Tugas Deliverable 4

II3240 Rekayasa Sistem dan Teknologi Informasi

Log-Shield

Sistem Intelijen Logistik Kemanusiaan Berbasis AI dan IoT

Untuk Pemenuhan Mandat Enam Tepat BNPB

Disusun oleh

Kelompok 09

Andhika M. Addiputra	18223005
Ferro Arka Berlian	18223027
Luckman Fakhmanidris A.	18223041
M. Naufal Fathan	18223059
Atharizza M. Athaya	18223079

PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2026

DAFTAR ISI

1. PENDAHULUAN.....8
1.1 Latar Belakang..... 8
1.2 Identifikasi Masalah..... 8
1.3 Tujuan dan Manfaat.....9
1.3.1 Tujuan.....9
1.3.2 Manfaat.....10
1.4 Ruang Lingkup.....10
1.5 Jadwal Kegiatan dan Pembagian Tugas..... 10
1.5.1 Jadwal Kegiatan.....11
1.5.2 Pembagian Tugas.....11
1.6 Sistematika Penulisan.....12
2. NEEDS ANALYSIS.....13
2.1 Operations Analysis..... 13
2.1.1 Analisis Kebutuhan & Identifikasi Kekurangan..... 13
2.1.2 Analisis Kebutuhan Pengguna..... 14
2.1.3 Penetapan Tujuan Operasional.....14
2.2 Functional Analysis.....15
2.2.1 Functional Requirements.....15
2.2.2 Proses Trade-Off..... 16
2.3 Feasibility Analysis.....16
2.3.1 Proses Pengembangan.....16
2.3.2 Strategi Pengembangan.....17
2.3.3 Estimasi Biaya.....17
2.3.4 Pendekatan Desain dan Metode Evaluasi.....17
2.3.5 Isu Produksi dan Konsep Operasional..... 18
2.3.6 Antisipasi Risiko..... 18
2.4 Needs Validation..... 18
2.4.1 Pentingnya Validasi Kebutuhan..... 19
2.4.2 Model Efektivitas Operasional.....19
2.4.3 Measures of Effectiveness (MoE)..... 19
2.4.4 Measures of Performance (MoP)..... 20
2.4.5 Piramida Analysis..... 20
3. CONCEPT EXPLORATION.....21
3.1 Operational Requirements Analysis.....21
3.1.1 Tujuan dan Process of Requirements Analysis..... 21
3.1.2 Requirements Elicitation.....21
3.1.3 Requirements Testing.....21
3.1.4 Requirements Validation..... 22
3.2 Performance Requirements Formulation..... 22

3.2.1 Transformation of Operational Requirements into Performance Requirements....22
3.2.2 Derivation of Sub-system Functions....23
3.2.3 Categorization of Functions....23
3.2.4 Formulation of Performance Characteristics....24
3.3 Implementation Concept Exploration....24
3.3.1 Sistem Predecessor....24
3.3.2 Pendekatan Inovatif....24
3.3.3 Pengembangan Sistem dan Sub-Sistem....25
3.3.4 Validasi dan Pengujian....25
3.3.5 Karakteristik Performa Sistem....26
3.4 Performance Validation Concept Exploration....27
3.4.1 Tujuan Validasi Kebutuhan Performa....27
3.4.2 Proses Integrasi Karakteristik Performa....27
3.4.3 Validasi Karakteristik Performa....27
3.4.4 Dokumentasi Kebutuhan....28
CONCEPT DEFINITION....29
3.5 Performance Requirements Analysis....29
3.5.1 Tujuan dan Ruang Lingkup Analisis....29
3.5.2 Kebutuhan Performa....29
3.5.3 Kebutuhan Kompatibilitas, RMA, dan Lingkungan....29
3.5.4 Analisis Siklus Hidup Sistem....30
3.5.5 Kebutuhan Affordability dan Lifespan....30
3.5.6 Peran Sistem Predecessor....31
3.5.7 Metode Penentuan Kebutuhan Pengguna....31
3.6 Functional Analysis and Formulation....32
3.6.1 Tujuan dan Metode Analisis Fungsional....32
3.6.2 Identifikasi Fungsi Utama dan Sub-Fungsi....33
3.6.3 Alokasi Fungsi: Hardware vs Software....33
3.6.4 Alokasi Fungsi Lanjutan: Kesesuaian & Interaksi....34
3.6.5 Analisis Trade-Off Fungsi....35
3.6.6 Formulasi Spesifikasi Fungsional Awal....35
3.6.7 Kuantifikasi Sementara Spesifikasi Fungsional....36
3.7 Concept Selection....36
3.7.1 Identifikasi dan Deskripsi Alternatif Konsep....36
3.7.2 Matriks Evaluasi Alternatif (Weighted Scoring)....37
3.7.3 Penilaian Performa, Biaya, Jadwal, dan Risiko....37
3.7.4 Penentuan Bobot dan Peringkat....37
3.7.5 Finalisasi dan Justifikasi Konsep Terpilih....37
3.8 Concept Validation....38
3.8.1 Tujuan dan Ruang Lingkup Analisis....38

3.8.2 Pemodelan Sistem dan Lingkungan..... 38
3.8.3 Analisis Hasil dan Validasi.....40
3.8.4 Iterasi Konsep Sistem dan Revisi Spesifikasi..... 41
3.8.5 Kesimpulan Concept Validation.....42
4. ADVANCED DEVELOPMENT..... 43
4.1 Requirement Analysis..... 43
4.1.1 System Functional Specification.....43
4.1.2 Requirement Derivation..... 44
4.1.3 Relationship with Operational Requirements..... 45
4.2 Functional Analysis and Design.....45
4.2.1 Critical Characteristic of System Functional Elements.....46
4.3 Functional Design..... 46
4.3.1 System Interaction.....46
4.4 Prototype Development.....47
4.4.1 Mobile-App Design.....47
4.4.2 Mobile-App Use Case..... 47
4.4.3 Website Design.....47
4.4.4 Website Use Case..... 48
4.4.5 IoT Design.....48
4.4.6 IoT Use Case..... 48
4.4.7 IoT Circuit Design.....49
4.5 Development Testing.....49
4.5.1 Test Planning..... 50
4.5.2 Test Scenario..... 50
5. ENGINEERING DESIGN.....52
5.1 Pendahuluan dan Ruang Lingkup..... 52
5.1.1 Tujuan Pengembangan Sistem..... 52
5.1.2 Pengguna Utama dan Kebutuhan Spesifik..... 52
5.1.3 Nilai Tambah Sistem Dibandingkan Sistem Konvensional..... 52
5.1.4 Batasan Ruang Lingkup Proyek.....52
5.1.5 Asumsi Pengembangan Sistem..... 52
5.2 Engineering Development Lifecycle.....53
5.2.1 Concept Development..... 53
5.2.2 Preliminary Design.....54
5.3 Detailed Design..... 55
5.3.1 Spesifikasi Teknis Komponen Hardware..... 55
5.3.2 Arsitektur Software Frontend, Backend, dan Subsistem Lain..... 55
5.3.3 Skema Database Sistem..... 56
5.3.4 Rancangan Komunikasi Antar Komponen.....61
5.3.5 Protokol Komunikasi.....62
5.4 Production & Deployment..... 62

5.4.1 Rencana Pengadaan Komponen Hardware dan Lisensi Software.....62
5.4.2 Strategi Integrasi dan Pengujian Subsistem..... 62
5.4.3 Rencana Deployment dari Prototipe ke Implementasi Penuh.....62
5.4.4 Lingkungan Pengujian Sebelum Deployment.....63
5.4.5 Rencana Menjamin Kualitas Manufaktur dan Assembly Komponen.....63
5.5 Operations & Support.....63
5.5.1 Mekanisme Monitoring dan Pemeliharaan Sistem Pasca Implementasi.....63
5.5.2 Rencana Upgrade Sistem di Masa Depan..... 63
5.5.3 Strategi Menghadapi Kegagalan Sistem..... 63
5.5.4 Dokumentasi Teknis untuk Dukungan Operasional.....63
5.5.5 Pelatihan Pengguna Sistem..... 64
5.6 Requirement Analysis..... 64
5.6.1 Kebutuhan Fungsional.....64
5.6.2 Mekanisme Penentuan Logika Utama Sistem.....64
5.6.3 Parameter yang Diukur/Diolah.....65
5.6.4 Fitur Utama pada Frontend, Backend, dan Komponen Lain.....65
5.6.5 Penanganan Kondisi Abnormal.....65
5.6.6 Kebutuhan Non-Fungsional..... 65
5.7 Risk Assessment.....66
5.7.1 Identifikasi Risiko Teknis Utama.....66
5.7.2 Risiko Keandalan Komponen.....66
5.7.3 Risiko Keamanan..... 66
5.7.4 Risiko Integrasi Antar Subsistem.....66
5.7.5 Risiko Lingkungan Operasional.....67
5.7.6 Assessment Risiko.....67
5.7.7 Dampak Kegagalan Komponen dan Sistem.....67
5.7.8 Strategi Mitigasi Risiko.....67
5.7.9 Residual Risk Management.....68
5.8 Configuration Management & Change Control.....68
5.8.1 Version Control..... 68
5.8.2 Change Requests..... 68
5.8.3 Impact Analysis.....69
5.8.4 Approval & Documentation..... 69
5.9 Project Management.....69
5.9.1 Timeline & Milestones.....69
5.9.2 Resource Allocation..... 69
5.9.3 Quality Assurance..... 70
5.9.4 Communication..... 70
6. INTEGRATION AND EVALUATION..... 71
6.1 Perencanaan dan Persiapan Pengujian (Test Planning and Preparation).....71
6.2 Integrasi Sistem (System Integration)..... 72

6.2.1 Konfigurasi Pengujian Fisik....72
6.2.2 Proses Integrasi Sistem....73
6.2.3 Integrasi Total Sistem....73
6.2.4 Fasilitas Pengujian Integrasi Sistem....74
6.3 Pengujian Sistem dalam Pengembangan (Developmental System Testing)....74
6.3.1 Tujuan Pengujian Sistem dalam Pengembangan....74
6.3.2 Lingkup Pengujian....75
6.3.3 Rencana Skenario dan Validasi Teknis....75
6.3.4 Hasil Pengujian....75
6.3.5 Analisis dan Kesimpulan....76
6.4 Evaluasi Operasional (Operational Test & Evaluation)....76
6.4.1 Tujuan Evaluasi Operasional....76
6.4.2 Skenario Pengujian....77
6.4.3 Metode Pelaksanaan....77
6.4.4 Parameter Evaluasi....77
6.4.5 Hasil Observasi....78
6.4.6 Kesimpulan....87
DAFTAR PUSTAKA....89

DAFTAR TABEL

Tabel 1.1 Jadwal Kegiatan Pengembangan Log-Shield MVP (8 Minggu)..... 14
Tabel 1.2 Pembagian Tugas Anggota Kelompok 09..... 14
Tabel 2.1 Aspek Analisis Kebutuhan Petugas.....17
Tabel 2.2 Pemetaan Permasalahan dan Objektif..... 17
Tabel 2.3 Functional Requirements.....18
Tabel 2.4 Estimasi Biaya Pengembangan Sistem.....20
Tabel 2.5 Antisipasi Risiko..... 21
Tabel 2.6 Measures of Effectiveness (MoE)..... 22
Tabel 2.7 Measures of Performance (MoP).....23
Tabel 3.1 Penjelasan 5W1H..... 24
Tabel 3.2 Transformasi Kebutuhan Operasional ke Kebutuhan Kinerja.....25
Tabel 3.3 Formulasi Karakteristik Performa..... 27
Tabel 3.4 Skenario Simulasi dan Pengujian Sistem..... 29
Tabel 3.5 Kebutuhan Performansi Sistem..... 32
Tabel 3.6 Kebutuhan Affordability dan Lifespan.....33
Tabel 3.7 Klasifikasi Kebutuhan Pengguna..... 34
Tabel 3.8 Identifikasi Fungsi Utama dan Sub-Fungsi..... 36
Tabel 3.9 Alokasi Fungsi: Hardware vs Software.....36
Tabel 3.10 Alokasi Fungsi Lanjutan.....37
Tabel 3.11 Analisis Trade-Off Fungsi..... 38
Tabel 3.12 Formulasi Spesifikasi Fungsional Awal..... 38
Tabel 3.13 Kuantifikasi Sementara Spesifikasi Fungsional..... 39
Tabel 3.14 Identifikasi dan Deskripsi Alternatif Konsep..... 39
Tabel 3.15 Matriks Evaluasi Alternatif (Weighted Scoring)..... 40
Tabel 3.16 Penentuan Boundary System.....42
Tabel 3.17 Asumsi Lingkungan Operasional..... 42
Tabel 3.18 Simulasi Kinerja Sistem..... 42
Tabel 3.19 Parameter dan Kondisi Uji Eksperimen Kritis..... 43
Tabel 3.20 Klasifikasi Ketidaksesuaian dan Akar Masalah..... 43
Tabel 3.21 Revisi dan Iterasi Desain..... 44
Tabel 4.1 Spesifikasi Fungsional Sistem Log-Shield.....46
Tabel 4.2 Requirement Derivation for System Log-Shield.....47
Tabel 4.3 Requirement Derivation for Website and Web Service.....47
Tabel 4.4 Requirement Derivation for Mobile App..... 48
Tabel 4.5 Requirement Derivation for IoT.....48
Tabel 4.6 Karakteristik Kritis pada Fungsi Elemen Sistem Log-Shield.....49
Tabel 4.7 Skenario Pengujian Sistem Log-Shield..... 53
Tabel 5.1 Spesifikasi Hardware Node IoT..... 58
Tabel 5.2 Arsitektur Software per Lapisan.....59

Tabel 5.3 Entity Diagram users..... 60
Tabel 5.4 Entity Diagram poskos..... 60
Tabel 5.5 Entity Diagram distributions..... 61
Tabel 5.6 Entity Diagram stock_readings..... 62
Tabel 5.7 Entity Diagram predictions.....62
Tabel 5.8 Entity Diagram requests..... 63
Tabel 5.9 Entity Diagram assets.....63
Tabel 5.10 Entity Diagram audit_logs.....64
Tabel 5.11 Strategi Deployment per Komponen..... 65
Tabel 5.12 Kebutuhan Fungsional Sistem.....67
Tabel 5.13 Fitur Utama per Komponen.....68
Tabel 5.14 Risk Assessment.....70
Tabel 5.15 Timeline Proyek..... 72
Tabel 5.16 Alokasi Sumber Daya Tim..... 73
Tabel 6.1 Pengembangan Sistem dan Perencanaan T&E.....74
Tabel 6.2 Detail Konfigurasi Pengujian Fisik..... 75
Tabel 6.3 Detail Subsystem Test Configuration.....76
Tabel 6.4 Analisis Konteks Pengintegrasian Sistem secara Total..... 76
Tabel 6.5 Analisis Fasilitas Pengujian.....77
Tabel 6.6 Hasil Pengujian.....78
Tabel 6.7 Parameter Evaluasi..... 80
Tabel 6.8 Tampilan Website..... 81
Tabel 6.9 Tampilan Mobile App.....85

1. PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara dengan tingkat kerentanan bencana alam tertinggi di dunia. Posisi geografis Indonesia yang berada di Cincin Api Pasifik (Ring of Fire) dan pertemuan tiga lempeng tektonik besar menjadikannya sangat rentan terhadap gempa bumi, tsunami, letusan gunung berapi, banjir, dan tanah longsor. Data Badan Nasional Penanggulangan Bencana (BNPB) mencatat rata-rata lebih dari 2.000 kejadian bencana per tahun dalam satu dekade terakhir, yang mengakibatkan jutaan warga terpaksa mengungsi dan membutuhkan bantuan logistik darurat.

Namun, di balik besarnya skala kebutuhan tersebut, sistem manajemen logistik kemanusiaan di Indonesia masih menghadapi tantangan fundamental yang belum terselesaikan. Sebagian besar proses pencatatan distribusi bantuan di tingkat lapangan masih dilakukan secara manual menggunakan formulir kertas atau spreadsheet yang disinkronisasi secara berkala. Pendekatan ini sangat rentan terhadap kesalahan manusia, keterlambatan informasi, dan inkonsistensi data antara gudang pusat dan posko pengungsian.

