tersedia.
3. Pengiriman data: Data dikirim dari *SQLite* ke server melalui *REST API* dalam format JSON.
4. Validasi dan penyimpanan server: Server akan memvalidasi data, lalu menyimpan ke dalam basis data pusat.
5. Konfirmasi & penanganan *error*: Aplikasi akan menerima respon sukses atau gagal dari server dan melakukan *retry* jika perlu.

Mekanisme Sinkronisasi Data

Sinkronisasi data merupakan elemen krusial dalam sistem ini untuk menjamin konsistensi data antara perangkat dan server pusat. Mekanisme sinkronisasi yang diterapkan meliputi:

1. Flag Sinkronisasi, diterapkan pada setiap entri data untuk menandai status apakah data sudah tersinkron atau belum.
2. *Batching* Data, digunakan untuk mengirimkan data dalam jumlah banyak sekaligus guna mengurangi *overhead* jaringan.
3. Mekanisme *Retry* dengan *Exponential Backoff*, diterapkan untuk mengantisipasi kegagalan saat pengiriman data. Jika pengiriman gagal, sistem akan menunggu selama waktu tertentu yang meningkat secara eksponensial sebelum mencoba kembali, guna menghindari beban berlebih pada jaringan atau server.
4. Penanganan Konflik, dilakukan dengan logika server *wins*, yaitu data pada server menjadi acuan utama.
5. Enkripsi Data, diterapkan pada lapisan komunikasi API (menggunakan HTTPS dan enkripsi payload) untuk menjamin keamanan data selama transmisi.

Strategi Sinkronisasi

Untuk menjamin integritas data dan efisiensi dalam proses sinkronisasi antara perangkat *mobile* dan server pusat, strategi berikut diterapkan:

1. Penerapan Flag Sinkronisasi
Setiap data yang dihasilkan di sisi klien diberi flag status sinkronisasi, misalnya *isSynced = false*. Flag ini akan diperbarui menjadi *true* hanya setelah data berhasil diterima dan dikonfirmasi oleh server. Strategi ini memudahkan pelacakan data yang belum terkirim.
2. *Batching* Data
Untuk meningkatkan efisiensi jaringan, data disusun secara *batch* (kelompok) sebelum dikirim. Mekanisme ini mengurangi jumlah permintaan HTTP secara signifikan, yang berdampak pada penurunan konsumsi *bandwidth* dan percepatan proses sinkronisasi.
3. Mekanisme *Retry* Berbasis *Exponential Backoff*
Pada kondisi jaringan yang tidak stabil atau terjadi kegagalan pengiriman data, sistem menerapkan mekanisme *retry* otomatis dengan pendekatan *exponential backoff*. Pendekatan ini memperpanjang waktu tunggu antar percobaan secara eksponensial, sehingga membantu menghindari *overloading* pada jaringan. Waktu tunggu dihitung menggunakan Persamaan (1):

$$T_n = t \times 2^{(n-1)} \quad (1)$$

di mana t adalah waktu tunggu awal dan n adalah urutan percobaan ulang, persamaan ini digunakan untuk menentukan waktu tunda yang semakin lama pada tiap upaya pengiriman ulang data.

Implementasi:

- a. *Retry* ke-1: 2 detik
 - b. *Retry* ke-2: 4 detik
 - c. *Retry* ke-3: 8 detik
 - d. *Retry* dihentikan setelah 5 kali percobaan atau bila berhasil Strategi ini efektif mengurangi kemungkinan kegagalan permanen dan mengoptimalkan penggunaan jaringan pada lingkungan dengan sinyal fluktuatif.
4. Penanganan Konflik (*Conflict Resolution*)
Jika terjadi konflik data antara klien dan server, maka diterapkan prinsip “*server wins*”, yakni data dari server dianggap sebagai sumber kebenaran utama. Alternatif lain adalah pencatatan versi data (*versioning*), yang memungkinkan sistem menyimpan riwayat perubahan dan melakukan rekonsiliasi di kemudian hari.
 5. Keamanan Sinkronisasi dengan Enkripsi
Untuk menjamin keamanan selama proses transmisi data, digunakan enkripsi pada lapisan *API*, seperti penggunaan protokol *HTTPS* dengan *TLS (Transport Layer Security)*. Hal ini bertujuan untuk melindungi data dari intersepsi atau modifikasi selama perjalanan di jaringan publik.

4 Hasil dan Pembahasan

Hasil dan Pembahasan menyajikan temuan secara sistematis berdasarkan tahapan yang dijelaskan dalam Metodologi, dengan analisis yang menghubungkan teori dan implementasi sistem. Perancangan aplikasi dilakukan menggunakan metode *Agile Scrum* secara iteratif dan inkremental dalam enam *sprint* berdurasi dua minggu. Proyek ini telah dilaksanakan dari 13 Januari hingga 6 April 2025, dengan total waktu pengembangan selama 11 minggu. Penelitian ini melaporkan hasil akhir dari proyek tersebut, yang telah selesai pada saat penulisan laporan ini dilakukan.

4.1 Analisa Kebutuhan

Perancangan Sistem Monitoring Operasional Alat Berat berbasis Android di PT Pamapersada Nusantara memerlukan perangkat lunak seperti *Windows 10*, *Android Studio*, *SQL Server Management Studio*, dan bahasa pemrograman *Java*. Perancangan dilakukan menggunakan laptop Lenovo ThinkPad T470 dengan menggunakan otak *prosesor Intel Core i5*, penyimpanan 16384 MB RAM, dan SSD (*Solid State Drive*) 256 GB untuk mendukung proses pengembangan dan pengujian aplikasi secara optimal.