Studi yang dilakukan oleh Sphere Project (2018) dan laporan evaluasi BNPB pasca-bencana gempa Cianjur (2022) mengidentifikasi beberapa pola kegagalan berulang dalam manajemen logistik bencana: (1) duplikasi distribusi bantuan ke posko yang sama akibat tidak adanya sistem pelacakan real-time; (2) kekurangan stok di posko terpencil akibat prediksi kebutuhan yang tidak akurat; (3) pemborosan komoditas perishable akibat distribusi tidak tepat waktu; dan (4) ketidakmampuan sistem mencatat kebutuhan khusus kelompok rentan seperti balita, lansia, ibu hamil, dan penyandang disabilitas.

Mandat BNPB melalui Peraturan Kepala BNPB No. 04/2017 tentang Sistem Komando Penanganan Darurat Bencana mengamanatkan prinsip Enam Tepat dalam penyaluran bantuan logistik, yaitu Tepat Waktu, Tepat Lokasi, Tepat Sasaran, Tepat Kualitas, Tepat Kuantitas, dan Tepat Kebutuhan. Namun, pemenuhan keenam dimensi ketepatan ini secara bersamaan tidak mungkin dicapai tanpa dukungan sistem informasi yang terintegrasi, real-time, dan berbasis data.

Untuk menjawab tantangan tersebut, Kelompok 09 mengusulkan pengembangan Log-Shield, sebuah sistem intelijen logistik kemanusiaan yang mengintegrasikan kecerdasan buatan (Artificial Intelligence), Internet of Things (IoT), dan arsitektur mobile Offline-First. Log-Shield dirancang untuk beroperasi bahkan di lingkungan dengan konektivitas internet yang sangat terbatas, sebagaimana sering ditemui di area terdampak bencana.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan utama yang menjadi fokus pengembangan Log-Shield dapat diidentifikasi sebagai berikut:

- Ketidakakuratan data stok logistik akibat pencatatan manual yang rawan kesalahan dan tidak dapat diperbarui secara real-time, mengakibatkan distribusi yang tidak tepat kuantitas.
- Ketiadaan sistem prediksi berbasis data untuk mengantisipasi kebutuhan logistik di setiap posko secara proaktif, sehingga respons distribusi selalu bersifat reaktif dan sering terlambat.
- Ketergantungan penuh pada konektivitas internet yang tidak dapat diandalkan di area bencana, menyebabkan petugas lapangan tidak dapat mengakses atau memperbarui data saat koneksi terputus.
- Tidak adanya mekanisme perlindungan data pribadi pengungsi yang sesuai dengan UU PDP No. 27 Tahun 2022, khususnya untuk data kelompok rentan (anak, lansia, data kesehatan).
- Fragmentasi data logistik antar lembaga akibat tidak adanya standar kode identifikasi bencana yang seragam, menyebabkan kesulitan rekonsiliasi dan audit pasca-bencana.
- Antarmuka sistem yang ada tidak mempertimbangkan kondisi psikologis dan tekanan kerja petugas lapangan di masa bencana, mengakibatkan tingkat kesalahan input yang tinggi.

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Tujuan pengembangan sistem Log-Shield adalah:

- Membangun platform terintegrasi yang memungkinkan pencatatan dan sinkronisasi data logistik bencana secara real-time maupun offline dengan akurasi tinggi.
- Mengimplementasikan model kecerdasan buatan berbasis Time Series Forecasting yang mampu memprediksi kebutuhan logistik harian per posko dengan akurasi lebih dari 88%.
- Menyediakan sistem validasi stok fisik otomatis menggunakan sensor IoT Load Cell yang dapat beroperasi di lingkungan bencana ekstrem.
- Membangun aplikasi mobile dengan arsitektur Offline-First yang memungkinkan operasional penuh selama minimal 72 jam tanpa koneksi internet, mendukung platform Android dan iOS.
- Memastikan keamanan dan kepatuhan pengelolaan data pribadi pengungsi sesuai UU PDP No. 27 Tahun 2022.

- Menggunakan KIB 16-digit sebagai standar identifikasi data untuk memastikan interoperabilitas dengan sistem Satu Data Indonesia.

1.3.2 Manfaat

Manfaat yang diharapkan dari pengembangan Log-Shield meliputi:

- Bagi BNPB/BPBD: Peningkatan efisiensi manajemen logistik bencana, reduksi pemborosan bantuan, dan kemudahan rekonsiliasi data lintas wilayah.
- Bagi Petugas Lapangan: Antarmuka yang mudah digunakan dalam kondisi tekanan tinggi, kemampuan bekerja tanpa internet, dan transparansi rekomendasi distribusi berbasis AI.
- Bagi Pengungsi: Distribusi bantuan yang lebih akurat sesuai kebutuhan spesifik (termasuk kelompok rentan), tepat waktu, dan tepat kuantitas.
- Bagi Akademisi dan Pengembang: Referensi implementasi sistem rekayasa terpadu berbasis prinsip Kossiakoff untuk konteks kemanusiaan di Indonesia.

1.4 Ruang Lingkup

Pengembangan Log-Shield dalam kerangka Minimum Viable Product (MVP) selama 8 minggu mencakup ruang lingkup berikut:

- Pengembangan model AI Time Series Forecasting untuk prediksi kebutuhan logistik dan implementasi Explainable AI (XAI) menggunakan teknik SHAP.
- Prototyping perangkat IoT berbasis ESP32 dengan sensor Load Cell HX711 untuk pemantauan stok fisik, dilengkapi protokol MQTT over Wi-Fi dan rating ketahanan IP67.
- Pengembangan aplikasi mobile lintas platform menggunakan Capacitor.js (Android dan iOS) dengan arsitektur Offline-First berbasis SQLite, dan Trauma-Aware UI/UX.
- Pembangunan REST API backend menggunakan Node.js dan Express dengan integrasi Supabase (PostgreSQL managed) sebagai basis data terpusat, mekanisme custom sync dua arah, dan Row Level Security.
- Implementasi standar KIB 16-digit sebagai kunci identifikasi data untuk sinkronisasi dengan ekosistem Satu Data Indonesia.

Pada pengembangan fase ini tidak mencakup integrasi penuh dengan sistem eksisting BNPB, pengembangan modul manajemen keuangan bantuan, fitur analitik jangka panjang pasca-bencana, serta deployment ke app store publik. Komponen-komponen tersebut direncanakan untuk pengembangan fase selanjutnya.

1.5 Jadwal Kegiatan dan Pembagian Tugas

1.5.1 Jadwal Kegiatan

Tabel 1.1 Jadwal Kegiatan Pengembangan Log-Shield MVP (8 Minggu)

Kegiatan	PIC	W1	W2	W3	W4	W5	W6	W7	W8
Needs Analysis	PM, AI, IoT, Mobile, BE	✓	✓						
Concept Exploration	PM, AI, IoT		✓	✓					
Concept Definition	Semua anggota			✓					
Desain Arsitektur Sistem	AI, IoT, Mobile, BE			✓	✓				
Prototyping IoT & Sensor	IoT Engineer				✓	✓			
Pengembangan AI Model	AI Scientist				✓	✓			
Pengembangan Mobile App	Mobile Developer				✓	✓	✓		
Backend & Cloud Setup	Backend Engineer				✓	✓			
Integrasi & Pengujian	Semua anggota						✓	✓	
Demo MVP & Dokumentasi	PM, Semua anggota								✓

1.5.2 Pembagian Tugas

Tabel 1.2 Pembagian Tugas Anggota Kelompok 09

Peran	Nama Anggota	Tanggung Jawab Utama
Project Manager	Andhika M. Addiputra	Roadmap produk, kepatuhan UU PDP No. 27/2022, koordinasi BNPB/BPBD, standarisasi KIB 16-digit, juga bertanggung jawab atas Mobile Development bersama Naufal.
AI & Data Scientist	Ferro Arka Berlian	Model Time Series Forecasting menggunakan TinyTimeMixers, Explainable AI (SHAP), pembuatan Rationale Chips, setup FastAPI dan retraining harian otomatis.
IoT & Edge Engineer	Luckman Fakhmanidris A.	Prototyping ESP32 + HX711 + Load Cell, firmware median filtering, buffer NVS, MQTT QoS 1, OTA update, kalibrasi sensor, sertifikasi IP67.
Mobile Developer	M. Naufal Fathan	Aplikasi Capacitor.js + React untuk Android dan iOS, integrasi SQLite lokal, halaman pencatatan distribusi, prediksi AI, laporan anomali, sync logic.
Backend & Cloud Engineer	Atharizza M. Athaya	REST API Express, integrasi Supabase Client dan RLS, MQTT subscriber, enkripsi AES-256, custom sync endpoint, deployment DigitalOcean.

1.6 Sistematika Penulisan

Laporan Deliverable 4 ini disusun dengan sistematika penulisan sebagai berikut:

Bab 1 — Pendahuluan menguraikan latar belakang masalah logistik bencana di Indonesia, identifikasi masalah, tujuan dan manfaat pengembangan Log-Shield, ruang lingkup MVP, jadwal kegiatan, pembagian tugas, dan sistematika penulisan.

Bab 2 — Needs Analysis membahas analisis kebutuhan sistem secara komprehensif mencakup analisis operasional, analisis fungsional, analisis kelayakan, dan validasi kebutuhan mengacu pada kerangka kerja Kossiakoff.

Bab 3 — Concept Exploration dan Concept Definition mengeksplorasi konsep implementasi sistem, mencakup analisis kebutuhan operasional, formulasi kebutuhan performa, eksplorasi konsep implementasi, validasi performa, dan pemilihan konsep terbaik melalui weighted scoring matrix.

Bab 4 — Advanced Development memaparkan pengembangan lanjutan sistem terpilih, mencakup analisis persyaratan terperinci, desain fungsional, pengembangan prototipe Mobile App, Website, dan IoT, serta rencana pengujian pengembangan.

Bab 5 — Engineering Design menjabarkan desain rekayasa terperinci sistem, mencakup spesifikasi hardware IoT, arsitektur software lima lapisan, skema database PostgreSQL Supabase, protokol komunikasi MQTT dan REST, strategi deployment pada DigitalOcean, analisis kebutuhan lanjutan, asesmen risiko, configuration management, dan manajemen proyek.

Bab 6 — Integration and Evaluation menjelaskan rencana integrasi dan evaluasi sistem secara terstruktur dalam empat tingkat: perencanaan T&E, integrasi sistem bertahap, pengujian developmental, dan evaluasi operasional dengan pengguna nyata.

2. NEEDS ANALYSIS

Bagian ini menjelaskan analisis kebutuhan sistem Log-Shield secara menyeluruh, mencakup analisis operasional, fungsional, kelayakan, dan validasi kebutuhan. Analisis dilakukan mengacu pada kerangka kerja Systems Engineering yang dikembangkan oleh Kossiakoff, Sweet, Seymour, dan Biemer (2011).

2.1 Operations Analysis

Operations analysis dilakukan untuk memahami kebutuhan operasional yang mendasari pengembangan Log-Shield. Analisis ini bertujuan mengidentifikasi dan menilai kekurangan dari sistem yang ada, serta mengevaluasi potensi perbaikan melalui integrasi teknologi AI, IoT, dan mobile.

2.1.1 Analisis Kebutuhan & Identifikasi Kekurangan

Sistem manajemen logistik bencana yang saat ini berjalan di Indonesia menghadapi lima kekurangan sistemik yang terdokumentasi:

Pertama, ketergantungan pada pencatatan manual. Sebagian besar operasi BPBD di tingkat kabupaten masih menggunakan formulir kertas yang direkap ke spreadsheet setiap beberapa jam sekali. Proses ini menciptakan jeda informasi (information lag) yang signifikan antara kondisi stok aktual di gudang dengan data yang tersedia untuk pengambilan keputusan distribusi.

Kedua, tidak adanya prediksi berbasis data. Tanpa model prediktif, petugas koordinasi hanya dapat merespons permintaan yang masuk secara reaktif. Penelitian oleh Balcik dan Beamon (2008) dalam Journal of Operations Management menunjukkan bahwa sistem manajemen logistik kemanusiaan reaktif memiliki efisiensi distribusi 40–60% lebih rendah dibandingkan sistem yang menggunakan pendekatan prediktif berbasis data historis.

Ketiga, fragmentasi data lintas lembaga. Setiap institusi (BNPB, BPBD, PMI, dan lembaga kemanusiaan lainnya) menggunakan format data dan kode identifikasi yang berbeda-beda, sehingga rekonsiliasi data pasca-bencana memakan waktu hingga berminggu-minggu.

Keempat, tidak ada perlindungan data yang memadai. Data pengungsi, termasuk informasi kesehatan, kondisi disabilitas, dan komposisi keluarga yang dikumpulkan selama bencana, seringkali tidak dikelola dengan protokol keamanan yang sesuai standar UU PDP No. 27 Tahun 2022.

Kelima, antarmuka yang tidak ramah pengguna dalam kondisi darurat. Sistem digital yang pernah dicoba di beberapa daerah mengalami tingkat kegagalan adopsi yang tinggi

karena antarmuka yang kompleks dan tidak mempertimbangkan kondisi stres pasca-bencana yang dialami petugas lapangan.

2.1.2 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan melalui pendekatan User Persona dan benchmarking terhadap sistem logistik kemanusiaan yang telah berhasil diimplementasikan di negara lain (HELIOS USAID di Nigeria, OpenLMIS di Tanzania).

Tabel 2.1 Aspek Analisis Kebutuhan Petugas

Aspek Survei	Detail
Jumlah Responden	Dua segmen utama: Petugas Lapangan (Primary User) dan Koordinator Logistik (Secondary User), divalidasi melalui benchmarking sistem serupa di luar negeri.
Metode Survei	Secondary Research dan User Persona berbasis laporan evaluasi BNPB pasca-bencana, panduan operasional Sphere Project (2018), dan dokumentasi sistem HELIOS USAID serta OpenLMIS.

Dua segmen pengguna utama diidentifikasi:

- **Petugas Lapangan (Primary User):** Bertugas di posko pengungsian dan gudang logistik. Karakteristik kunci: tingkat stres tinggi, terbatasnya waktu untuk input data, sering beroperasi tanpa sinyal internet, dan tidak selalu memiliki latar belakang teknologi informasi. Kebutuhan utama: antarmuka yang sangat sederhana (maksimal 3 langkah input), kemampuan offline penuh, dan notifikasi yang jelas dan tidak ambigu.
- **Koordinator Logistik (Secondary User):** Bertugas di tingkat kabupaten atau provinsi. Karakteristik kunci: membutuhkan gambaran menyeluruh stok dan prediksi kebutuhan, perlu melakukan rekonsiliasi data dari banyak posko, dan bertanggung jawab atas pelaporan ke BNPB. Kebutuhan utama: dashboard real-time, visualisasi prediksi AI dengan penjelasan yang dapat dipahami (Rationale Chips), dan laporan terstandarisasi menggunakan KIB 16-digit.

2.1.3 Penetapan Tujuan Operasional

Tabel 2.2 Pemetaan Permasalahan dan Objektif

Kode	Objektif	Permasalahan
O01	Meningkatkan akurasi distribusi logistik bencana	Distribusi logistik bencana masih berbasis estimasi manual sehingga terjadi kekurangan atau penumpukan stok di titik tertentu.
O02	Mempercepat waktu respons penyaluran bantuan	Proses pencatatan manual menyebabkan keterlambatan rata-rata 48–72 jam dalam sinkronisasi data antara gudang pusat dan posko lapangan.

Kode	Objektif	Permasalahan
O03	Menjamin validasi stok secara real-time	Tidak ada mekanisme otomatis untuk memantau sisa stok fisik, sehingga data stok sering tidak sesuai dengan kondisi aktual di gudang.
O04	Memastikan kepatuhan data pribadi pengungsi	Pengelolaan data pengungsi (termasuk lansia, anak, dan data kesehatan) tidak mematuhi standar perlindungan data pribadi yang berlaku (UU PDP No. 27/2022).
O05	Mengintegrasikan sistem dengan standar Satu Data Indonesia	Sistem logistik eksisting tidak menggunakan format KIB (Kode Identitas Bencana) 16-digit sehingga menyulitkan rekonsiliasi data lintas lembaga.
O06	Memungkinkan operasi di area dengan koneksi internet terbatas	Petugas lapangan tidak dapat mengakses atau memperbarui data saat koneksi internet terputus, yang umum terjadi di lokasi bencana.

Keenam objektif di atas membentuk kerangka tujuan operasional yang menjadi landasan seluruh keputusan desain sistem Log-Shield. Objektif O01 hingga O03 berkaitan dengan peningkatan efisiensi operasional, O04 dan O05 berkaitan dengan kepatuhan regulasi dan standarisasi, sedangkan O06 berkaitan dengan resiliensi sistem di lingkungan bencana yang tidak ideal. Objectives Tree sistem Log-Shield divisualisasikan pada Gambar 1.1 yang akan disisipkan pada saat finalisasi.