4.2 Product Backlog

Tabel 2. *Product backlog*

No	<i>Product backlog</i>	Deskripsi	Prioritas
1	<i>Database</i>	Membuat Rancangan <i>Database</i>	Tinggi
2	<i>Login user</i>	<i>Autentikasi</i> user (operator) sebelum akses aplikasi	Tinggi
3	Peta Lokasi & Koordinat	Menampilkan posisi alat/operator secara <i>real-time</i> di peta	Tinggi
4	Status Aktivitas Operator	Status seperti “ <i>Hauling</i> ”, “ <i>Ready</i> ”, “ <i>Ready Production</i> ”	Tinggi
5	Kecepatan Alat (0 km/h)	Informasi <i>speed</i> kendaraan saat ini	Sedang
6	Jumlah Rit	Menampilkan jumlah ritase saat ini	Tinggi
7	Identitas Alat/Operator	Contoh: “HS TEST 001” dan “Agung (2020)”	Tinggi
8	Status Koneksi (<i>WiFi Icon</i>)	Menunjukkan konektivitas perangkat	Sedang
9	<i>Dump Assignment</i>	Menampilkan tujuan pembuangan material	Tinggi
10	Pesan Sistem	Panel notifikasi atau <i>error (No Message Found)</i>	Rendah
11	Menu Navigasi (Kiri)	Akses menu seperti profil, <i>checklist</i> , pengaturan, peringatan	Tinggi
12	<i>Refresh</i> dan <i>Center Map</i>	Tombol untuk perbarui dan atur ulang tampilan peta	Sedang

13	Versi Aplikasi	Informasi versi aplikasi Untuk keperluan <i>debugging/support</i>	Rendah
14	Waktu dan Tanggal	Waktu <i>real-time</i> di bawah tampilan peta	Sedang

Pada Tabel 2 pengumpulan kebutuhan sistem dilakukan secara sistematis agar aplikasi sesuai dengan kebutuhan operasional PT. Pamapersada Nusantara. Kebutuhan yang dihimpun mencakup fitur penting seperti desain *database*, *autentikasi login*, pelacakan posisi alat secara *real-time*, status aktivitas operator, identitas alat, serta informasi ritase dan kecepatan. Kebutuhan non-fungsional meliputi antarmuka pengguna, panel notifikasi, versi aplikasi, tombol *refresh*, dan navigasi menu. Seluruh kebutuhan dianalisis dan dirinci ke dalam *backlog item* berdasarkan prioritas: tinggi, sedang, dan rendah. Setiap item backlog memiliki nilai bisnis dan dikembangkan bertahap melalui *sprint*. Daftar *backlog* ini disusun oleh *Product Owner* sebagai acuan kerja tim pengembang dalam setiap siklus *Scrum*.

4.3 Sprint Planning

Tabel 3. *Sprint planning*

<i>Sprint</i>	Tanggal Mulai-Selesai	Fitur yang dikerjakan	Prioritas	Keterangan
1	13 Jan – 26 Jan 2025	<i>Database Login User</i> Peta lokasi & Koordinat,	Tinggi	pondasi awal aplikasi pelacakan posisi alat berat
2	27 Jan – 9 Feb 2025	Status Aktivitas Operator, Jumlah Rit, Identitas Alat/Operator	Tinggi	Monitoring aktivitas alat dan identitas operator
3	10 Feb – 23 Feb 2025	<i>Dump Assignment</i> , Menu Navigasi (Kiri), Kecepatan Alat	Tinggi & Sedang	Fitur navigasi, pengelompokan ritase, dan pemantauan kecepatan
4	24 Feb – 9 Mar 2025	Status Koneksi (<i>WiFi</i>), <i>Refresh & Center Map</i> , Waktu & Tanggal	Sedang	Fitur konektivitas dan pembaruan tampilan peta secara <i>real-time</i>
5	10 Mar – 23 Mar 2025	Pesan Sistem, Versi Aplikasi	Rendah	Fitur tambahan untuk notifikasi dan informasi versi aplikasi
6	24 Mar – 6 Apr 2025	Perbaikan <i>bug</i> , optimasi performa, validasi akhir	Sedang	<i>Sprint</i> final untuk penyempurnaan, uji coba, dan persiapan sebelum implementasi penuh

Tabel 3 merangkum rencana enam *sprint* pengembangan aplikasi monitoring alat berat selama 11 minggu (13 Januari – 6 April 2025), dengan tiap *sprint* berdurasi dua minggu. *Sprint* 1–2 difokuskan pada pembangunan fondasi sistem seperti *database*, *login*, peta lokasi, status aktivitas, dan data operator. *Sprint* 3–4 menambahkan fitur operasional lanjutan seperti *dump assignment*, navigasi menu, kecepatan alat, koneksi *WiFi*, dan *refresh* peta. *Sprint* 5 berisi pengembangan fungsi pendukung (pesan sistem dan info versi aplikasi), sedangkan *Sprint* 6 difokuskan pada perbaikan *bug*, optimasi performa, dan validasi akhir. Setiap fitur diklasifikasikan berdasarkan prioritas (Tinggi, Sedang, Rendah) dan dilengkapi deskripsi singkat untuk memastikan arah pengembangan sesuai nilai bisnis dan kebutuhan pengguna.

4.4 Sprint Backlog

Tabel 4. *Sprint backlog*

<i>Sprint</i>	Item <i>Sprint backlog</i>	Keterangan
1	- Perancangan <i>database</i> - <i>Autentikasi login</i> pengguna - Integrasi peta lokasi & koordinat alat	Fondasi awal sistem: struktur data, <i>autentikasi</i> , dan pelacakan alat berat secara <i>real-time</i>
2	- Tampilan status aktivitas operator - Informasi jumlah ritase - Identitas alat dan operator	Fitur inti monitoring alat dan operator dalam kegiatan operasional harian
3	- <i>Dump assignment</i> (penugasan ritase)	Navigasi dan pengelolaan informasi ritase

	- Menu navigasi (kiri) - Tampilan kecepatan alat	serta pelaporan kecepatan kendaraan
4	- Status koneksi jaringan (<i>icon WiFi</i>) - Tombol <i>refresh & center map</i> - Penambahan waktu & tanggal	Fitur tambahan untuk kestabilan koneksi dan akurasi tampilan data di aplikasi
5	- Pesan sistem dan notifikasi - Versi aplikasi (<i>about app</i>)	Informasi pendukung sistem, <i>debugging</i> , dan informasi rilis
6	- Perbaikan <i>bug</i> - Optimasi performa - Validasi fitur secara menyeluruh	<i>Sprint</i> final untuk pengujian menyeluruh, penyempurnaan sistem, dan persiapan rilis aplikasi

Tabel 4 merangkum *Sprint Backlog* untuk enam iterasi pengembangan aplikasi, di mana setiap *sprint* mencantumkan daftar tugas terperinci yang diambil dari *Product Backlog*. *Sprint* 1–2 difokuskan pada pondasi teknis dan *autentikasi* pengguna (perancangan *database*, *login*, peta lokasi, status aktivitas, dan identitas operator), *Sprint* 3–4 menitikberatkan pada fitur operasional lanjutan (*dump assignment*, navigasi menu, kecepatan alat, koneksi *WiFi*, tombol *refresh*, serta penambahan waktu dan tanggal), *Sprint* 5 mengimplementasikan fungsi pendukung (pesan sistem, notifikasi, dan informasi versi aplikasi), dan *Sprint* 6 dialokasikan untuk perbaikan *bug*, optimasi performa, serta validasi menyeluruh. Kolom keterangan di sebelahnya menjelaskan tujuan utama setiap *sprint*, sehingga tim pengembang memiliki panduan yang jelas untuk mencapai *deliverable* sesuai prioritas dan jadwal.

4.5 Development dan daily Sprint

Development

Tahap *development* merupakan inti pengerjaan sistem, di mana *item* dalam *sprint backlog* dikembangkan oleh tim selama satu *sprint* (2 minggu) secara iteratif dan inkremental sesuai prinsip *Scrum*. Fitur dikembangkan menggunakan Android Studio, *Java*, dan *SQL Server*, dengan tanggung jawab tim pada kualitas kode dan kesesuaian kebutuhan pengguna. Pendekatan *agile* memungkinkan adanya perubahan selama proses berlangsung. Setiap akhir *sprint*, fitur dievaluasi untuk memastikan siap diuji atau digunakan.

Daily Sprint

Selama proses *development* dalam satu *sprint*, tim melaksanakan *Daily Scrum* atau pertemuan harian. Kegiatan ini dilakukan setiap hari kerja, berdurasi sekitar 15 menit, dan bertujuan untuk meningkatkan transparansi serta koordinasi antar anggota tim. Dalam *daily scrum*, setiap anggota menyampaikan 3 hal utama:

1. Apa saja yang telah dikerjakan dari pertemuan sebelumnya,
2. Apa saja yang akan dikerjakan tim hari ini,
3. Kendala atau hambatan yang dihadapi.

Scrum Master bertugas memfasilitasi pertemuan ini agar tetap fokus, efisien, dan membantu menyelesaikan kendala yang mungkin menghambat progress *sprint*. Dengan adanya *daily scrum*, tim dapat menyesuaikan beban kerja secara dinamis dan memastikan target *sprint* tetap tercapai sesuai jadwal.

Tabel 5. *Daily scrum sprint 1 (13–26 jan 2025)*

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–2	<i>Setup</i> repo, <i>database</i> awal	Koneksi <i>database</i> Android	JDBC tidak kompatibel	Gunakan <i>REST API</i>
Hari 3–5	<i>Login</i> UI & <i>backend</i>	Validasi & integrasi	Tidak ada	-
Hari 6–10	Integrasi <i>Google Maps</i>	Tampilkan <i>marker</i> alat berat	Masalah koordinat & <i>API key</i>	<i>Update</i> konfigurasi <i>Maps API</i>
Hari 11–13	<i>Layout</i> finalisasi	Uji coba <i>login</i> + peta	<i>Layout</i> tumpang tindih	Perbaiki struktur UI
Hari 14	<i>Sprint Review</i> &	Evaluasi fitur dan	-	Disepakati

<i>Retrospective</i>	<i>sprint</i>	perbaikan
----------------------	---------------	-----------

Pada Tabel 5 akhir dari *Sprint 1*, dilakukan *Sprint Review* terhadap fitur awal sistem yaitu *login*, integrasi peta lokasi, serta koneksi ke *database*. Fitur *login* berhasil diuji dengan validasi dasar, dan *Google Maps* berhasil ditampilkan dengan marker alat berat. Kendala utama yang dihadapi adalah ketidakcocokan metode koneksi langsung ke *database*, yang diselesaikan dengan pengalihan ke *REST API*. Dalam *Sprint Retrospective*, tim menyepakati untuk memperbaiki struktur UI agar lebih rapi serta mendokumentasikan lebih rinci konfigurasi *API Maps*.

Tabel 6. Daily scrum sprint 2 (27 jan – 9 feb 2025)

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–3	Tampilan aktivitas operator	Sinkronisasi data ritase	-	-
Hari 4–6	Hitung jumlah rit per alat	Validasi nilai otomatis	Logika <i>looping error</i>	<i>Debugging</i> dan revisi algoritma
Hari 7–10	Identitas alat/operator	Integrasi dengan <i>login</i>	Data tidak sinkron	Tambahkan validasi ID
Hari 11–13	Tes fitur monitoring	Sinkronisasi <i>backend</i>	-	-
Hari 14	<i>Sprint Review & Retrospective</i>	Evaluasi	-	<i>Update</i> pada struktur data

Tabel 6 *Sprint Review* pada *Sprint 2* difokuskan pada evaluasi fitur aktivitas operator, jumlah ritase, dan identitas alat berat. Fitur berhasil menampilkan data dengan akurasi dasar, namun ditemukan logika *looping* perhitungan ritase yang masih belum konsisten. Dalam *Sprint Retrospective*, disepakati perbaikan algoritma perhitungan serta validasi ID alat dan operator agar data tidak tumpang tindih pada tampilan.

Tabel 7. Daily scrum sprint 3 (10–23 feb 2025)

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–3	Struktur <i>dump assignment</i>	Relasi ke alat dan operator	Salah <i>mapping dump</i>	Tambahkan validasi <i>dump ID</i>
Hari 4–6	Menu navigasi (sidebar/kiri)	Navigasi antar fitur	<i>Layout error</i> pada Android 10	Atur ulang <i>responsive UI</i>
Hari 7–10	Pengambilan data kecepatan alat	Simulasi gerakan alat	Sensor simulasi <i>delay</i>	Gunakan nilai <i>dummy real-time</i>
Hari 11–13	Uji fitur baru	Revisi koneksi <i>dump & kecepatan</i>	-	-
Hari 14	<i>Sprint Review & Retrospective</i>	Evaluasi hasil <i>sprint</i>	-	Disepakati optimasi performa

Pada Tabel 7 akhir *Sprint 3*, dilakukan *Sprint Review* untuk mengevaluasi fungsionalitas fitur *dump assignment*, navigasi menu, dan kecepatan alat berat. Secara umum, fitur-fitur berhasil diimplementasikan dengan baik, namun ditemukan kendala pada proses *mapping dump* yang kurang akurat serta keterlambatan dalam pembacaan data kecepatan alat. Dalam *Sprint Retrospective*, tim menyepakati perlunya validasi ID *dump* yang lebih ketat, serta optimalisasi simulasi sensor kecepatan. Selain itu, dilakukan revisi minor pada tampilan navigasi agar lebih responsif di berbagai versi Android.