2.2 Functional Analysis

2.2.1 Functional Requirements

Functional requirements merujuk pada spesifikasi tentang fungsi atau fitur yang harus dimiliki oleh sistem.

Tabel 2.3 Functional Requirements

Kode	Deskripsi Kebutuhan Fungsional	Sub-Sistem Terkait
FR01	Sistem harus mampu memproses data historis logistik dan memprediksi volume kebutuhan harian per posko menggunakan model Time Series Forecasting	Sub-sistem AI/Prediksi
FR02	Sistem harus menampilkan penjelasan hasil prediksi AI dalam format narasi sederhana (Rationale Chips) yang dapat dipahami petugas non-teknis	Sub-sistem AI/Prediksi
FR03	Sistem harus secara otomatis memantau sisa stok logistik fisik menggunakan sensor Load Cell yang terpasang di palet gudang	Sub-sistem IoT/Sensor
FR04	Sistem harus dapat beroperasi di lingkungan bencana ekstrem dengan rating ketahanan minimum IP67 dan mampu bertransmisi data via MQTT/Wi-Fi	Sub-sistem IoT/Sensor
FR05	Aplikasi mobile harus tetap berfungsi penuh secara offline menggunakan SQLite sebagai Single Source of Truth, dan melakukan sinkronisasi otomatis saat koneksi pulih	Sub-sistem Aplikasi Mobile

Kode	Deskripsi Kebutuhan Fungsional	Sub-Sistem Terkait
FR06	Antarmuka pengguna aplikasi mobile harus dirancang dengan prinsip Trauma-Aware UI/UX, membatasi alur input maksimal 3 langkah	Sub-sistem Aplikasi Mobile
FR07	Backend harus menyediakan API yang mendukung sinkronisasi data dua arah dengan strategi resolusi konflik Last Write Wins (LWW)	Sub-sistem Backend/Cloud
FR08	Sistem harus menerapkan enkripsi data at-rest dan in-transit, serta kontrol akses berbasis peran (RBAC) sesuai mandat UU PDP	Sub-sistem Backend/Cloud
FR09	Setiap transaksi data logistik harus menggunakan Kode Identitas Bencana (KIB) 16-digit sebagai kunci primer	Seluruh Sub-sistem
FR10	Sistem harus mendukung profil kebutuhan khusus (lansia, balita, ibu hamil, disabilitas) dalam perhitungan prediksi dan distribusi logistik	Sub-sistem AI/Prediksi & Aplikasi

2.2.2 Proses Trade-Off

Dalam proses identifikasi functional requirements, beberapa trade-off kritis dihadapi dan diselesaikan sebagai berikut:

Akurasi vs Latensi AI: Model AI yang lebih kompleks dapat mencapai akurasi prediksi yang lebih tinggi, namun membutuhkan waktu inferensi lebih lama dan sumber daya komputasi yang lebih besar. Diputuskan menggunakan model TinyTimeMixers yang ringan dengan target MAE < 15%, mengikuti prinsip Edge AI yang diprioritaskan untuk efisiensi operasional.

Cakupan Sensor vs Biaya Deployment: Sensor Load Cell di seluruh palet gudang memberikan visibilitas stok yang sempurna, namun biaya hardware dan instalasi menjadi sangat tinggi. Untuk MVP, diputuskan bahwa sensor ditempatkan hanya pada titik-titik stok strategis (komoditas prioritas tinggi: beras, air, dan paket kesehatan), sehingga cakupan 80% stok kritis dapat dicapai dengan 30% dari biaya deployment penuh.

Kapasitas Offline vs Kebutuhan Penyimpanan: Penyimpanan data offline yang lebih lama meningkatkan resiliensi petugas lapangan, namun meningkatkan kebutuhan storage perangkat mobile dan potensi konflik data saat sinkronisasi. Ditetapkan batas 72 jam data operasional offline sebagai keseimbangan optimal.

2.3 Feasibility Analysis

2.3.1 Proses Pengembangan

Pengembangan Log-Shield mengikuti metodologi Agile Sprint dengan timeline 8 minggu yang dibagi menjadi empat fase:

- Fase Initiating (Minggu 1): Finalisasi requirements, setup infrastruktur development, pengadaan komponen hardware IoT (ESP32, Load Cell HX711).
- Fase Design & Planning (Minggu 2–3): Desain arsitektur sistem, skema database, wireframe UI/UX aplikasi mobile, spesifikasi API.
- Fase Development (Minggu 4–6): Parallel development AI model training, prototyping IoT, pengembangan aplikasi mobile, dan setup backend cloud.
- Fase Testing & Integration (Minggu 7–8): Integrasi komponen, pengujian end-to-end, bug fixing, dan persiapan demo MVP.

2.3.2 Strategi Pengembangan

Pengembangan berlangsung secara paralel di lima domain keahlian. AI Scientist mulai melatih model Time Series Forecasting (TinyTimeMixers) menggunakan dataset simulasi sambil membangun pipeline SHAP. IoT Engineer merakit node sensor, memvalidasi akurasi pembacaan Load Cell, dan menguji transmisi data via MQTT pada jarak bertahap. Mobile Developer membangun kerangka aplikasi Capacitor.js untuk Android dan iOS dengan arsitektur SQLite dan menguji skenario offline-sync. Backend Engineer membangun struktur API Express dan mengonfigurasi Supabase dengan Row Level Security. Project Manager memantau progress sprint mingguan, mengelola ketergantungan antar komponen, dan memastikan seluruh keputusan teknis tetap mematuhi UU PDP No. 27/2022.

2.3.3 Estimasi Biaya

Tabel 2.4 Estimasi Biaya Pengembangan Sistem

Komponen	Jumlah	Harga Satuan (Rp)	Total (Rp)
ESP32 Modul	3 unit	185.000	555.000
Sensor Load Cell 50kg + Modul HX711	3 set	75.000	225.000
Casing Waterproof IP67	3 unit	95.000	285.000
Kabel, Jumper, PCB Protoboard	1 set	85.000	85.000
Power Bank 20.000 mAh	2 unit	250.000	500.000
Domain + SSL Certificate (1 tahun)	1 paket	180.000	180.000
DigitalOcean Droplet (2 bulan) + Supabase Free Tier	1 paket	350.000	350.000
Biaya Pengujian & Transportasi	1 paket	200.000	200.000
TOTAL			Rp 2.380.000

2.3.4 Pendekatan Desain dan Metode Evaluasi

Pengembangan Log-Shield mengadopsi tiga pendekatan desain yang diterapkan secara komplementer. Pertama, User-Centered Design (UCD) yang diterapkan pada komponen aplikasi mobile melalui prinsip Trauma-Aware UX dengan alur input maksimal tiga langkah, tipografi besar, dan kontras tinggi. Kedua, Modular Design pada seluruh lapisan sistem sehingga setiap sub-sistem dapat diuji dan ditingkatkan independen. Ketiga, Security by Design dengan integrasi AES-256, Row Level Security, dan audit trail sejak tahap perancangan.

Metode evaluasi mencakup empat tingkatan: unit testing pada fungsi kritis, integration testing untuk alur data antar sub-sistem, user acceptance testing (UAT) dengan minimal lima pengguna representatif, serta analisis data real-time melalui dashboard monitoring DigitalOcean dan Supabase.

2.3.5 Isu Produksi dan Konsep Operasional

Dari sisi ketersediaan komponen hardware, ESP32 dan sensor Load Cell HX711 tersedia di pasaran lokal daring (Tokopedia/Shopee), dengan waktu pengiriman 3–7 hari kerja untuk wilayah Pulau Jawa. Risiko dimitigasi dengan pemesanan di awal masa pengembangan dan menyiapkan satu unit cadangan per jenis komponen. Proses kalibrasi sensor memerlukan waktu 2–3 jam per unit menggunakan beban referensi terkalibrasi.

Konsep operasional harian Log-Shield berjalan dalam siklus 24 jam: pukul 00.00–05.59 WIB sistem backend menjalankan batch agregasi data dan pelatihan ulang AI; pukul 06.00 WIB model AI menghasilkan prediksi harian; pukul 06.00–21.00 WIB sistem beroperasi aktif dengan sensor IoT memperbarui data setiap 30 detik; pukul 21.00–23.59 WIB sistem memasuki mode pemeliharaan ringan dengan sinkronisasi akhir hari dan health check seluruh node IoT.

2.3.6 Antisipasi Risiko

Tabel 2.5 Antisipasi Risiko

Jenis Risiko	Deskripsi	Potensi Dampak	Mitigasi
Teknis	Kegagalan sensor Load Cell di lingkungan ekstrem	Data stok tidak terbaca, distribusi berbasis estimasi	Sensor IP67; redundansi cadangan; fallback input manual
Teknis	Model AI tidak akurat karena data terbatas	Distribusi tidak tepat kuantitas	Fallback rata-rata historis 30 hari; SHAP transparansi
Operasional	Petugas menolak adopsi sistem	Tingkat adopsi rendah	Trauma-Aware UI 3 langkah; pelatihan 30 menit; offline support
Operasional	Konflik data sinkronisasi bersamaan	Data stok inkonsisten	LWW timestamp hybrid; audit log; resolusi manual
Keamanan	Kebocoran data pribadi pengungsi	Pelanggaran UU PDP	AES-256 + TLS 1.3 + RLS + audit trail

Jenis Risiko	Deskripsi	Potensi Dampak	Mitigasi
Jadwal	Pengembangan 8 minggu melampaui batas	MVP terlambat	Agile sprint ketat; buffer 1 minggu; MoSCoW prioritas

2.4 Needs Validation

2.4.1 Pentingnya Validasi Kebutuhan

Validasi kebutuhan merupakan proses kritis dalam Systems Engineering untuk memastikan bahwa requirements yang telah diidentifikasi benar-benar mencerminkan kebutuhan operasional aktual pengguna dan pemangku kepentingan, bukan hanya asumsi tim pengembang. Dalam konteks Log-Shield, validasi kebutuhan sangat penting karena sistem akan digunakan dalam kondisi darurat yang tidak dapat disimulasikan sepenuhnya, pengguna primer memiliki karakteristik unik, dan kegagalan sistem saat bencana dapat berakibat pada hilangnya nyawa manusia.

2.4.2 Model Efektivitas Operasional

Model efektivitas operasional Log-Shield distrukturkan dalam tiga lapisan: kondisi lingkungan operasional, respons sistem, dan pengukuran hasil. Empat skenario operasional dievaluasi:

Skenario 1 — Respons Awal 0–72 Jam: Skenario paling kritis dengan koneksi internet terputus total. Sistem dievaluasi pada kemampuan beroperasi offline penuh, mencatat transaksi dengan KIB, dan sinkronisasi otomatis saat koneksi pulih.

Skenario 2 — Operasi Rutin Hari 3-30: Koneksi internet intermittent, model AI memiliki data historis cukup. Efektivitas dievaluasi dari akurasi prediksi AI yang meningkat dan kebermanfaatan Rationale Chips untuk pengambilan keputusan.

Skenario 3 — Kondisi Stok Kritis: Komoditas mendekati habis dengan jalur pasokan terhambat. Sistem dievaluasi pada kecepatan notifikasi darurat dan akurasi rekomendasi redistribusi AI.

Skenario 4 — Bencana Skala Besar Multi-Posko: Lonjakan pengungsi mendadak, koordinasi lintas posko. Sistem dievaluasi pada kemampuan scaling backend dan konsistensi data lintas perangkat.

2.4.3 Measures of Effectiveness (MoE)

MoE dirancang untuk mengukur sejauh mana sistem Log-Shield berhasil memenuhi mandat Enam Tepat BNPB.

Tabel 2.6 Measures of Effectiveness (MoE)

Kode	Indikator Efektivitas	Formula Pengukuran	Target
MoE1	Akurasi Prediksi Kebutuhan Logistik	(1 - |Prediksi - Aktual| / Aktual) × 100%	> 88%
MoE2	Ketepatan Distribusi Bantuan (Enam Tepat)	Distribusi memenuhi ≥5 kriteria Tepat / Total	> 90%
MoE3	Ketersediaan Sistem (Uptime)	(Operasional - Henti) / Operasional × 100%	> 95%
MoE4	Kecepatan Respons Darurat	Waktu dari alert hingga distribusi	< 4 jam
MoE5	Tingkat Adopsi Petugas Lapangan	Petugas aktif / Total target	> 80% dalam 4 minggu
MoE6	Reduksi Pemborosan Logistik	(Stok terbuang sebelum - sesudah) / Sebelum	> 30%

2.4.4 Measures of Performance (MoP)

MoP berfungsi sebagai parameter teknis yang dapat diukur dan diverifikasi selama proses pengujian.

Tabel 2.7 Measures of Performance (MoP)

Kode	Indikator Kinerja	Metrik Pengukuran	Target
MoP1	Akurasi Sensor Load Cell	Deviasi berat dari nilai terkalibrasi	±150 gram
MoP2	Latensi Transmisi Data Sensor	Waktu sensor hingga simpan backend	< 10 detik
MoP3	Waktu Sinkronisasi Offline→Online	Durasi sync setelah koneksi pulih	< 60 detik
MoP4	Latensi Prediksi AI	Waktu komputasi model	< 5 detik
MoP5	Waktu Respons UI Aplikasi Mobile	Aksi pengguna hingga respons layar	< 2 detik
MoP6	Throughput API Backend	Request per detik kondisi normal	≥ 100 req/detik
MoP7	Konsumsi Daya Perangkat IoT	Konsumsi rata-rata transmisi aktif	< 500 mA @ 5V
MoP8	Akurasi Model Time Series (MAE)	Mean Absolute Error harian	< 15%

2.4.5 Piramida Analysis

Piramida analisis Log-Shield terdiri dari tiga tingkatan yang saling mendukung. Pada tingkat dasar terdapat Sumber Daya, mencakup lima anggota tim dengan keahlian terdiversifikasi (PM, AI Scientist, IoT Engineer, Mobile Developer, Backend Engineer), infrastruktur cloud DigitalOcean + Supabase, komponen sensor IoT, dan framework pengembangan open-source. Pada tingkat menengah terdapat Kapabilitas sistem yang dibangun dari sumber daya tersebut, yakni kemampuan prediksi AI, monitoring IoT real-time, dan manajemen data terenkripsi. Pada puncak piramida terdapat Dampak

Operasional yang diharapkan, yaitu pemenuhan Enam Tepat BNPB, peningkatan efisiensi distribusi logistik bencana, dan kepatuhan UU PDP No. 27 Tahun 2022.

3. CONCEPT EXPLORATION

Bagian ini mengeksplorasi konsep implementasi sistem Log-Shield, mencakup analisis kebutuhan operasional terstruktur, formulasi kebutuhan performa, eksplorasi alternatif konsep implementasi, dan validasi performa awal.

3.1 Operational Requirements Analysis

3.1.1 Tujuan dan Process of Requirements Analysis

Tujuan analisis persyaratan operasional Log-Shield adalah mengubah kebutuhan operasional yang bersifat naratif (mandat Enam Tepat BNPB, kepatuhan UU PDP, resiliensi di area bencana) menjadi persyaratan kinerja sistem yang terukur dan dapat diverifikasi. Proses analisis dilakukan dalam empat tahap berurutan: requirements elicitation menggunakan pendekatan 5W1H, requirements testing terhadap kriteria kelayakan, requirements validation melalui benchmarking dan tinjauan internal, serta dokumentasi requirements dalam format yang dapat ditelusuri.

3.1.2 Requirements Elicitation

Tabel berikut menjelaskan konteks operasional sistem menggunakan pendekatan 5W1H.