Tabel 8. Daily scrum sprint 4 (24 feb – 9 mar 2025)

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–3	Deteksi status <i>WiFi</i>	Notifikasi saat <i>offline</i>	Tidak tampil di semua <i>device</i>	Tambah pengecekan sistem
Hari 4–6	Tombol <i>refresh</i> dan	<i>Test</i> lokasi &	<i>Marker</i> hilang	Tambah reload

	<i>center map</i>	ulang tampil peta	saat <i>refresh</i>	fungsi <i>marker</i>
Hari 7–10	Waktu dan tanggal dinamis	Format waktu lokal	Salah zona waktu	Gunakan zona GMT+7
Hari 11–13	Tes keseluruhan fitur <i>sprint</i>	Finalisasi UI kecil	-	-
Hari 14	<i>Sprint Review & Retrospective</i>	Evaluasi dan persiapan <i>sprint 5</i>	-	-

Pada Tabel 8 *Sprint Review Sprint 4* mengevaluasi fitur status koneksi alat, tombol *refresh* peta, serta penyajian waktu dan tanggal *real-time*. Secara umum fitur berjalan dengan baik, namun terdapat kendala pada deteksi *WiFi* di beberapa perangkat serta *marker* peta yang hilang setelah *refresh*. Dalam *Sprint Retrospective*, tim menyepakati untuk menambahkan pengecekan status sistem secara berkala dan memperbaiki logika *reload marker* agar peta selalu menampilkan posisi alat secara akurat.

Tabel 9. Daily scrum sprint 5 (10–23 mar 2025)

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–3	Sistem notifikasi & pesan	<i>Format alert & toast</i>	Tidak muncul di <i>background</i>	Gunakan <i>Notification Manager</i>
Hari 4–6	Halaman versi aplikasi	Info versi & tim <i>dev</i>	Versi tidak <i>update</i>	Sinkronkan dengan <i>Gradle</i>
Hari 7–10	Uji coba aplikasi pada perangkat	Validasi tampilan dan info	-	-
Hari 11–13	<i>Debug</i> minor issues	Optimalisasi kecepatan	-	-
Hari 14	<i>Sprint Review & Retrospective</i>	Klarifikasi kelengkapan fitur	-	Siap validasi final

Pada tabel 9 akhir *Sprint 5*, dilakukan *Sprint Review* terhadap fitur pesan sistem dan halaman informasi versi aplikasi. Notifikasi berhasil ditampilkan saat kondisi tertentu, namun tidak muncul jika aplikasi berjalan di *background*. Selain itu, informasi versi aplikasi belum otomatis mengikuti *update build*. Dalam *Sprint Retrospective*, tim menyepakati solusi berupa penggunaan *Notification Manager* Android serta sinkronisasi info versi dengan sistem *build Gradle*.

Tabel 10 Daily scrum sprint 6 (24 mar – 6 apr 2025)

Hari	Progress	Rencana	Kendala	Solusi/tindak lanjut
Hari 1–3	Uji sistem menyeluruh	Cek integrasi semua fitur	Minor <i>bug</i> ritase	Fix urutan logika
Hari 4–6	Optimasi performa & UI	Cek <i>smooth</i> UI	Animasi kurang responsif	Gunakan <i>transition</i> ringan
Hari 7–10	Pengujian perangkat berbeda	Kompatibilitas Android 9–13	Versi 9 error GPS	Tambah izin lokasi manual
Hari 11–13	<i>Bug fixing</i> terakhir	<i>Backup</i> proyek dan dokumentasi	-	-
Hari 14	<i>Sprint Review & Penutupan</i>	<i>Retrospective</i> proyek	-	Siap diserahkan ke <i>stakeholder</i>

Pada tabel 10 *Sprint Review* pada *Sprint 6* menandai akhir dari proses pengembangan. Fitur-fitur diuji secara keseluruhan untuk memastikan integrasi berjalan baik, termasuk pengujian di berbagai versi Android. Ditemukan *bug* minor pada fitur ritase serta *error* GPS di Android 9. Dalam *Sprint Retrospective*, tim menyepakati perbaikan logika ritase, optimalisasi performa UI, serta penyesuaian izin lokasi pada Android versi lama. Aplikasi dinyatakan selesai dan siap untuk diserahkan kepada *stakeholder*.



Gambar 3. Pelatihan penggunaan aplikasi *minetrack* kepada operator lapangan PT Pamapersada Nusantara

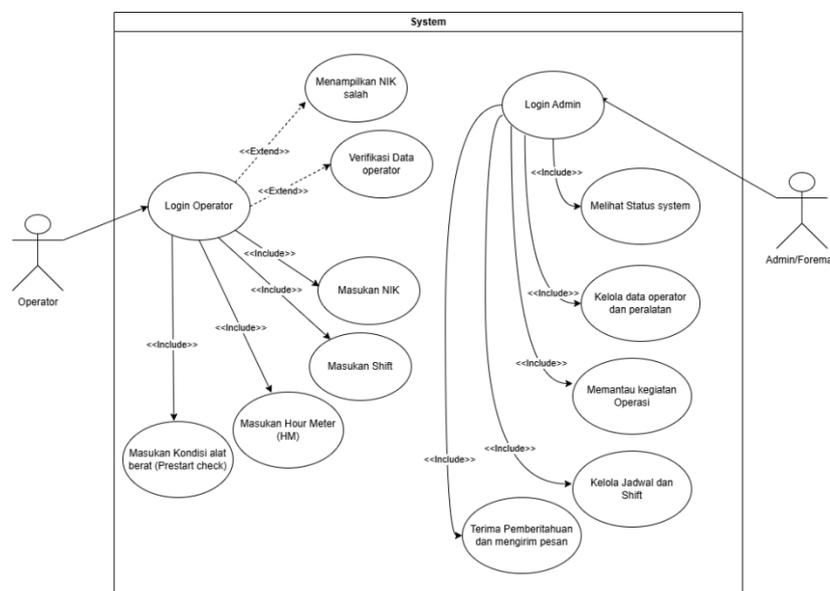
Sebagai bagian dari validasi akhir pada gambar 3, dilakukan kegiatan pelatihan dan demonstrasi penggunaan aplikasi kepada operator dan pengawas lapangan di PT Pamapersada Nusantara. Kegiatan ini dilakukan langsung di lapangan menggunakan perangkat Android dan didampingi oleh penyampaian materi pelatihan melalui presentasi digital. Dalam sesi ini, pengguna mencoba fitur pelaporan status alat berat, pelacakan posisi GPS, dan input ritase.