Tabel 3.1 Penjelasan 5W1H

Aspek	Penjelasan
Why	Sistem dikembangkan karena keterbatasan sistem logistik bencana eksisting yang masih manual, tidak dapat memenuhi mandat Enam Tepat BNPB secara simultan, dan tidak memiliki perlindungan data pribadi sesuai UU PDP No. 27/2022. Setiap bencana di Indonesia menelan korban jiwa tambahan akibat distribusi bantuan yang tidak tepat sasaran atau terlambat.
What	Sistem harus mampu melakukan empat fungsi utama: (1) prediksi kebutuhan logistik harian berbasis AI; (2) pemantauan stok fisik otomatis via sensor IoT; (3) pencatatan distribusi offline-first di mobile; (4) dashboard koordinasi real-time untuk koordinator BPBD.
Who	Pengguna primer adalah Petugas Lapangan BPBD di posko dan gudang. Pengguna sekunder adalah Koordinator Logistik di tingkat kabupaten/provinsi. Pengguna tersier adalah Administrator BNPB pusat yang mengelola akun dan parameter sistem.
Where	Sistem diimplementasikan di posko pengungsian, gudang logistik daerah, dan kantor koordinasi BPBD. Lokasi-lokasi ini umumnya memiliki koneksi internet yang tidak stabil, suhu tinggi, kelembaban tinggi, dan akses listrik intermittent.
When	Sistem aktif 24/7 selama fase respons bencana (umumnya 30–90 hari pasca-kejadian) dan tetap dipertahankan dalam mode standby untuk kesiapan bencana berikutnya. Periode kritis adalah 72 jam pertama setelah bencana ketika konektivitas paling buruk.
How	Sistem menggunakan arsitektur monorepo dengan komponen IoT (ESP32 + Load Cell), backend (Node.js + Supabase PostgreSQL), AI Engine (Python FastAPI + TinyTimeMixers), Mobile (Capacitor.js Android+iOS dengan SQLite), dan Web Dashboard (React). Komunikasi via MQTT (IoT) dan REST/WebSocket (klien).

3.1.3 Requirements Testing

Setiap requirement yang teridentifikasi diuji terhadap lima kriteria kelayakan: ketertelusuran (traceability) dari requirement ke objektif operasional dan permasalahan; bebas redundansi (no overlap antar requirement); dapat diverifikasi (verifiable) melalui pengujian objektif; kesesuaian teknis dengan stack teknologi yang dipilih; dan kelayakan biaya/jadwal dalam batasan MVP 8 minggu dengan anggaran Rp 2.380.000. Requirements yang tidak memenuhi salah satu kriteria direvisi atau diturunkan prioritasnya menggunakan metode MoSCoW (Must, Should, Could, Won't).

3.1.4 Requirements Validation

Validasi requirements Log-Shield dilakukan melalui empat mekanisme. Pertama, tinjauan internal antar anggota tim dengan checklist kriteria SMART (Specific, Measurable, Achievable, Relevant, Time-bound). Kedua, perbandingan dengan solusi existing internasional (HELIOS USAID, OpenLMIS) untuk memastikan requirements Log-Shield setidaknya setara atau lebih baik dalam aspek kemanusiaan. Ketiga, penilaian kelayakan teknis oleh masing-masing PIC sub-sistem terhadap stack teknologi yang dipilih. Keempat, evaluasi keselarasan dengan tujuan keberlanjutan dan kepatuhan regulasi Indonesia.

3.2 Performance Requirements Formulation

3.2.1 Transformation of Operational Requirements into Performance Requirements

Kebutuhan operasional Log-Shield ditransformasikan menjadi kebutuhan kinerja terukur melalui pemetaan langsung antara objektif operasional (O01–O06), MoE, dan MoP. Sebagai contoh, objektif O01 (Meningkatkan akurasi distribusi logistik) diturunkan menjadi MoE1 (Akurasi prediksi > 88%) yang kemudian diturunkan lagi menjadi MoP1 (Akurasi sensor ±150 gram) dan MoP8 (MAE model < 15%). Pendekatan ini memastikan setiap parameter teknis dapat ditelusuri kembali ke kebutuhan operasional asal.

Tabel 3.2 Transformasi Kebutuhan Operasional ke Kebutuhan Kinerja

Kebutuhan Operasional	Parameter Kinerja	Target	Sub-sistem
O01: Akurasi distribusi	Akurasi prediksi AI	MAE < 15%	AI Engine
O01: Akurasi distribusi	Akurasi sensor Load Cell	±150 gram	IoT
O02: Kecepatan respons	Latensi transmisi data	< 10 detik E2E	IoT + Backend
O02: Kecepatan respons	Waktu sinkronisasi	< 60 detik untuk 500 transaksi	Mobile + Backend

Kebutuhan Operasional	Parameter Kinerja	Target	Sub-sistem
O03: Validasi stok real-time	Interval pembacaan sensor	30 detik	IoT
O04: Kepatuhan UU PDP	Enkripsi at-rest dan in-transit	AES-256 + TLS 1.3	Backend
O05: Standar Satu Data	Format identifier data	KIB 16-digit pada semua entitas	Seluruh sub-sistem
O06: Operasi offline	Durasi offline maksimum	72 jam	Mobile

3.2.2 Derivation of Sub-system Functions

Berdasarkan kebutuhan kinerja, lima sub-sistem utama Log-Shield diturunkan beserta fungsi-fungsi spesifiknya.

- **Sub-sistem IoT/Sensor:** Membaca bobot stok via Load Cell HX711, menerapkan median filtering, mempublikasikan data via MQTT QoS 1, dan menyimpan buffer NVS saat offline.
- **Sub-sistem AI/Prediksi:** Melatih model Time Series Forecasting harian, menghasilkan prediksi 7 hari ke depan beserta Rationale Chips, dan memberikan fallback rata-rata historis jika MAE > 20%.
- **Sub-sistem Backend/Cloud:** Menyediakan REST API gateway, MQTT subscriber, manajemen autentikasi via Supabase Auth, enkripsi field sensitif, custom sync endpoint untuk mobile, dan audit trail otomatis.
- **Sub-sistem Mobile:** Pencatatan distribusi 3 langkah Trauma-Aware, prediksi AI dengan Rationale Chips, laporan anomali, Background Sync (Android+iOS), penyimpanan SQLite lokal.
- **Sub-sistem Web Dashboard:** Summary cards 4 metrik, grafik time-series, regional heatmap, manajemen request logistik, data posko, RBAC personnel, dan pengaturan sistem.

3.2.3 Categorization of Functions

Fungsi-fungsi sistem Log-Shield dikategorikan ke dalam tiga kelompok utama:

- **Fungsi Input:** Pembacaan sensor Load Cell (otomatis), input pencatatan distribusi (manual via mobile), pengajuan request logistik dari posko, login dan autentikasi pengguna, input parameter konfigurasi sistem oleh administrator.
- **Fungsi Transformasi:** Median filtering data sensor, validasi rentang nilai bobot, enkripsi AES-256 field sensitif, pelatihan model TinyTimeMixers, generasi

Rationale Chips dari SHAP values, resolusi konflik Last Write Wins, agregasi data demografi per posko.

- **Fungsi Output:** Publikasi data MQTT dari ESP32, push notifikasi Supabase Realtime untuk stok kritis, rendering grafik dan heatmap di dashboard, ekspor audit log, panggilan langsung ke PJ posko via fitur CALL.

3.2.4 Formulation of Performance Characteristics

Tabel 3.3 Formulasi Karakteristik Performa

Kategori	Parameter	Satuan	Target
Teknis	Akurasi sensor Load Cell	gram	±150
Teknis	Latensi MQTT end-to-end	detik	< 10
Teknis	Konsumsi daya node IoT	mA @ 5V	< 500
Teknis	Throughput API backend	req/detik	≥ 100
Teknis	Latensi inferensi AI	detik	< 5
Operasional	Durasi offline mobile	jam	72
Operasional	Waktu sync 500 transaksi	detik	< 60
Kualitas	MAE model Time Series	persen	< 15
Kualitas	Uptime sistem per bulan	persen	≥ 95

3.3 Implementation Concept Exploration

3.3.1 Sistem Predecessor

Sistem predecessor Log-Shield adalah pendekatan manajemen logistik bencana konvensional yang masih dominan digunakan di Indonesia, terdiri dari pencatatan formulir kertas, rekapitulasi spreadsheet manual, dan koordinasi via WhatsApp/SMS antara petugas lapangan dan koordinator. Kelebihan sistem predecessor: tidak memerlukan infrastruktur teknologi, biaya implementasi rendah, dan tidak bergantung pada koneksi internet. Keterbatasan utama: jeda informasi 6–48 jam antara kondisi lapangan aktual dengan data terkonsolidasi, kerentanan terhadap kesalahan transkripsi, tidak adanya kemampuan prediktif, dan kesulitan rekonsiliasi data lintas wilayah pasca-bencana.

3.3.2 Pendekatan Inovatif

Log-Shield mengadopsi tiga pendekatan inovatif yang membedakannya dari sistem predecessor. Pertama, integrasi IoT dengan sensor Load Cell HX711 yang mengotomatisasi pemantauan stok fisik secara real-time, menghilangkan kebutuhan stocktake manual. Kedua, aplikasi mobile lintas platform berbasis Capacitor.js yang mendukung Android dan iOS dari satu codebase, dengan SQLite lokal sebagai Single

Source of Truth untuk operasi offline-first. Ketiga, infrastruktur cloud berbasis Supabase (PostgreSQL managed) yang menyediakan autentikasi, Realtime subscriptions, dan Row Level Security dalam satu platform, di-deploy pada DigitalOcean Droplet Singapore untuk latensi rendah ke Indonesia.

3.3.3 Pengembangan Sistem dan Sub-Sistem

3.3.3.1 Sub-sistem Pengomposan IoT

Sub-sistem IoT Log-Shield terdiri dari mikrokontroler ESP32-WROOM-32 sebagai pengolah data utama, modul ADC HX711 24-bit untuk pembacaan sensor Load Cell, dan Load Cell 50 kg sebagai transduser bobot fisik. Mikrokontroler dilengkapi dengan firmware Arduino C++ yang melakukan pembacaan setiap 30 detik dengan median filtering 15 sampel untuk menekan noise. Casing IP67 melindungi rangkaian dari debu dan kelembaban tinggi yang umum di area bencana. LED RGB WS2812 menyediakan diagnostik visual lapangan dengan kode warna: biru (booting), hijau (operasional), kuning (buffering offline), merah (fault).

3.3.3.2 Sub-sistem Pemantauan

Sub-sistem pemantauan dibangun di sekitar broker MQTT Mosquitto yang menerima publikasi data dari node IoT dengan QoS 1 (at-least-once delivery). Backend Node.js + Express bertindak sebagai MQTT subscriber, memvalidasi payload menggunakan shared-types dari packages/types, dan menyimpan pembacaan ke tabel stock_readings di Supabase PostgreSQL. Saat backend mendeteksi stok di bawah threshold kritis, Supabase Realtime memicu push notifikasi ke seluruh klien web yang berlangganan, memberikan peringatan instan kepada koordinator tanpa menunggu siklus polling 30 detik.

3.3.3.3 Sub-sistem Aplikasi

Sub-sistem aplikasi mobile Log-Shield dibangun menggunakan Capacitor.js yang memungkinkan satu codebase React dideployment sebagai aplikasi native Android dan iOS. SQLite lokal berperan sebagai Single Source of Truth: semua operasi tulis (pencatatan distribusi, update stok manual, laporan anomali) ditulis terlebih dahulu ke SQLite, lalu disinkronkan ke Supabase melalui custom sync endpoint saat koneksi tersedia. Capacitor Background Task Plugin memastikan sinkronisasi tetap berjalan meski aplikasi di background. Trauma-Aware UI menerapkan tombol minimal 48 dp, kontras rasio 7:1, tipografi body 18 sp, dan alur input maksimal 3 langkah.

3.3.4 Validasi dan Pengujian

3.3.4.1 Analisis Kelayakan

Kelayakan teknis dijamin oleh tiga faktor: kematangan teknologi (Capacitor 1 juta+ proyek aktif, ESP32 ekosistem masif, Supabase 1 juta+ proyek), ketersediaan keahlian tim (5 anggota dengan spesialisasi terdiversifikasi), dan dukungan komunitas open-source pada seluruh stack. Kelayakan ekonomi dijamin oleh total biaya MVP Rp 2.380.000

untuk hardware dan $43/bulan untuk infrastruktur cloud, jauh di bawah anggaran proyek mahasiswa pada umumnya. Kelayakan operasional dijamin oleh prinsip Trauma-Aware UI yang dirancang khusus untuk pengguna non-teknis dalam kondisi stres tinggi.

3.3.4.2 Eksperimen Kritis

Empat eksperimen kritis dijadwalkan untuk memvalidasi fitur desain yang esensial: (1) pengujian akurasi median filtering 15 sampel dengan beban referensi 5/10/20/40 kg, target deviasi ±150 gram; (2) pengujian retraining model TinyTimeMixers dengan dataset simulasi 30 hari, target MAE < 15%; (3) pengujian sinkronisasi SQLite ke Supabase setelah 72 jam offline dengan 200 transaksi, target zero data loss; (4) pengujian konsistensi resolusi konflik LWW dengan dua perangkat mengedit record yang sama secara offline.

3.3.4.3 Simulasi dan Pengujian Sistem

Skenario simulasi dirancang untuk menguji ketahanan sistem dalam kondisi ekstrem yang umum terjadi di lapangan bencana.

Tabel 3.4 Skenario Simulasi dan Pengujian Sistem

Skenario	Kondisi Uji	Hasil yang Diharapkan
Fluktuasi Lingkungan Ekstrem	Suhu 55–60°C, kelembaban 95% RH, vibrasi simulasi forklift gudang	Sensor tetap akurat dalam toleransi ±150 gram; casing IP67 tetap kedap
Variasi Beban Limbah/Stok	Beban 0 hingga 50 kg dengan increment 5 kg	Linearity sensor terverifikasi; tidak ada drift > 50 gram
Gangguan Konektivitas	Wi-Fi terputus 50 menit, ESP32 buffer 100 pembacaan ke NVS	Seluruh 100 pembacaan dipublikasikan ulang QoS 1 saat koneksi pulih; zero duplikasi di backend
Lonjakan Data Mendadak	100 node IoT simulator publish bersamaan setiap 30 detik	Backend throughput > 200 msg/detik; latensi p95 < 10 detik; CPU < 60%

3.3.5 Karakteristik Performa Sistem

3.3.5.1 Kapasitas Pemrosesan

Sistem Log-Shield dirancang untuk menangani kapasitas operasional skala kabupaten dengan target satu instance backend mendukung hingga 100 node IoT paralel, 50 posko aktif, dan 1.000+ transaksi distribusi per hari. Backend Express dengan Supabase connection pooling diproyeksikan menangani 150 req/detik pada konfigurasi DigitalOcean Droplet 2 vCPU + 2 GB RAM.

3.3.5.2 Efisiensi Dekomposisi

Tidak relevan untuk konteks Log-Shield yang berfokus pada manajemen logistik bencana, bukan pengomposan limbah. Parameter ekuivalen yang lebih relevan adalah efisiensi sinkronisasi: 500 transaksi tersinkron dalam < 60 detik pada jaringan 3G, dengan delta sync yang hanya mengirim record yang berubah.

3.3.5.3 Akurasi Sensor

Sensor Load Cell 50 kg dengan ADC HX711 24-bit dan median filtering 15 sampel mencapai akurasi deviasi ±150 gram pada rentang 5–40 kg beban referensi (verifikasi eksperimen kritis Deliverable 2 menunjukkan ±95 gram). Drift sensor dalam 10 menit pengamatan < 50 gram pada suhu stabil.

3.3.5.4 Pengalaman User

Antarmuka mobile dirancang dengan prinsip Trauma-Aware UI: tombol minimal 48 dp untuk dapat ditekan dengan sarung tangan, kontras warna rasio 7:1 untuk keterbacaan di bawah sinar matahari, tipografi 18 sp pada body text, alur input maksimal 3 langkah untuk pencatatan distribusi, dan undo button 5 detik untuk menghindari kesalahan fatal. Indikator sinkronisasi persisten di pojok atas setiap halaman memberikan transparansi status data kepada petugas. Waktu respons UI < 2 detik untuk seluruh interaksi karena operasi baca dilakukan terhadap SQLite lokal.

3.4 Performance Validation Concept Exploration

3.4.1 Tujuan Validasi Kebutuhan Performa

Validasi kebutuhan performa Log-Shield bertujuan memastikan bahwa parameter teknis yang ditetapkan dalam MoP dapat dicapai oleh konsep sistem terpilih dengan margin yang aman, mempertimbangkan ketidakpastian lingkungan operasional bencana. Validasi dilakukan sebelum memasuki tahap engineering development untuk menghindari kebutuhan re-design yang mahal di tahap akhir.

3.4.2 Proses Integrasi Karakteristik Performa

Integrasi karakteristik performa Log-Shield dilakukan melalui pendekatan top-down: dimulai dari kebutuhan tingkat sistem (MoE), diturunkan ke parameter sub-sistem (MoP), kemudian dialokasikan ke komponen-komponen spesifik. Setiap keputusan rekayasa didokumentasikan dalam tabel trade-off yang memperlihatkan alternatif yang dipertimbangkan dan alasan pemilihan.