Meskipun tidak dilakukan pengujian formal berbasis kuesioner, umpan balik lisan dari peserta pelatihan menunjukkan bahwa aplikasi *MineTrack* mudah digunakan, fitur-fitur berjalan sesuai kebutuhan operasional, serta mendukung efisiensi pelaporan dan akurasi data lapangan. Oleh karena itu, pelatihan ini sekaligus menjadi bentuk validasi pengguna awal terhadap sistem yang dikembangkan.

4.6 *Sprint review & Retrospective*

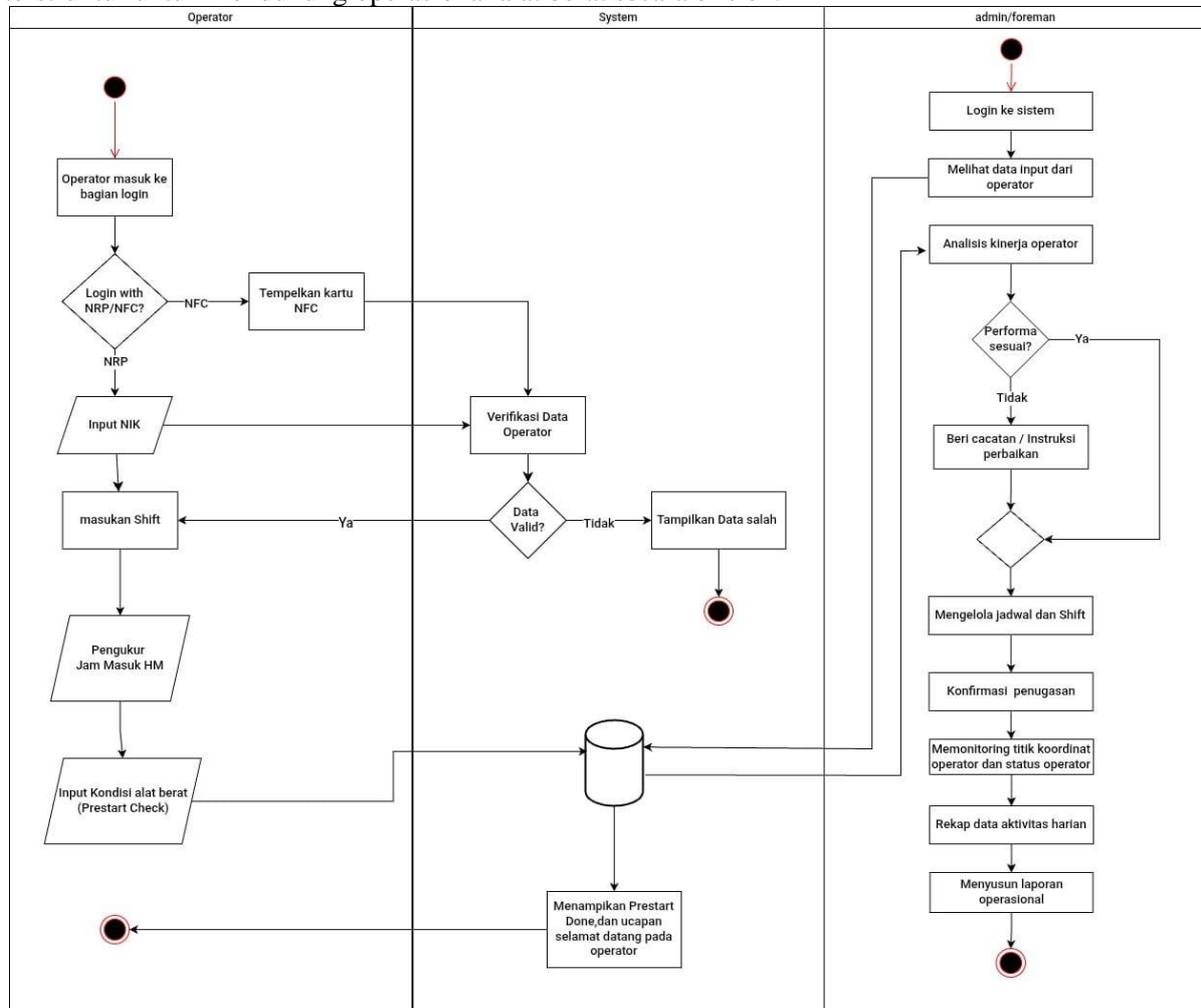
Tahapan *Sprint Review* dan *Retrospective* telah dibahas terpadu pada bagian 4.5 bersamaan dengan aktivitas harian tiap *sprint*, guna efisiensi tanpa mengurangi substansi evaluasi. Setiap akhir *sprint* mencakup penilaian fitur (*Review*) dan refleksi tim (*Retrospective*), sehingga keberhasilan dan kendala tiap *sprint* tersaji secara jelas dan sistematis.