Pembelajaran utama dari sistem predecessor adalah perlunya prioritas pada resiliensi (operasi offline) daripada fitur kompleks (analitik canggih). Hal ini tercermin pada keputusan menggunakan model TinyTimeMixers yang ringan dibandingkan deep learning kompleks, serta SQLite lokal sebagai SSOT dibandingkan ketergantungan penuh pada cloud.

3.4.3 Validasi Karakteristik Performa

Model efektivitas yang lebih rinci mendefinisikan kondisi-kondisi spesifik di mana sistem dianggap efektif. Untuk MoE1 (Akurasi Prediksi > 88%), validasi memerlukan: (a) data historis distribusi minimal 21 hari, (b) tidak ada anomali besar dalam jumlah pengungsi

(perubahan < 50% dari rata-rata), dan (c) MAE model di bawah 15%. Jika salah satu kondisi tidak terpenuhi, sistem otomatis fallback ke rata-rata historis 30 hari.

Dalam istilah rekayasa, kebutuhan performa diterjemahkan menjadi spesifikasi implementasi konkret: AES-256 untuk enkripsi at-rest, TLS 1.3 untuk enkripsi in-transit, Row Level Security pada level database, JWT dengan masa berlaku 24 jam untuk autentikasi, dan audit trail append-only dengan retensi minimum 2 tahun.

3.4.4 Dokumentasi Kebutuhan

Seluruh kebutuhan performa Log-Shield didokumentasikan dalam struktur yang dapat ditelusuri dari objektif operasional hingga implementasi kode. Format dokumentasi: setiap kebutuhan memiliki kode unik (MoE1–MoE6, MoP1–MoP8, FR01–FR10), deskripsi naratif, formula pengukuran atau metrik, target nilai, dan referensi ke implementasi (file source code, skema database, atau test case). Dokumentasi ini disimpan dalam packages/types pada monorepo dan diversion-control bersama kode.

CONCEPT DEFINITION

3.5 Performance Requirements Analysis

Bagian ini membahas analisis kebutuhan performa Log-Shield, mencakup identifikasi parameter kunci, kompatibilitas, siklus hidup, keterjangkauan, sistem predecessor, dan metode penentuan kebutuhan pengguna.

3.5.1 Tujuan dan Ruang Lingkup Analisis

Tujuan analisis kebutuhan performa adalah mengidentifikasi parameter operasional dan kinerja kritis Log-Shield seperti kapasitas pemrosesan request API, akurasi sensor Load Cell, ketersediaan operasional sistem, dan efisiensi energi node IoT. Ruang lingkup analisis mencakup pengumpulan data melalui simulasi internal, benchmarking sistem konvensional dan international, serta pengujian prototipe pada tahap eksperimen kritis. Hasil analisis menjadi dasar pengambilan keputusan rekayasa pada tahap selanjutnya.

3.5.2 Kebutuhan Performa

Tabel 3.5 Kebutuhan Performansi Sistem

Kode	Nama Kebutuhan Performansi	Deskripsi
PR01	Kapasitas Throughput API	Backend mampu memproses minimal 100 request per detik dalam kondisi normal.
PR02	Kapasitas Node IoT Paralel	Satu instance backend mendukung hingga 100 node IoT yang publish data secara bersamaan.
PR03	Akurasi Sensor Load Cell	Deviasi pembacaan ±150 gram dari nilai aktual setelah median filtering 15 sampel.
PR04	Latensi End-to-End IoT	Waktu dari pembacaan sensor hingga simpan di Supabase tidak melebihi 10 detik (p95).
PR05	Akurasi Model Prediksi AI	Mean Absolute Error model TinyTimeMixers harus < 15% pada validasi 7 hari terakhir.
PR06	Durasi Operasi Offline	Aplikasi mobile mampu beroperasi penuh 72 jam tanpa internet menggunakan SQLite lokal.

3.5.3 Kebutuhan Kompatibilitas, RMA, dan Lingkungan

Kompatibilitas: Aplikasi mobile mendukung Android 8.0+ (API 26+) dan iOS 15.0+ untuk mencakup >95% perangkat pengguna di Indonesia. Backend kompatibel dengan stack standar (Node.js 20, PostgreSQL 15 via Supabase). Node IoT menggunakan Wi-Fi 2.4 GHz yang didukung mayoritas router di gudang logistik.

Reliability, Maintainability, Availability (RMA): MTBF node IoT > 8.760 jam (1 tahun kontinu) berdasarkan datasheet ESP32. Maintainability dijamin oleh OTA update firmware dual-partition yang dapat di-rollback otomatis jika gagal boot. Availability sistem cloud > 95% per bulan, didukung SLA DigitalOcean 99.99% dan Supabase 99.9%.

Lingkungan Operasional: Sistem berfungsi pada rentang suhu -10°C hingga 60°C, kelembaban 30–95% RH, dan ketinggian 0–3.000 mdpl. Casing IP67 melindungi terhadap debu vulkanik dan banjir hingga rendam 1 meter selama 30 menit.

3.5.4 Analisis Siklus Hidup Sistem

3.5.4.1 Concept Development Phase

- **Need Analysis:** Identifikasi masalah logistik bencana di Indonesia, kebutuhan Enam Tepat BNPB, dan kepatuhan UU PDP No. 27/2022.
- **Concept Exploration:** Eksplorasi alternatif dari sistem manual hingga IoT terintegrasi penuh dengan AI predictive.
- **Concept Definition:** Pemilihan Konsep B (Semi-Otomatis dengan Digital Monitoring) sebagai keseimbangan terbaik antara fungsionalitas, biaya, dan timeline.
- **Output:** Spesifikasi fungsional awal (FR01–FR10), functional block diagram, dan dokumen kebutuhan performa (PR01–PR06).

3.5.4.2 Engineering Development Phase

- **Advanced Development:** Alokasi fungsi-fungsi sistem ke komponen fisik (ESP32, sensor) dan software (Capacitor, FastAPI, Supabase).
- **Engineering Design:** Pengembangan firmware ESP32, algoritma prediksi TinyTimeMixers, integrasi pipeline SHAP, dan prototipe aplikasi mobile.
- **Integration and Evaluation:** Integrasi bertahap sub-sistem IoT+Backend, Mobile+Backend, AI+Backend, lalu integrasi total end-to-end.
- **Output:** Model integrasi komponen lengkap, spesifikasi produksi node IoT, rencana pengujian sistem.

3.5.4.3 Post Development Phase

- **Production:** Penggandaan node IoT untuk pilot testing 3–5 posko di satu kabupaten.
- **Operations and Support:** Pemantauan via dashboard DigitalOcean + Supabase, OTA update firmware, dan pelatihan petugas BPBD.
- **Output:** Dokumen operasional akhir, rencana pemeliharaan periodik, dan roadmap pengembangan fitur fase selanjutnya.

3.5.5 Kebutuhan Affordability dan Lifespan

Tabel 3.6 Kebutuhan Affordability dan Lifespan

Aspek	Deskripsi
Biaya Hardware per Unit Node IoT	Tidak melebihi Rp 800.000 per unit (ESP32 + HX711 + Load Cell + casing IP67).
Biaya Operasional Cloud	Total $43/bulan untuk infrastruktur (DigitalOcean Droplet $12 + AI Droplet $6 + Supabase Pro $25).
Komponen Hardware	Seluruh komponen tersedia di pasaran lokal Indonesia dengan waktu pengiriman 3–7 hari.
Platform Cloud	Supabase Free Tier untuk MVP; Supabase Pro untuk produksi.
Lifespan Hardware	Estimasi 3–5 tahun dengan pemeliharaan rutin; Load Cell dan HX711 < 5 tahun, ESP32 > 5 tahun.
Lifespan Software	Stack yang dipilih memiliki dukungan jangka panjang: Node.js 20 LTS hingga 2026, PostgreSQL 15 hingga 2027.
Total Cost of Ownership (TCO) 3 tahun	Estimasi Rp 25.000.000 untuk 10 node IoT + cloud + pelatihan.
Skenario ROI	Reduksi pemborosan logistik 30% per bencana berskala kabupaten = penghematan Rp 50–100 juta per kejadian, ROI < 1 bencana.

3.5.6 Peran Sistem Predecessor

Sistem predecessor (pencatatan manual + spreadsheet + WhatsApp) memberikan tiga insight kunci untuk pengembangan Log-Shield. Pertama, alur kerja yang dapat dipertahankan: konsep pencatatan per transaksi distribusi dengan identifikasi penerima yang jelas dipertahankan, hanya media-nya yang digantikan dari kertas ke aplikasi mobile. Kedua, keterbatasan yang harus diatasi: jeda informasi, ketiadaan prediksi, dan kerentanan terhadap kesalahan manusia mendorong adopsi prediksi AI dan sensor IoT. Ketiga, prioritas yang harus dipegang teguh: kesederhanaan operasional untuk petugas lapangan harus tetap diutamakan, karena kompleksitas adalah musuh adopsi pada konteks bencana.

Keterbatasan utama sistem predecessor (jeda 6–48 jam, error rate input 15–25%, ketidakmampuan prediktif) menjadi pendorong langsung pengembangan solusi berbasis IoT dengan sensor terintegrasi, aplikasi mobile dengan validasi input, dan AI Time Series Forecasting.

3.5.7 Metode Penentuan Kebutuhan Pengguna

Penentuan kebutuhan pengguna Log-Shield menggunakan pendekatan secondary research dan benchmarking karena keterbatasan akses ke pengguna BPBD aktual selama fase Deliverable 1–2. Metode yang digunakan: studi literatur (Sphere Project Handbook 2018, BNPB Annual Reports, evaluasi pasca-bencana Cianjur 2022), wawancara informal dengan kontak akademisi yang memiliki pengalaman BPBD, benchmarking sistem internasional (HELIOS USAID di Nigeria, OpenLMIS di Tanzania), dan pembuatan user persona berdasarkan profil khas petugas dan koordinator BPBD Indonesia.

Tabel 3.7 Klasifikasi Kebutuhan Pengguna

Kategori Kebutuhan	Contoh untuk Log-Shield
Kebutuhan Fungsional	Pencatatan distribusi offline, prediksi kebutuhan logistik, pemantauan stok via sensor
Kebutuhan Performa	Notifikasi real-time stok kritis, akurasi sensor ±150 gram, sync < 60 detik
Kebutuhan Usability	Antarmuka 3 langkah, kontras tinggi, tombol besar untuk petugas dengan sarung tangan
Kebutuhan Keamanan	Enkripsi data pengungsi sesuai UU PDP, RBAC, audit trail 2 tahun
Kebutuhan Interoperabilitas	KIB 16-digit Satu Data Indonesia, format JSON standar untuk pertukaran data

3.6 Functional Analysis and Formulation

Bagian ini menjelaskan tujuan dan metode analisis fungsional, diagram fungsional, identifikasi fungsi utama dan sub-fungsi, alokasi hardware/software, analisis trade-off, dan formulasi spesifikasi fungsional awal Log-Shield.

3.6.1 Tujuan dan Metode Analisis Fungsional

3.6.1.1 Tujuan Analisis Fungsional

Functional Analysis (FA) Log-Shield bertujuan untuk:

- Mengidentifikasi fungsi-fungsi utama dan turunan yang harus dilakukan sistem untuk memenuhi kebutuhan operasional Enam Tepat BNPB dan kebutuhan pengguna petugas BPBD.
- Memecah sistem kompleks Log-Shield menjadi lima sub-sistem logis (IoT, Backend, AI, Mobile, Web) yang saling berinteraksi, kemudian dialokasikan ke elemen fisik dan software.
- Menentukan alur dan ketergantungan antar fungsi sebagai dasar desain arsitektur sistem dan spesifikasi teknis yang terukur, terdokumentasi dalam packages/types monorepo.

3.6.1.2 Metode Analisis Fungsional

Tiga metode utama yang digunakan dalam analisis fungsional Log-Shield:

- **Functional Flow Block Diagram (FFBD):** Digunakan untuk memvisualisasikan alur urutan fungsi sistem mulai dari input sensor IoT, pemrosesan backend, hingga output dashboard. Kelebihan: memperlihatkan dependency dan paralelisme. Langkah: identifikasi fungsi utama → dekomposisi → hubungkan dengan alur (sequence/OR/AND) → tandai input/output.
- **IDEF0 (Integration DEFinition for Function Modeling):** Diterapkan pada modul AI Engine untuk memodelkan ICOM (Input, Control, Output, Mechanism)

pelatihan model TinyTimeMixers. Berbasis standar ISO/IEC 19501, cocok untuk sistem kompleks yang memerlukan tracking input/output formal.

- **Use Case Analysis + Activity Diagram:** Digunakan untuk menggambarkan interaksi pengguna di aplikasi mobile dan dashboard web. Aktor yang teridentifikasi: Petugas Lapangan, Koordinator, Administrator, dan komponen non-manusia (Sensor IoT, AI Engine, Supabase Backend).

3.6.2 Identifikasi Fungsi Utama dan Sub-Fungsi

Tabel 3.8 Identifikasi Fungsi Utama dan Sub-Fungsi

Kode	Fungsi Utama	Sumber	Sub-Fungsi	Deskripsi Sub-Fungsi
F01	Pemantauan Stok Fisik	FR03, FR04, MoP1	F01.1	Pembacaan Bobot Otomatis – Sensor Load Cell HX711 baca bobot stok setiap 30 detik.
			F01.2	Median Filtering – ESP32 menerapkan filter 15 sampel untuk menekan noise.
F02	Prediksi Kebutuhan Logistik	FR01, FR02, MoE1	F02.1	Pelatihan Model Time Series – TinyTimeMixers dilatih harian dengan data 30 hari terakhir.
			F02.2	Generasi Rationale Chips – SHAP values dikonversi ke narasi Bahasa Indonesia.
F03	Pencatatan Distribusi Mobile	FR05, FR06	F03.1	Alur 3 Langkah Trauma-Aware – pilih komoditas → input kuantitas → konfirmasi KIB.
			F03.2	Penyimpanan SQLite Lokal – semua transaksi disimpan offline-first.
F04	Sinkronisasi Dua Arah	FR07	F04.1	Custom Sync Endpoint – batch upsert dari mobile ke Supabase via REST API.
			F04.2	Last Write Wins Resolution – konflik diselesaikan berdasarkan timestamp hybrid.
F05	Keamanan & Audit	FR08	F05.1	Enkripsi AES-256 field sensitif – nama, NIK, kondisi kesehatan.
			F05.2	RBAC + RLS – kontrol akses berbasis role di aplikasi dan database.

3.6.3 Alokasi Fungsi: Hardware vs Software

Tabel 3.9 Alokasi Fungsi: Hardware vs Software

Kode	Nama Sub-Fungsi	Jenis Fungsi	Hardware	Software
F01.1	Pembacaan Bobot Otomatis	Input	Load Cell 50kg + HX711	Driver HX711 Library + ESP32 firmware
F01.2	Median Filtering	Transformasi	ESP32 ARM Cortex-M4	Algoritma insertion sort C++

Kode	Nama Sub-Fungsi	Jenis Fungsi	Hardware	Software
F02.1	Pelatihan Model Time Series	Transformasi	DigitalOcean Droplet 1 vCPU	Python + FastAPI + TinyTimeMixers
F02.2	Generasi Rationale Chips	Output	DigitalOcean Droplet 1 vCPU	SHAP library + template rule-based Bahasa Indonesia
F03.1	Alur 3 Langkah Distribusi	Input	Smartphone Android/iOS	Capacitor.js + React + Trauma-Aware UI Components
F03.2	Penyimpanan SQLite Lokal	Transformasi	Storage internal perangkat	Capacitor SQLite Plugin
F04.1	Custom Sync Endpoint	Output	DigitalOcean Droplet 2 vCPU	Node.js + Express + Supabase Client
F04.2	LWW Resolution	Transformasi	DigitalOcean Droplet 2 vCPU	Conflict resolver service + audit logger
F05.1	Enkripsi AES-256	Transformasi	DigitalOcean Droplet 2 vCPU	Node.js crypto module + key dari DO Secrets
F05.2	RBAC + RLS	Output	Supabase PostgreSQL	Row Level Security policies + Express middleware

3.6.4 Alokasi Fungsi Lanjutan: Kesesuaian & Interaksi

Alokasi fungsi lanjutan mempertimbangkan interaksi antar fungsi, fleksibilitas pengembangan, dan ketercapaian MoP/MoE. Setiap sub-fungsi dipetakan ke kombinasi komponen yang optimal.

Tabel 3.10 Alokasi Fungsi Lanjutan

Sub-Fungsi	Komponen / Entitas	Justifikasi Kesesuaian & Interaksi
F01.1 Pembacaan Bobot	Load Cell + HX711 + ESP32	Load Cell membaca hingga 50 kg, dikonversi 24-bit oleh HX711, dikirim ESP32 untuk validasi rentang nilai dan median filtering.
F02.1 Pelatihan Model AI	FastAPI + Supabase	FastAPI di Droplet terpisah agar tidak membebani backend; akses Supabase via supabase-py untuk batch read efisien.
F02.2 Rationale Chips	SHAP + Backend Express	SHAP values dihitung di AI Engine, diubah jadi narasi via template, disimpan di kolom JSONB di tabel predictions.
F03.1 Alur 3 Langkah	Capacitor.js + React	Capacitor menyediakan native bridge ke Android/iOS, React menyediakan komponen UI yang dapat di-test secara cepat.
F04.1 Custom Sync	Express + Supabase Client	Express memvalidasi payload, Supabase Client melakukan upsert efisien dengan parameterized query.

Sub-Fungsi	Komponen / Entitas	Justifikasi Kesesuaian & Interaksi
F05.1 Enkripsi AES-256	Node.js crypto + DO Secrets	Kunci enkripsi terpisah dari kode di DigitalOcean Secrets Manager, rotasi berkala via CI/CD.

3.6.5 Analisis Trade-Off Fungsi

Analisis trade-off dilakukan untuk mengevaluasi opsi implementasi teknis dari fungsi-fungsi utama berdasarkan biaya, performa, risiko, dan kesesuaian kebutuhan pengguna.

Tabel 3.11 Analisis Trade-Off Fungsi

Fungsi/Sub-Fungsi	Alternatif Konfigurasi	Target Kinerja (MoP)	Risiko Teknis	Keputusan
F01 Pemantauan Stok	Sensor Load Cell HX711 + ESP32 + Wi-Fi	Akurasi ±150 g, latensi < 10 s	Kalibrasi berkala, sensitif terhadap vibrasi	Dipilih
F01 Pemantauan Stok	Sensor RFID untuk setiap item	Tracking item-level	Biaya 10x lipat, instalasi kompleks	Tidak dipilih
F02 Prediksi AI	TinyTimeMixers (Time Series ringan)	MAE < 15%, latensi < 5 s	Akurasi terbatas data historis	Dipilih
F02 Prediksi AI	Deep Learning LSTM	Akurasi lebih tinggi potensial	Butuh GPU, latensi tinggi	Tidak dipilih untuk MVP
F04 Sinkronisasi	Custom sync REST endpoint	< 60 detik untuk 500 transaksi	Implementasi conflict resolver	Dipilih
F04 Sinkronisasi	PouchDB-CouchDB native	Native protocol matang	Tidak kompatibel dengan Supabase PostgreSQL	Tidak dipilih

3.6.6 Formulasi Spesifikasi Fungsional Awal

Tabel 3.12 Formulasi Spesifikasi Fungsional Awal

Kode	Sub-Fungsi	Spesifikasi Fungsional	Parameter	Satuan	Target
FS01	F01.1	Sistem mengukur bobot stok komoditas via sensor	Kapasitas pembacaan	kg	0–50
FS02	F01.1	Sistem menerapkan median filtering pada pembacaan sensor	Akurasi setelah filtering	gram	±150
FS03	F02.1	Sistem memprediksi kebutuhan logistik 7 hari ke depan	MAE prediksi	%	< 15
FS04	F03.1	Sistem mencatat distribusi dalam alur 3 langkah	Waktu input rata-rata	detik	< 30

Kode	Sub-Fungsi	Spesifikasi Fungsional	Parameter	Satuan	Target
FS05	F04.1	Sistem sinkronisasi data offline ke server	Durasi 500 transaksi	detik	< 60
FS06	F05.1	Sistem mengenkripsi field sensitif	Algoritma enkripsi	-	AES-256

3.6.7 Kuantifikasi Sementara Spesifikasi Fungsional

Tabel 3.13 Kuantifikasi Sementara Spesifikasi Fungsional

Kode	Spesifikasi Fungsional	Parameter Dikontrol	Nilai Kuantitatif Sementara
FS01	Pembacaan bobot stok	Rentang beban	0–50 kg (Load Cell 50kg half-bridge)
FS02	Akurasi sensor	Deviasi pembacaan	±150 gram (target), ±95 gram (eksperimen D2)
FS03	Akurasi prediksi	MAE relatif	< 15% target, ~11.4% proyeksi
FS04	Latensi UI mobile	Respons interaksi	< 2 detik target, < 1 detik proyeksi
FS05	Throughput backend	Request per detik	≥ 100 target, ~150 proyeksi
FS06	Operasi offline	Durasi maksimum	72 jam target dengan zero data loss

3.7 Concept Selection

3.7.1 Identifikasi dan Deskripsi Alternatif Konsep

Empat alternatif konsep dieksplorasi pada Deliverable 1.

Tabel 3.14 Identifikasi dan Deskripsi Alternatif Konsep

Alternatif	Deskripsi Ringkas	Kelebihan	Kekurangan
Konsep A	Manual Spreadsheet + WhatsApp – sistem konvensional yang saat ini digunakan	Murah, tidak butuh infrastruktur, tidak butuh internet	Jeda info 48–72 jam, error 15–25%, tidak ada prediksi
Konsep B	Semi-Otomatis dengan Digital Monitoring – IoT sensor + AI prediktif + Mobile offline-first	Akurasi tinggi, prediktif, offline-first, biaya terjangkau	Butuh investasi awal hardware + cloud
Konsep C	Otomasi Penuh AI Real-time – seluruh transaksi otomatis dengan computer vision + AI	Performa terbaik, minim intervensi manusia	Biaya >500 juta, butuh >6 bulan, overkill MVP
Konsep D	Platform SaaS Pihak Ketiga (HELIOS, OpenLMIS)	Implementasi cepat, fitur lengkap	Data di server luar negeri, tidak patuh UU PDP

3.7.2 Matriks Evaluasi Alternatif (Weighted Scoring)

Tabel 3.15 Matriks Evaluasi Alternatif (Weighted Scoring)

Kriteria	Bobot	Konsep A	Konsep B	Konsep C	Konsep D
Performa	0.30	1	4	5	4
Biaya	0.30	5	4	1	2
Risiko	0.20	3	4	2	3
Jadwal (timeline)	0.20	5	5	1	4
Skor Tertimbang	1.00	3.10	4.70	2.30	3.20

3.7.3 Penilaian Performa, Biaya, Jadwal, dan Risiko

- **Konsep A:** Biaya dan jadwal sangat baik, namun performa sangat terbatas dan tidak menyelesaikan masalah inti. Tidak layak sebagai sistem strategis.
- **Konsep B:** Keseimbangan optimal di seluruh kriteria. Performa baik dengan biaya terjangkau dan timeline 8 minggu yang realistis. Risiko teknis terkendali melalui mitigasi yang terdokumentasi.
- **Konsep C:** Performa tertinggi namun biaya dan timeline tidak realistis untuk MVP mahasiswa. Cocok untuk fase pengembangan lanjutan setelah MVP berhasil.
- **Konsep D:** Implementasi cepat namun melanggar mandat kedaulatan data Indonesia (UU PDP No. 27/2022). Tidak dapat dipilih meskipun secara teknis layak.

3.7.4 Penentuan Bobot dan Peringkat

Bobot kriteria ditentukan berdasarkan fokus proyek pada validasi cepat dengan biaya terkendali:

- **Performa (30%)** – kemampuan sistem memenuhi MoE/MoP yang ditetapkan.
- **Biaya (30%)** – keterjangkauan dalam anggaran proyek mahasiswa dan kelayakan operasional jangka panjang.
- **Risiko (20%)** – probabilitas kegagalan teknis, keamanan, dan operasional.
- **Jadwal (20%)** – kemampuan menyelesaikan MVP dalam timeline 8 minggu.

Hasil evaluasi: Konsep B memperoleh skor tertinggi 4.70 karena memberikan keseimbangan optimal antara kemampuan teknis dan keterjangkauan, serta mendukung penyelesaian tepat waktu.

3.7.5 Finalisasi dan Justifikasi Konsep Terpilih

Konsep B: Semi-Otomatis dengan Digital Monitoring dipilih sebagai konsep final Log-Shield karena:

- Memberikan keseimbangan antara kemampuan teknis (prediksi AI, monitoring IoT, mobile offline) dan keterjangkauan biaya (Rp 2.380.000 + $43/bulan).
- Dapat dikembangkan dalam timeline 8 minggu dengan 5 anggota tim, memenuhi milestone akademis.
- Memungkinkan pengujian dan validasi performa sistem secara efektif dengan risiko teknis yang lebih rendah dibandingkan Konsep C.
- Bersifat modular dan dapat dikembangkan lebih lanjut menjadi sistem otomatis penuh (transisi ke Konsep C) di fase pengembangan selanjutnya tanpa perombakan arsitektur.
- Mematuhi UU PDP No. 27/2022 dan standar Satu Data Indonesia (KIB 16-digit), berbeda dengan Konsep D yang menggunakan server di luar negeri.

3.8 Concept Validation

3.8.1 Tujuan dan Ruang Lingkup Analisis

Validasi konsep Log-Shield bertujuan memastikan bahwa Konsep B (Semi-Otomatis dengan Digital Monitoring) telah memenuhi kelayakan dari sisi teknis, performa, dan operasional, serta dapat diteruskan ke tahap pengembangan rekayasa lanjutan.

Metodologi validasi mencakup lima tahapan utama:

- Pemodelan sistem dan lingkungan: memetakan hubungan antar sub-sistem menggunakan skenario realistis berbasis kebutuhan pengguna BPBD.
- Simulasi performa berbasis MoP/MoE: mengevaluasi kemampuan sistem terhadap parameter kritis seperti akurasi sensor, efisiensi sinkronisasi, dan waktu respons.
- Eksperimen kritis terhadap prototipe: menguji sistem dalam kondisi ekstrem untuk memverifikasi fitur desain yang krusial.
- Analisis hasil validasi: mengidentifikasi defisiensi sistem dan menetapkan sumber masalah.
- Iterasi konsep dan revisi spesifikasi: menyempurnakan desain berdasarkan hasil validasi.

3.8.2 Pemodelan Sistem dan Lingkungan

3.8.2.1 Tujuan Pemodelan

- Menentukan batas sistem (system boundary) yang membedakan elemen yang dikendalikan sistem dengan elemen lingkungan eksternal.
- Mendeskripsikan arsitektur fungsional mencakup hubungan antara sensor, mikrokontroler, backend, AI engine, dan antarmuka pengguna.

- Mengidentifikasi parameter masukan utama dari lingkungan (suhu, kelembaban, vibrasi, kualitas Wi-Fi) yang memengaruhi perilaku sistem.
- Menyediakan dasar untuk skenario simulasi yang akan dijalankan pada model efektivitas.

3.8.2.2 Penentuan Boundary System

Mengacu pada prinsip Kossiakoff, boundary system Log-Shield didefinisikan sebagai berikut:

Tabel 3.16 Penentuan Boundary System

Kategori	Deskripsi
Bagian Dalam Sistem (System)	• ESP32 sebagai unit pengendali node IoT • Sensor Load Cell + HX711 (deteksi bobot stok) • Backend Express + Mosquitto (DigitalOcean) • AI Engine FastAPI (DigitalOcean) • Supabase PostgreSQL + Auth + Realtime • Aplikasi Mobile Capacitor.js (Android+iOS) • Dashboard Web React • Firmware ESP32 (median filter, NVS buffer, OTA)
Bagian Luar Sistem (Lingkungan)	• Stok logistik fisik dengan variasi bobot • Kondisi suhu (20–60°C) dan kelembaban (30–95% RH) gudang • Koneksi Wi-Fi 2.4 GHz yang intermittent di area bencana • Petugas BPBD dengan latar belakang non-teknis • Pengungsi dengan profil demografi beragam (balita, lansia, disabilitas) • Sistem eksisting BNPB yang akan diintegrasi di fase lanjutan

3.8.2.3 Asumsi Lingkungan Operasional

Tabel 3.17 Asumsi Lingkungan Operasional

Kategori	Asumsi
Fisik	• Suhu gudang berkisar antara 25–55°C • Kelembaban relatif 50–95% RH • Beban stok per Load Cell maksimal 40 kg • Vibrasi gudang dari aktivitas bongkar-muat forklift
Infrastruktur	• Wi-Fi 2.4 GHz tersedia di gudang utama tetapi intermittent • Daya PLN 220V tersedia umumnya • Backup power bank 20.000 mAh mencukupi 24–36 jam
Pengguna	• Petugas tidak memiliki latar belakang IT • Interaksi terbatas: input distribusi, baca prediksi, lapor anomali • Petugas mungkin memakai sarung tangan saat operasi • Koordinator memiliki perangkat desktop/laptop dengan akses internet stabil

3.8.2.4 Model Efektivitas Sistem

Model efektivitas Log-Shield didefinisikan oleh MoE dan MoP yang telah ditetapkan pada Tabel 2.5 dan 2.6.

Tabel 3.18 Simulasi Kinerja Sistem

No	Skenario	Parameter Diuji	Respons Sistem yang Diharapkan
1	Stok kritis terdeteksi	Bobot stok < threshold	Supabase Realtime push notifikasi ke dashboard < 1 detik, status Critical Items diperbarui
2	Beban Maksimum Sensor	Bobot 49.5 kg di Load Cell	Pembacaan stabil, tidak overflow ADC, akurasi tetap ±150 gram

No	Skenario	Parameter Diuji	Respons Sistem yang Diharapkan
3	Wi-Fi Terputus 50 menit	Koneksi terputus	100 pembacaan ter-buffer di NVS ESP32, dikirim ulang berurutan QoS 1 saat pulih, zero data loss
4	Mobile Offline 72 jam	Aplikasi tanpa koneksi	200 transaksi tercatat di SQLite, sync sukses < 60 detik saat online, conflict LWW deterministik

3.8.2.5 Eksperimen Kritis

Eksperimen kritis dilakukan untuk memverifikasi fitur desain yang bersifat esensial dalam kondisi ekstrem yang tidak dapat diwakili sepenuhnya melalui simulasi komputer.

Tabel 3.19 Parameter dan Kondisi Uji Eksperimen Kritis

Skenario	Kondisi Ekstrem	Tujuan Uji
Kelembaban tinggi gudang	> 90% RH selama 24 jam	Menguji kekedapan casing IP67 dan kestabilan sensor
Beban maksimum	49.5 kg pada Load Cell	Menguji ketahanan transduser dan akurasi pembacaan di batas atas
Wi-Fi flapping	Putus-sambung 10 siklus 5 menit	Menguji robustness reconnect MQTT dan integritas data NVS
Lonjakan data	100 node IoT publish bersamaan	Menguji throughput backend dan latensi end-to-end

3.8.3 Analisis Hasil dan Validasi

3.8.3.1 Tujuan Analisis

- Mengidentifikasi penyebab ketidaksesuaian antara performa aktual eksperimen kritis dengan ekspektasi sistem.
- Mengklasifikasikan ketidaksesuaian: defisiensi asumsi desain, kelemahan model pengujian, atau persyaratan terlalu ketat.
- Menentukan perbaikan yang diperlukan melalui iterasi desain sebelum tahap engineering development.

Tabel 3.20 Klasifikasi Ketidaksesuaian dan Akar Masalah

No	Jenis Ketidaksesuaian	Kasus	Analisis Penyebab	Rekomendasi
1	Defisiensi asumsi	Pembacaan sensor tidak stabil pada vibrasi tinggi	Asumsi awal lingkungan tenang, padahal forklift sering melewati area gudang	Tambahkan median filtering 15 sampel dan validasi rentang nilai
2	Kelemahan model uji	iOS Background Task tidak terjamin intervalnya	Model tidak mempertimbangkan kebijakan Apple	Tambahkan fallback sync saat app dibuka dan manual trigger

No	Jenis Ketidaksesuaian	Kasus	Analisis Penyebab	Rekomendasi
			membatasi background execution	
3	Persyaratan terlalu ketat	MAE prediksi < 10% sulit dicapai dengan < 21 hari data	Target awal optimis, data historis terbatas pada awal deployment	Revisi target ke < 15% MAE dengan fallback rata-rata historis

3.8.4 Iterasi Konsep Sistem dan Revisi Spesifikasi

3.8.4.1 Tujuan Iterasi

- Menyesuaikan desain sistem dengan temuan validasi terhadap kondisi riil lingkungan gudang dan area bencana.
- Menghindari overdesign akibat persyaratan terlalu ketat, dengan tetap menjaga kualitas performa sistem.
- Memastikan bahwa sistem tetap layak secara teknis dan ekonomis untuk dilanjutkan ke tahap engineering.