4.7 *Product Increment*



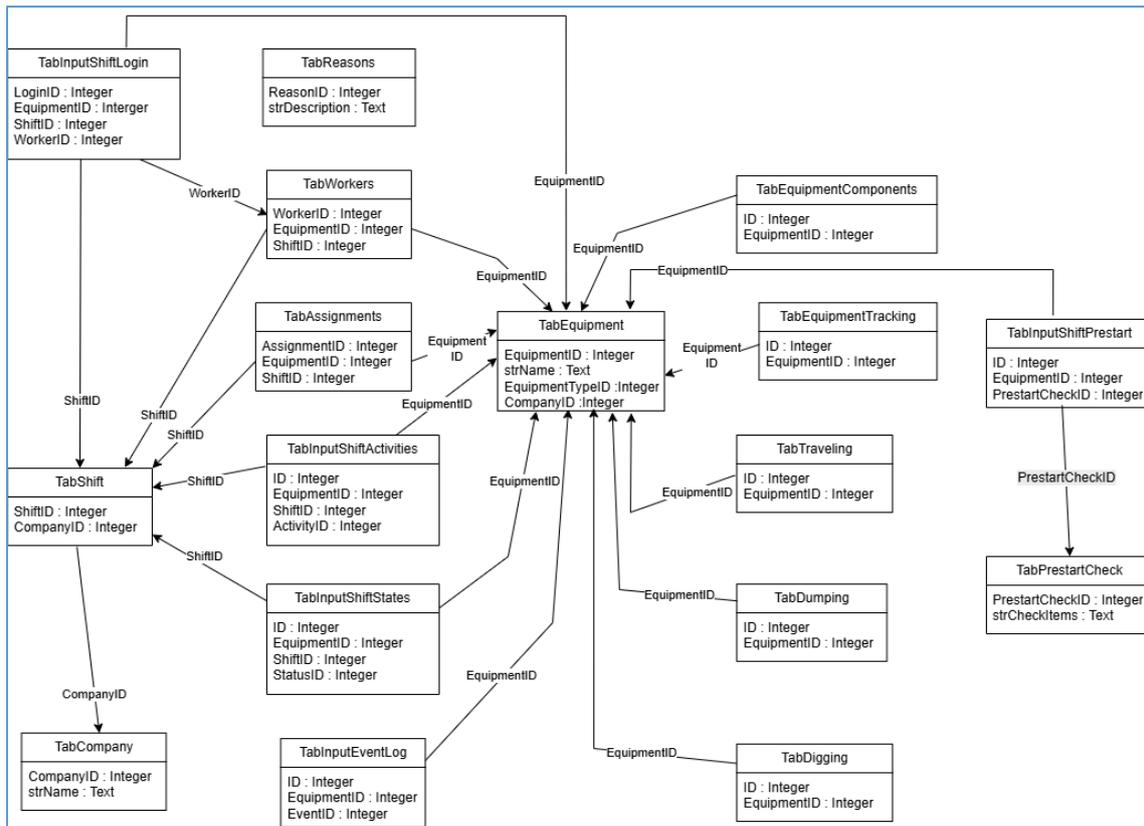
Gambar 4. Use case diagram

Gambar 4 menunjukkan *use case diagram* yang menggambarkan interaksi antara Operator dan Admin/Foreman dalam sistem monitoring alat berat. Operator melakukan *login*, verifikasi data, serta *input shift*, HM, dan kondisi alat melalui *form pre start check*. Sementara itu, Admin/Foreman memiliki peran lebih luas seperti mengelola data, memantau operasi, mengatur jadwal, melihat status sistem, serta berkomunikasi dengan operator. Diagram ini menegaskan pembagian tugas yang terstruktur untuk mendukung operasional alat berat secara efisien.



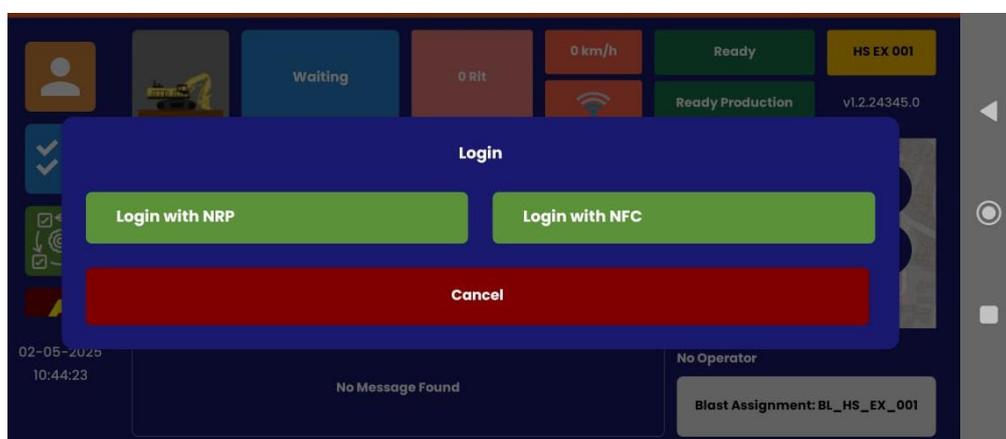
Gambar 5. Activity diagram

Gambar 5 menunjukkan diagram aktivitas sistem monitoring alat berat dengan dua peran utama: operator dan admin/foreman. Operator *login* menggunakan NRF atau NFC, lalu menginput NIK, *shift*, *hour meter* (HM), dan kondisi alat berat melalui *pre start check*. Setelah data tervalidasi, sistem menampilkan notifikasi sukses. Sementara itu, admin/foreman *login* untuk memantau *input* operator, menganalisis kinerja, mengelola jadwal dan *shift*, memantau koordinat operator, merekap data harian, dan menyusun laporan operasional. Diagram ini menggambarkan alur kerja kolaboratif untuk menjamin akurasi dan kelancaran proses monitoring.



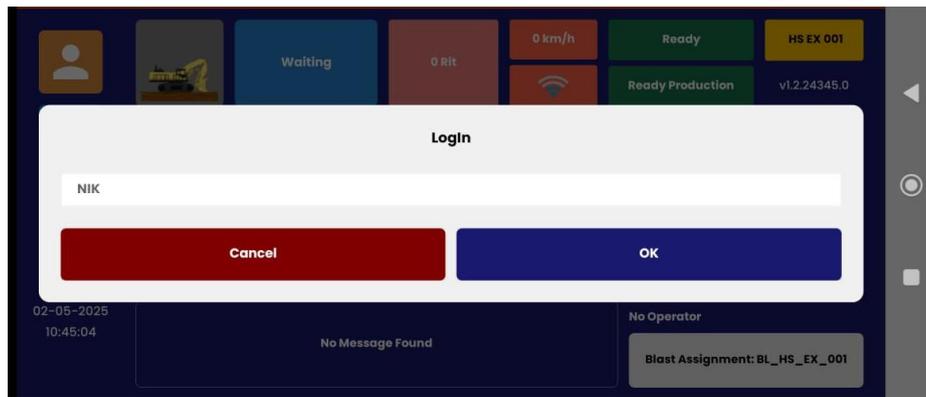
Gambar 6. Class diagram

Gambar 6 menunjukkan *class diagram* yang menggambarkan relasi antar entitas utama dalam sistem monitoring alat berat berbasis *shift*. Entitas seperti *tabEquipment*, *tabShift*, dan *tabWorkers* menjadi pusat integrasi data, terhubung dengan aktivitas, status, *log* kejadian, pelacakan, hingga data operasional seperti *dumping* dan *digging*. *Login* operator dicatat melalui *tabInputShiftLogin*, sementara hasil *pre-start check* disimpan pada *tabInputShiftPrestart* yang merujuk ke *tabPrestartChecks*. *tabCompany* mengelompokkan alat dan *shift* berdasarkan kepemilikan, sedangkan *tabAssignments* mencatat penugasan alat. Diagram ini merepresentasikan keterpaduan sistem dalam mendukung pelacakan aktivitas dan analisis performa alat secara menyeluruh.



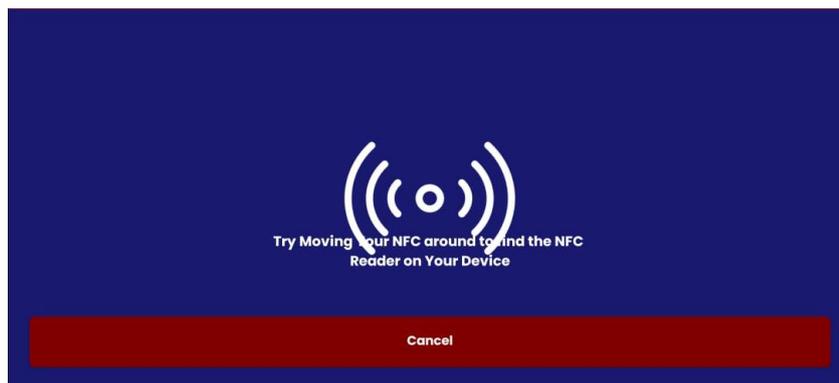
Gambar 7. Tampilan login

Gambar 7 menunjukkan halaman *login* aplikasi monitoring alat berat dengan dua metode *otentikasi*: NRP dan NFC. NRP memungkinkan *login* manual, sedangkan NFC memfasilitasi *login* otomatis dengan pemindaian. Antarmuka dirancang sederhana, responsif, dan aman, dengan tombol kontras untuk kemudahan penggunaan di lapangan.



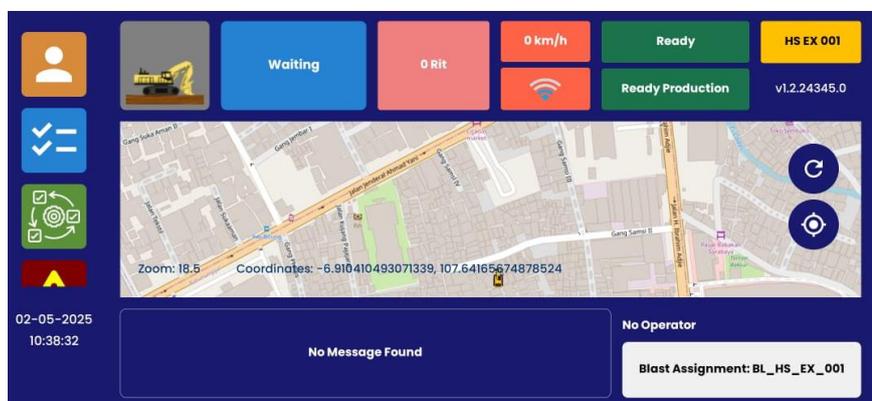
Gambar 8. Tampilan login NRP

Gambar 8 menunjukkan antarmuka login dengan metode NRP, di mana pengguna memasukkan NIK untuk autentikasi. Tampilan minimalis dengan tombol “Cancel” dan “OK” memudahkan proses login manual yang cepat dan aman.



Gambar 9. Tampilan login NFC

Gambar 9 menampilkan antarmuka login menggunakan NFC, di mana pengguna cukup mendekatkan kartu atau perangkat ke pemindai untuk autentikasi otomatis. Fitur ini mempercepat proses login dan mengurangi risiko kesalahan input pada sistem monitoring alat berat.



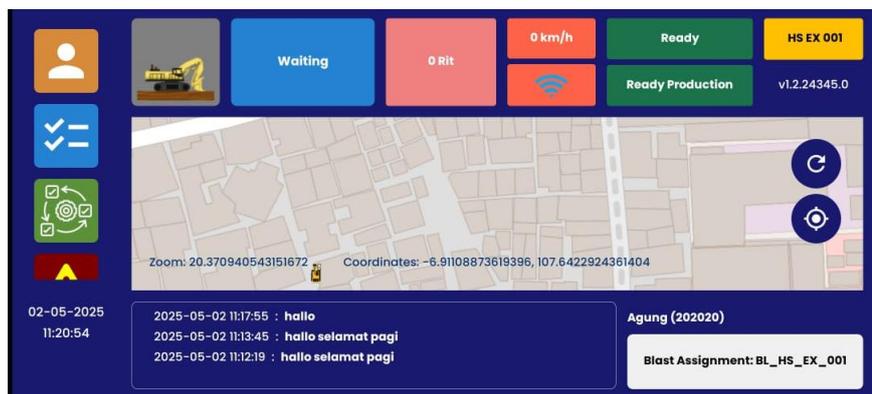
Gambar 10. Tampilan halaman utama

Gambar 10 menampilkan halaman utama aplikasi monitoring alat berat berbasis Android yang menyajikan data *real-time* seperti status alat, kecepatan, ritase, dan identitas unit. Peta GPS di tengah menunjukkan posisi alat dengan fitur pembaruan dan pengaturan, sementara bagian bawah menampilkan informasi waktu, status operator, dan *assignment* untuk mendukung operasional lapangan secara langsung.



Gambar 11. Tampilan status operator

Gambar 11 menunjukkan tampilan pemilihan status alat berat dengan empat opsi (*Ready*, *Delay*, *Down*, *StandBy*) yang dibedakan oleh warna dan *icon*. Antarmuka ini dirancang agar mudah dan cepat digunakan operator untuk memperbarui kondisi alat secara *real-time*, dengan tombol *Cancel* untuk membatalkan tindakan.



Gambar 12. Notifikasi pesan sistem

Gambar 12 menampilkan panel notifikasi di bawah peta, yang menampilkan pesan sistem dengan *timestamp*, identitas operator, dan *blast assignment* aktif. Fitur ini memudahkan operator memantau instruksi atau status sistem secara *real-time* dalam aplikasi.