3.8.4.2 Revisi dan Iterasi yang Dilakukan

Tabel 3.21 Revisi dan Iterasi Desain

No	Elemen Diubah	Revisi/Iterasi	Alasan Perubahan	Dampak
1	Target MAE prediksi	Dari < 10% → < 15%	Data historis < 21 hari pada awal deployment	Realistis untuk fase MVP; fallback historical mean
2	Strategi sync mobile	Dari PouchDB-CouchDB native → Custom Sync Endpoint	Supabase tidak mendukung PouchDB native replication	Implementasi lebih fleksibel, kontrol penuh LWW
3	iOS Background Task	Dari fixed 15 menit → adaptive + manual trigger	Apple membatasi background execution	Fallback sync saat app dibuka memastikan tidak ada data loss
4	Cloud provider	Dari AWS → DigitalOcean + Supabase	AWS billing kompleks dan mahal	Biaya $43/bulan prediktif, antarmuka simpler
5	Platform mobile	Hanya Android → Android + iOS	Petugas BPBD juga menggunakan iPhone	Cakupan pengguna lebih luas via Capacitor.js

3.8.4.3 Hasil Iterasi dan Keputusan Final

- Revisi tidak mengubah arsitektur utama sistem, hanya menyesuaikan parameter dan stack teknologi untuk meningkatkan resiliensi dan kelayakan.

- Tim menyimpulkan bahwa Konsep B (Semi-Otomatis + Digital Monitoring) tetap menjadi solusi paling optimal dan realistis untuk konteks BPBD Indonesia.
- Rekomendasi: sistem dapat dilanjutkan ke fase engineering development dengan rancangan versi final berdasarkan iterasi ini.

3.8.5 Kesimpulan Concept Validation

Pemodelan sistem menunjukkan bahwa arsitektur semi-otomatis berbasis IoT + AI + Mobile Offline-First mampu berinteraksi efektif dengan lingkungan operasional BPBD Indonesia yang beragam. Simulasi dan eksperimen kritis membuktikan bahwa sistem memenuhi sebagian besar MoP dan MoE, serta menunjukkan ketahanan dalam kondisi ekstrem (lingkungan, konektivitas, beban). Konsep B: Semi-Otomatis dengan Digital Monitoring telah memenuhi kriteria kelayakan teknis (komponen tersedia, ekosistem matang), lingkungan (rating IP67, suhu/kelembaban ekstrem), dan operasional (Trauma-Aware UI, biaya terjangkau). Sistem dapat dilanjutkan ke tahap engineering development dengan risiko yang dapat diterima.

4. ADVANCED DEVELOPMENT

Fase Pengembangan Lanjutan berfokus pada pematangan konsep sistem Log-Shield terpilih (Konsep B) dan teknologi kritis hingga titik di mana desain rekayasa skala penuh dapat dilanjutkan dengan risiko yang dapat diterima. Fase ini melibatkan analisis persyaratan terperinci, desain fungsional, pengembangan prototipe, dan pengujian pengembangan awal, didukung oleh strategi pengurangan risiko yang kuat.

4.1 Requirement Analysis

4.1.1 System Functional Specification

Sistem Log-Shield terdiri dari beberapa komponen utama yaitu website (dashboard koordinator), mobile application (aplikasi petugas lapangan Android dan iOS), backend cloud (Node.js + Express + Supabase), AI engine (Python FastAPI), dan IoT (ESP32 + Load Cell + HX711). Seluruh komponen berkomunikasi melalui Supabase PostgreSQL sebagai single source of truth dan broker MQTT Mosquitto sebagai jalur data sensor real-time.

Tabel 4.1 Spesifikasi Fungsional Sistem Log-Shield

Kode	Kebutuhan Sistem	Deskripsi
F01	Sistem dapat memprediksi kebutuhan logistik per posko	AI engine menghasilkan prediksi 7 hari ke depan untuk setiap kombinasi posko dan komoditas berdasarkan model TinyTimeMixers yang dilatih ulang setiap pukul 02.00 WIB.
F02	Sistem menyediakan Rationale Chips Explainable AI	Setiap prediksi dilengkapi 2–3 chip narasi dalam Bahasa Indonesia berbasis SHAP values, menjelaskan fitur dominan yang memengaruhi hasil prediksi.
F03	Sistem memantau stok fisik secara real-time	Node IoT ESP32 dengan sensor Load Cell HX711 mengirim pembacaan bobot setiap 30 detik via MQTT QoS 1 ke broker Mosquitto.
F04	Sistem mendukung operasi mobile offline 72 jam	Aplikasi mobile menyimpan transaksi di SQLite lokal dan sinkronisasi otomatis ke Supabase saat koneksi pulih.
F05	Sistem menerapkan resolusi konflik LWW	Konflik data antar perangkat diselesaikan deterministik dengan timestamp hybrid, revisi yang kalah dicatat di audit_logs.
F06	Sistem menyediakan dashboard real-time	Dashboard web menampilkan 4 summary cards, grafik time-series, regional heatmap, dan halaman manajemen (Logistics, Assets, Posko, Personnel, Settings).
F07	Sistem menggunakan KIB 16-digit	Seluruh entitas data menggunakan Kode Identitas Bencana 16-digit sebagai identifier sesuai standar Satu Data Indonesia.

Kode	Kebutuhan Sistem	Deskripsi
F08	Sistem mengenkripsi data sensitif	Field nama, NIK, kondisi kesehatan dienkripsi AES-256 sebelum disimpan ke Supabase; transmisi via TLS 1.3.
F09	Sistem menerapkan RBAC 4 tingkat	Role Admin, Koordinator, Lapangan dengan akses berbeda; ditegakkan via Supabase Auth JWT + Row Level Security PostgreSQL.

4.1.2 Requirement Derivation

Requirement derivation dilakukan untuk meninjau kembali skenario-skenario yang akan dihadapi oleh sistem, mencakup persyaratan terkait kompatibilitas, reliabilitas, dan faktor lingkungan.

1. Requirement Derivation for System

Tabel 4.2 Requirement Derivation for System Log-Shield

Kode	Requirement
R-1	Sistem dapat memprediksi kebutuhan logistik harian per posko dengan MAE < 15% berdasarkan data historis minimal 21 hari.
R-2	Sistem dapat memantau bobot stok fisik dengan akurasi ±150 gram dan latensi end-to-end < 10 detik.
R-3	Sistem dapat beroperasi offline pada mobile selama 72 jam tanpa data loss menggunakan SQLite lokal.
R-4	Sistem dapat menyinkronkan 500 transaksi dari mobile ke Supabase dalam waktu < 60 detik pada jaringan 3G.
R-5	Sistem menggunakan KIB 16-digit pada seluruh entitas data sebagai identifier Satu Data Indonesia.
R-6	Sistem menerapkan enkripsi AES-256 pada field sensitif dan TLS 1.3 untuk seluruh transmisi data.
R-7	Sistem mendukung RBAC 4 tingkat (Admin, Koordinator, Lapangan) via Supabase Auth JWT + RLS.

2. Requirement Derivation for Website and Web Service

Pada sub-sistem website dan web service, seluruh requirement derivation berlaku karena web service bertanggung jawab atas logika bisnis dan persistensi data untuk seluruh klien (website, mobile, dan IoT).

Tabel 4.3 Requirement Derivation for Website and Web Service

Kode	Requirement
R-1	Web service memproses prediksi dari AI engine dan menyajikannya via REST API.
R-2	Web service menerima data MQTT dari IoT, validasi via shared-types, simpan ke Supabase.
R-4	Web service menyediakan custom sync endpoint POST /api/v1/sync untuk batch upsert dari mobile.

Kode	Requirement
R-5	Web service memvalidasi format KIB 16-digit pada semua endpoint create/update.
R-6	Web service mengenkripsi field sensitif sebelum simpan; dekripsi hanya untuk user authorized via RBAC.
R-7	Web service menerapkan middleware JWT validation dan RBAC pada setiap endpoint.

3. Requirement Derivation for Mobile App

Tabel 4.4 Requirement Derivation for Mobile App

Kode	Requirement
R-3	Mobile menyimpan transaksi di SQLite lokal sebagai Single Source of Truth offline.
R-4	Mobile melakukan batch sync via Capacitor Background Task Plugin setiap 15 menit di Android (best-effort di iOS).
R-5	Mobile memvalidasi format KIB 16-digit di sisi client sebelum input ke SQLite.
R-6	Mobile menerapkan certificate pinning untuk mencegah MITM saat sinkronisasi.
R-7	Mobile menyimpan JWT secara aman di Capacitor Secure Storage (Keychain iOS / Keystore Android).

4. Requirement Derivation for IoT

Tabel 4.5 Requirement Derivation for IoT

Kode	Requirement
R-2	IoT membaca Load Cell setiap 30 detik dengan median filtering 15 sampel.
R-2	IoT mempublikasikan data via MQTT QoS 1 ke topik logshield/warehouse/{id}/scale/{node}/reading.
R-2	IoT menyimpan buffer NVS hingga 100 pembacaan saat Wi-Fi terputus, mengirim ulang berurutan saat pulih.
R-6	IoT mendukung TLS pada port 8883 jika broker MQTT eksternal (port 1883 untuk komunikasi internal).

4.1.3 Relationship with Operational Requirements

Setiap requirement teknis (R-1 hingga R-7) dapat ditelusuri kembali ke objektif operasional (O01–O06) yang ditetapkan di Bab 2. Sebagai contoh, R-1 (akurasi prediksi MAE < 15%) merupakan implementasi langsung dari O01 (akurasi distribusi logistik) yang diukur melalui MoE1. R-3 (offline 72 jam) memenuhi O06 (operasi di area terbatas koneksi). R-5 (KIB 16-digit) memenuhi O05 (integrasi dengan Satu Data Indonesia). R-6 (enkripsi AES-256) memenuhi O04 (kepatuhan UU PDP). Hubungan traceable ini terdokumentasi dalam packages/types pada monorepo, memastikan setiap perubahan requirement teknis tetap konsisten dengan kebutuhan operasional asal.

4.2 Functional Analysis and Design

Sub-fase ini berfokus pada pendefinisian arsitektur fungsional sistem dan bagaimana elemen-elemennya berinteraksi sebagai dasar untuk desain terperinci.

4.2.1 Critical Characteristic of System Functional Elements

Tabel 4.6 Karakteristik Kritis pada Fungsi Elemen Sistem Log-Shield

No	System Functions	Functional Elements	Critical Characteristics
1	Pemantauan Stok Fisik	Sensor Load Cell + HX711 + ESP32	Akurasi ±150 gram setelah median filtering, latensi MQTT < 10 detik, ketahanan IP67
2	Prediksi Kebutuhan AI	FastAPI + TinyTimeMixers + SHAP	MAE < 15%, latensi inferensi < 5 detik, retraining harian 02.00 WIB, Rationale Chips terbaca
3	Pencatatan Distribusi Mobile	Capacitor + React + SQLite	Alur 3 langkah, respons UI < 2 detik, durasi offline 72 jam, undo button 5 detik
4	Sinkronisasi Dua Arah	Custom Sync Endpoint + LWW Resolver	Sync 500 transaksi < 60 detik, deterministik conflict resolution, audit log lengkap
5	Keamanan dan Audit	AES-256 + RLS + Supabase Auth	Zero plain-text sensitif di DB, JWT 24 jam, 100% aksi teraudit, RBAC 4 tingkat
6	Dashboard Real-time	React + Recharts + Realtime	Polling 30 detik, rendering 10K points < 1 detik, push notifikasi < 1 detik stok kritis
7	KIB 16-digit Validation	Shared-types + DB constraint	CHECK constraint PostgreSQL + compile-time TypeScript validation

4.3 Functional Design

4.3.1 System Interaction

Interaksi antar sub-sistem Log-Shield mengikuti pola publish-subscribe dan request-response yang dirancang untuk meminimalkan coupling dan memaksimalkan resiliensi.

Alur Data Sensor (Use Case 1 — Pemantauan Stok): ESP32 membaca Load Cell setiap 30 detik → median filtering 15 sampel → publish MQTT QoS 1 ke topik logshield/warehouse/{id}/scale/{node}/reading → Broker Mosquitto meneruskan ke Backend Express (subscriber) → validasi shared-types → simpan ke tabel stock_readings Supabase. Jika bobot di bawah threshold kritis, Supabase Realtime memicu push notifikasi ke dashboard.

Alur Pencatatan Distribusi (Use Case 2 — Offline-First): Petugas membuka aplikasi mobile → alur 3 langkah Trauma-Aware (komoditas → kuantitas → KIB) → simpan ke SQLite lokal dengan synced=false → Capacitor Background Task Plugin trigger sync setiap 15 menit → POST /api/v1/sync ke backend dengan batch transaksi → backend

upsert ke Supabase distributions dengan LWW resolution → audit log otomatis tercatat. Petugas dapat melanjutkan operasi tanpa internet hingga 72 jam.

Alur Prediksi AI (Use Case 3 — Daily Retraining): Setiap pukul 02.00 WIB, AI Engine (FastAPI) dipicu oleh cron job → tarik data distributions dan stock_readings 30 hari terakhir dari Supabase via supabase-py → latih model TinyTimeMixers per posko → hitung SHAP values untuk fitur dominan → generasi Rationale Chips via template Bahasa Indonesia → simpan ke tabel predictions dengan model_version. Dashboard dan mobile membaca prediksi terbaru saat halaman dimuat.

4.4 Prototype Development

4.4.1 Mobile-App Design

Aplikasi mobile Log-Shield dirancang dengan prinsip Trauma-Aware UI/UX yang mempertimbangkan kondisi psikologis petugas BPBD di lapangan bencana. Prinsip desain mencakup: tombol minimal 48 dp untuk akses dengan sarung tangan, kontras warna rasio 7:1 untuk keterbacaan di bawah sinar matahari, tipografi sans-serif 18 sp pada body text, alur input maksimal 3 langkah untuk transaksi distribusi, undo button 5 detik sebelum commit final, dan indikator status sinkronisasi persisten di pojok atas setiap halaman.

4.4.2 Mobile-App Use Case

Use case utama aplikasi mobile Log-Shield:

- Login dengan NIP + PIN 6 digit atau biometrik (fingerprint Android, Face ID iOS).
- Melihat dashboard posko dengan summary stok terkini, prediksi besok, dan jumlah distribusi belum selesai.
- Mencatat distribusi logistik melalui alur 3 langkah Trauma-Aware (pilih komoditas → input kuantitas → konfirmasi KIB penerima).
- Update stok manual untuk komoditas yang tidak terpasang sensor IoT.
- Melihat detail prediksi AI per komoditas dengan Rationale Chips (2 fitur dominan untuk mobile).
- Melaporkan anomali (kerusakan komoditas, sensor tidak responsif, kondisi khusus pengungsi) dengan opsi foto via kamera.
- Memantau status sinkronisasi: jumlah dokumen pending, waktu sync terakhir, indikator koneksi.

Mockup wireframe Mobile App akan disisipkan pada saat finalisasi dokumen.

4.4.3 Website Design

Dashboard web Log-Shield dirancang sebagai pusat koordinasi untuk Koordinator BPBD tingkat kabupaten/provinsi dan Administrator BNPB pusat. Layout mengikuti pola sidebar navigasi kiri + main content area dengan 6 halaman utama yang dapat diakses cepat: Dashboard (summary cards, grafik, heatmap), Logistics (request dari posko), Assets (inventori gudang), Posko (data demografi), Personnel (manajemen RBAC), dan Settings (profil + parameter sistem + audit log).

4.4.4 Website Use Case

Use case utama dashboard web Log-Shield:

- Login Koordinator via email + password atau Google OAuth (Supabase Auth).
- Memantau 4 summary cards: Total Posko Aktif, Critical Items, Pending Requests, AI Health.
- Menganalisis grafik Daily Stock Weight per kategori komoditas (line chart Kebutuhan vs Persediaan).
- Melihat Regional Heatmap untuk identifikasi posko dengan kebutuhan kritis.
- Memantau Pemenuhan Distribusi Kelompok Rentan (Balita, Lansia, Ibu Hamil, Disabilitas).
- Memproses request logistik dari posko (alur 2 langkah: review → setujui/tolak).
- Mengelola data posko: tambah baru, update demografi, panggil PJ langsung via tombol CALL.
- Mengelola personnel via tabel RBAC: tambah akun, ubah role, hapus akun (audit otomatis).
- Mengkonfigurasi parameter sistem: frekuensi laporan, batas stok kritis, zona waktu.
- Mengekspor audit log untuk pelaporan BNPB pusat dan compliance UU PDP.