5 Kesimpulan

Penelitian ini berhasil merancang sistem monitoring operasional alat berat berbasis Android bernama *MineTrack* untuk PT Pamapersada Nusantara sebagai solusi atas kendala monitoring manual seperti pencatatan aktivitas yang tidak akurat, status operator yang tidak konsisten, dan ketiadaan informasi lokasi *real-time*. Aplikasi ini mengintegrasikan GPS untuk pelacakan, SQLite untuk penyimpanan lokal, serta dikembangkan menggunakan metode *Agile Scrum* selama enam sprint dalam 11 minggu. Hasil implementasi menunjukkan peningkatan efisiensi pelaporan, akurasi data operasional, percepatan pengambilan keputusan, dan pengurangan risiko *downtime*. Validasi dilakukan melalui pelatihan pengguna sebagai bentuk awal *User Acceptance Testing* (UAT) informal, dengan umpan balik positif terkait kemudahan penggunaan dan manfaatnya di lapangan. Aplikasi ini juga dirancang agar tetap berjalan dalam kondisi jaringan terbatas dengan memanfaatkan penyimpanan lokal dan jaringan LAN. Secara keseluruhan, *MineTrack* memberikan kontribusi signifikan dalam efisiensi manajemen alat berat di industri tambang dan berpotensi dikembangkan lebih lanjut melalui integrasi *cloud*, analitik prediktif, dan sistem notifikasi otomatis untuk mendukung transformasi digital sektor pertambangan secara menyeluruh.

Referensi

- [1] M. Asof, R. Pebrianto, and A. Z. Ataariq, "Stabilisasi Lereng Timbunan Ditinjau dari Komposisi Mineral Penyusun Tanah pada Lereng Timbunan Tambang *Embankment Slope Stability Viewed from the Composition of Soil Minerals on the Mine Emission Slope*," *Agustus*, Vol. 7, No. 3, 2023, [Online]. Available: <http://ejournal.ft.unsri.ac.id/index.php/JP>
- [2] J. Plavsic and I. Miskovic, "VR-based Digital Twin for Remote Monitoring of Mining Equipment : Architecture and a case study," 2024, doi: 10.1016/j.vrih.2023.12.02.
- [3] A. E. Purwandanu and M. T. Toha, "Parameter Affecting Productivity Drilling, Blasting, and Shovel-Dump Truck System in Andesite Mining," *Jurnal Pertambangan*, Vol. 4, No. 2, 2020, [Online]. Available: <http://ejournal.ft.unsri.ac.id/index.php/JP>
- [4] H. R. Alla, R. Hall, and D. B. Apel, "Performance Evaluation of Near Real-Time Condition Monitoring in Haul Trucks," *Int J Min Sci Technol*, Vol. 30, No. 6, pp. 909–915, Nov. 2020, doi: 10.1016/j.ijmst.2020.05.024.
- [5] M. Ardy Zailani. S.T and S. Eko Purnomo, "Optimasi Pengawasan Tambang Quarry Berbasis Teknologi Digital Melalui System Terpadu Quarry Mning Commad Center (QMCC) di PT. Semen Indonesia (Persero) TBK (Site Tuban 1)," 2022.
- [6] F. Damayanti and W. Joko Pranoto, "Perancangan Website Departement Profile Plant Hauling District Indo PT. Pamapersada Nusantara Kalimantan Timur," *Nusantara Jurnal Pengabdian Kepada Masyarakat*, Vol. 4, No. 1, pp. 83–92, Dec. 2023, doi: 10.55606/nusantara.v4i1.2357.
- [7] R. N. Dasmen, S. Elfahmi, and W. D. Septiani, "Analisa Jaringan Local Area Network (LAN) dengan Aplikasi Cisco Packet Tracer," *Decode: Jurnal Pendidikan Teknologi Informasi*, Vol. 2, No. 2, pp. 45–49, Jun. 2022, doi: 10.51454/decode.v2i2.34.
- [8] N. S. P. K. Yadati, "GPS Accuracy and Enhancement in Android Development," *Academy and Industry Research Collaboration Center (AIRCC)*, Aug. 2024, pp. 53–58. doi: 10.5121/csit.2024.141504.
- [9] H. Tri Nurpebri, "Mobile Information System Pendataan Hasil Pengolahan Biji Timah dengan Alat Meja Goyang pada PT Timah Tbk," *Sistem Informasi dan Komputer*, Vol. 10, pp. 197–203, Jul. 2021, doi: 10.32736/sisfokom.v10i02.1162.
- [10] F. A. N. Fatin, M. Nurkamid, R. Meimaharani, and A. B. Maskula, "Real-Time Monitoring of Gas Fields: Prototype at PT Gamma Energi Pratama Bogor," *Jurnal Teknik Informatika*, Vol. 16, No. 1, pp. 89–99, May 2023, doi: 10.15408/jti.v16i1.31832.
- [11] E. Mulyadi, A. Trihariprasetya, I. Gede Wiryawan, J. Teknologi Informasi, and P. Negeri Jember Jember, "Penerapan Sistem Presensi Mobile dengan menggunakan Sensor GPS (Klinik Pratama X di Jember)," Mar. 2020.
- [12] H. Suhendi, G. Devira Ramady, and J. Y. Prasetyo, "Aplikasi Sistem Pelacakan Lokasi Kendaraan Trans Metro Bandung berbasis Android dengan GPS Tracking Real Time," Apr. 2023.
- [13] M. Waruni Kasrani dkk, A. Fitri Saiful Rahman, and R. Ramadoni, "Monitoring Lokasi Komponen Alat Berat berbasis Internet of Things Pada PT. Thiess BSF Batakan," 2023.
- [14] A. Yerintra Djafniel Yudanur, "Sistem Monitoring Tegangan, Arus, dan Suhu pada Unit Alat Berat berbasis Internet of Things di PT Armada Hada Graha," 2024.
- [15] H. Prasetyo, N. Radita, and A. Tirtana, "Aplikasi Manajemen Pendakian berbasis Android menggunakan Metode Scrum," *Teknika*, Vol. 11, No. 3, pp. 180–189, Oct. 2022, doi: 10.34148/teknika.v11i3.514.
- [16] D. P. Ayuningsih, I. N. Dewi, and A. Rohmani, "Scrum Framework Implementation of Fish Mobile Auction Module in Pasar Iwak Marketplace," *Sinkron*, Vol. 8, No. 2, pp. 752–761, Apr. 2023, doi: 10.33395/sinkron.v8i2.12096.
- [17] M. N. Amin, A. B. P. Negara, and A. Perwitasari, "Implementasi Mesin Penerjemah Statistik pada Aplikasi Chatting berbasis Android dengan Moses Decoder," *InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan*, 2021, doi: 10.30743/infotekjar.v6i1.4025.