Mockup wireframe Website akan disisipkan pada saat finalisasi dokumen, mencakup 6 halaman utama yang telah didokumentasikan pada Deliverable 3.

4.4.5 IoT Design

Desain node IoT Log-Shield terdiri dari rangkaian elektronik minimalis: ESP32-WROOM-32 sebagai mikrokontroler utama (3.3V logic, Wi-Fi 2.4 GHz, RAM 520 KB, Flash 4 MB), modul ADC HX711 24-bit (gain 128x kanal A, eksitasi 2.7–5.5V), Load Cell 50 kg half-bridge dengan 3 kabel (Merah, Putih, Hitam), LED RGB WS2812 untuk diagnostik visual, tombol reset/tare di GPIO 0, dan voltage divider untuk monitoring tegangan baterai. Seluruh rangkaian ditempatkan dalam casing IP67 dengan gland waterproof untuk ketahanan terhadap debu dan kelembaban gudang.

4.4.6 IoT Use Case

Use case utama node IoT Log-Shield:

- Membaca bobot Load Cell setiap 30 detik dengan median filtering 15 sampel.
- Mempublikasikan data hasil pembacaan ke broker MQTT Mosquitto via Wi-Fi 2.4 GHz dengan QoS 1.
- Mengirim health report (RSSI, uptime, battery_mv, free_heap_kb, nvs_buffered) setiap 5 menit.
- Menerima dan mengeksekusi config commands dari backend (tare, recalibrate, ota, restart) via topik /config.
- Menyimpan buffer hingga 100 pembacaan ke flash NVS saat Wi-Fi terputus, mengirim ulang berurutan saat koneksi pulih.
- Menjalankan OTA firmware update dual-partition dengan rollback otomatis jika boot gagal.
- Menampilkan status operasional via LED RGB: biru (boot), hijau (OK), kuning (buffering), merah (fault), putih (OTA).

4.4.7 IoT Circuit Design

Skema rangkaian IoT Log-Shield mengikuti konfigurasi standar untuk Load Cell 3 kabel (half-bridge) dengan adaptasi non-konvensional yang diadopsi pada prototipe MVP.

Koneksi Load Cell → HX711: Kabel Merah ke pin E+ (excitation positif), Kabel Putih ke pin E- (excitation negatif), Kabel Hitam ke pin A- (signal reference). Pin A+ tidak digunakan. Konfigurasi ini berbeda dari skema standar (yang umumnya menggunakan A+ sebagai signal positif), namun terbukti berfungsi pada eksperimen prototipe dengan polaritas yang dapat dibalik melalui parameter calibration factor di firmware.

Koneksi HX711 → ESP32: Pin DT (Data) ke GPIO 16 ESP32, pin SCK (Serial Clock) ke GPIO 4, VCC ke 3V3, GND ke GND common. Pemilihan GPIO 16 dan 4 menghindari strapping pins ESP32 (GPIO 0, 2, 5, 12, 15) yang dapat memengaruhi boot mode jika digunakan.

Koneksi Tambahan: GPIO 23 ke LED WS2812 (data line, dengan resistor pull-up 330Ω), GPIO 0 ke tombol reset/tare (active LOW dengan pull-up internal), GPIO 34 ke voltage divider 2:1 untuk monitor tegangan baterai (input only ADC1_CH6), GPIO 35 ke deteksi sumber daya PLN/baterai (input only ADC1_CH7).

Diagram rangkaian Fritzing dan foto perangkat akan disisipkan pada saat finalisasi dokumen. Konfigurasi pin lengkap telah didokumentasikan pada Lampiran Konfigurasi Pin IoT Deliverable 2.

4.5 Development Testing

4.5.1 Test Planning

Pengujian pengembangan sistem Log-Shield dilaksanakan pada tanggal 25 Mei 2026 pada branch cobaan-v2 oleh tester Folfox. Metodologi pengujian mencakup unit testing pada modul backend (Jest) dan AI engine (Pytest), integration testing pada alur API end-to-end, serta functional testing pada seluruh fitur utama sistem. Lingkungan pengujian menggunakan Docker lokal dengan empat container aktif (logshield-couchdb, logshield-mosquitto, logshield-ai-engine, logshield-backend), web dashboard pada port 5173, dan mobile app pada port 5174.

4.5.2 Test Scenario

Bagian ini akan diisi setelah eksekusi pengujian. Skenario pengujian akan mencakup test cases yang telah didokumentasikan pada Deliverable 3 Bab 2.3.3, dengan hasil aktual yang menggantikan proyeksi pada saat finalisasi.

Tabel 4.7 Skenario Pengujian Sistem Log-Shield

No	Skenario Uji	Langkah Pengujian	Hasil yang Diharapkan	Status
1	Unit: Kalkulasi prediksi kebutuhan logistik (recalcForecastQty)	Dijalankan 10 test case dengan variasi kelas komoditas, populasi, dan threshold	Qty dihitung sesuai formula per kelas; tidak negatif; requestedQty dan criticalThreshold override berjalan	Pass
2	Unit: Analisis tren permintaan (computeTrendAnalysis)	Di-input distribusi historis dengan variasi pct7d dan pct30d	Tren rising/falling/stable terdeteksi dengan ambang batas ±20%	Pass
3	Unit: Generasi Rationale Chips (buildRecommendationChips)	Di-input kondisi stok, kelas komoditas, rasio rentan, dan tren	Chip relevan terbentuk; maksimal 3 chip; chip "rentan" muncul saat rasio > 15%	Pass
4	Unit: Normalisasi nama komoditas (normalizeItemName)	Di-input nama dengan variasi case, spasi, dan alias	Resolusi ke kode standar yang benar; unknown item di-slug	Pass
5	Unit: Validasi skema dokumen	Di-input dokumen user, posko, distribusi, prediksi, audit log dengan data valid dan invalid	Dokumen valid lolos; field wajib yang kosong dan nilai enum invalid ditolak	Pass
6	Unit: Enkripsi dan autentikasi	Hash NIK, enkripsi/dekripsi secret, normalisasi email	Hash deterministik; enkripsi reversibel; email lowercase dan trimmed	Pass

No	Skenario Uji	Langkah Pengujian	Hasil yang Diharapkan	Status
7	Integration: Health check endpoint	GET /api/health dan GET /api/couchdb/health	Mengembalikan info layanan dan status CouchDB	Pass
8	Integration: Autentikasi endpoint	POST /api/auth/login dengan kredensial kosong	Mengembalikan 401 Unauthorized	Pass
9	Integration: CRUD laporan anomali	Dibuat, baca, filter, update status, full lifecycle reported → investigating → resolved	Semua operasi CRUD berhasil; validasi field dan enum berjalan; endpoint tanpa auth ditolak 401	Pass
10	Integration: Quick request logistik	Dibuat request dengan kombinasi field valid dan invalid; verifikasi status awal	Request valid tersimpan dengan status menunggu; field wajib kosong ditolak 400; posko_id invalid ditolak 400	Pass
11	Integration: Stok dan audit log	GET summary dan kategori stok; POST pembacaan sensor IoT; POST audit log	Endpoint stok mengembalikan data; endpoint sensor dan audit log menerima data tanpa auth (sesuai desain IoT)	Pass

5. ENGINEERING DESIGN

5.1 Pendahuluan dan Ruang Lingkup

Fase Engineering Design merupakan tahapan ketiga dalam siklus hidup Systems Engineering menurut Kossiakoff (2011), yang menerjemahkan konsep sistem Log-Shield terpilih (Konsep B) menjadi rancangan teknis terperinci yang siap diimplementasikan.

5.1.1 Tujuan Pengembangan Sistem

Tujuan utama pengembangan Log-Shield adalah membangun sistem intelijen logistik kemanusiaan yang mampu memenuhi mandat Enam Tepat BNPB secara bersamaan melalui integrasi AI, IoT, dan arsitektur mobile offline-first. Sistem berkontribusi terhadap efisiensi distribusi bantuan bencana dengan menggantikan proses manual menjadi platform berbasis data yang prediktif, terenkripsi, dan auditable. Kemampuan utama mencakup prediksi kebutuhan logistik harian, validasi stok via Load Cell, pencatatan distribusi offline 72 jam, dan dashboard real-time.

5.1.2 Pengguna Utama dan Kebutuhan Spesifik

Tiga segmen pengguna: petugas lapangan BPBD (membutuhkan antarmuka sederhana, offline penuh, notifikasi jelas), koordinator logistik tingkat kabupaten/provinsi (membutuhkan dashboard real-time, visualisasi prediksi AI dengan Rationale Chips, laporan KIB-standardized), dan administrator BNPB pusat (membutuhkan manajemen RBAC, konfigurasi parameter, audit trail lengkap). Distribusi APK Android dan IPA iOS via internal distribution memenuhi kebutuhan kemudahan instalasi pada fase MVP.

5.1.3 Nilai Tambah Sistem Dibandingkan Sistem Konvensional

Lima keunggulan kompetitif Log-Shield: (1) arsitektur offline-first 72 jam yang mengatasi titik kegagalan sistem digital konvensional di area bencana; (2) prediksi kebutuhan berbasis AI yang mengubah paradigma dari reaktif menjadi proaktif, dengan Rationale Chips berbahasa Indonesia; (3) validasi stok fisik otomatis via sensor IoT yang mengeliminasi jeda informasi 2–6 jam; (4) enkripsi AES-256 dan Row Level Security sesuai UU PDP No. 27/2022; (5) standardisasi KIB 16-digit untuk interoperabilitas Satu Data Indonesia.

5.1.4 Batasan Ruang Lingkup Proyek

MVP 8 minggu dengan 5 anggota tim. Tidak termasuk: integrasi penuh sistem eksisting BNPB, modul manajemen keuangan bantuan, analitik jangka panjang pasca-bencana, deployment ke app store publik. Sensor IoT hanya pada titik stok strategis (komoditas prioritas tinggi). Anggaran hardware Rp 2.380.000. Cloud Supabase Free/Pro + DigitalOcean Droplet dengan operasional $43/bulan.

5.1.5 Asumsi Pengembangan Sistem

Asumsi pengembangan: ketersediaan Wi-Fi minimal intermittent di gudang logistik pusat, akses petugas ke Android 8.0+ atau iOS 15.0+, komponen hardware tersedia di pasaran lokal dengan pengiriman maksimal 7 hari, Supabase Free Tier mencukupi untuk MVP, data historis distribusi dapat diperoleh dari laporan BNPB, dan petugas lapangan bersedia mengikuti pelatihan singkat 30 menit.

5.2 Engineering Development Lifecycle

5.2.1 Concept Development

Concept Development Log-Shield telah melalui dua iterasi pada Deliverable 1 dan 2. Pada D1, Konsep B (Semi-Otomatis dengan Digital Monitoring) dipilih dengan skor weighted 4.70. Pada D2, konsep dimatangkan menjadi 10 spesifikasi fungsional (F01–F10), 20 kebutuhan teknis turunan (R-1 hingga R-20), dan 7 karakteristik kritis.

Keputusan arsitektur paling signifikan yang berevolusi dari D2 ke D3 adalah migrasi dari CouchDB ke Supabase (PostgreSQL managed) sebagai basis data utama, dan dari AWS ke DigitalOcean sebagai penyedia infrastruktur. Migrasi Supabase memberikan PostgreSQL managed + Auth + Realtime + RLS dalam satu platform. Migrasi DigitalOcean memberikan harga prediktif ($6–12/bulan per droplet) dan region Singapore latensi rendah. Untuk sinkronisasi offline mobile, PouchDB-CouchDB native replication digantikan dengan SQLite + custom sync endpoint.

5.2.1.1 Konsep Dasar Sistem

Sistem terdiri dari empat komponen utama: (1) node IoT ESP32 + Load Cell HX711 yang mengirim data via MQTT; (2) backend Node.js + Express yang menerima MQTT dari Mosquitto dan mengelola REST API + custom sync endpoint, berinteraksi dengan Supabase PostgreSQL; (3) AI engine Python FastAPI yang melatih TinyTimeMixers dan menghasilkan Rationale Chips via SHAP; (4) aplikasi klien berupa Capacitor.js mobile (Android+iOS) dan React web dashboard. Supabase menyediakan PostgreSQL + Auth + Realtime + RLS sebagai platform terpusat.

5.2.1.2 Studi Kelayakan (Feasibility Study)

Kelayakan teknis: seluruh komponen menggunakan teknologi matang open-source (ESP32 ratusan ribu proyek, TinyTimeMixers NeurIPS 2023, Supabase 1 juta+ proyek, Capacitor full-supported Ionic). Kelayakan ekonomi: Rp 2.380.000 hardware + $43/bulan cloud sangat terjangkau. Kelayakan operasional: Trauma-Aware UI 3 langkah dirancang untuk petugas non-teknis dalam kondisi stres tinggi.

5.2.1.3 Alternatif Solusi yang Dipertimbangkan

Empat alternatif dievaluasi pada D1: Konsep A (manual spreadsheet, skor 3.10), Konsep B (semi-otomatis, skor 4.70 — dipilih), Konsep C (otomasi penuh AI real-time, skor 2.30), Konsep D (SaaS pihak ketiga, skor 3.20). Konsep C tidak dipilih karena biaya

>500 juta dan timeline >6 bulan. Konsep D tidak dipilih karena data tersimpan di server luar negeri, tidak memenuhi UU PDP dan kedaulatan data Indonesia.

5.2.1.4 Keunikan Konsep Sistem

Tiga diferensiasi utama: (1) Offline-First Architecture dengan SQLite lokal + custom sync ke Supabase PostgreSQL, memungkinkan operasi penuh tanpa internet 72 jam; (2) Explainable AI via SHAP yang mengonversi output model menjadi Rationale Chips berbahasa Indonesia untuk petugas non-teknis; (3) KIB 16-digit sebagai universal key untuk interoperabilitas Satu Data Indonesia.

5.2.1.5 Pemenuhan Kebutuhan Pengguna

Petugas lapangan: Trauma-Aware UI 3 langkah dengan tombol 48dp, kontras 7:1, offline 72 jam, Rationale Chips. Koordinator: dashboard 6 halaman dengan stok real-time, prediksi, heatmap, request logistik, data posko, manajemen personnel. Administrator: Supabase Auth (RBAC), pengaturan parameter, audit log exportable.

5.2.2 Preliminary Design

5.2.2.1 Arsitektur Sistem Keseluruhan

Arsitektur tiga lapisan: (1) Lapisan Akuisisi Data — ESP32 + HX711 + MQTT over Wi-Fi; (2) Lapisan Pemrosesan dan Penyimpanan — Node.js + Express + Supabase PostgreSQL + Mosquitto + FastAPI; (3) Lapisan Presentasi — Capacitor.js mobile Android/iOS dengan SQLite + React web dashboard. Monorepo pnpm workspaces: apps/mobile, apps/dashboard, apps/backend, apps/ai-engine; packages/types, ui-core, utils; firmware/esp32. Diagram arsitektur akan disisipkan sebagai Gambar 5.1.

5.2.2.2 Subsistem Utama yang Dikembangkan

Lima sub-sistem: IoT/Sensor (ESP32, HX711, Load Cell, firmware Arduino C++); Backend (Express REST API, MQTT.js, Supabase client, AES-256, RBAC); AI Engine (FastAPI, TinyTimeMixers, SHAP, supabase-py); Mobile (Capacitor.js, React, SQLite, Background Sync Plugin); Web Dashboard (React, Recharts, TanStack Query, Supabase Realtime).

5.2.2.3 Interaksi Antar Subsistem

Alur utama: Sensor Load Cell → ESP32 (median filtering) → MQTT Broker Mosquitto → Backend Express (subscriber) → Supabase PostgreSQL ← Mobile SQLite (custom sync endpoint) ← Petugas Lapangan. Cabang paralel: AI Engine ↔ Supabase (batch read/write via supabase-py); Dashboard Web ← Backend REST API (polling 30 detik) + Supabase Realtime (push notifikasi kritis). Diagram interaksi akan disisipkan sebagai Gambar 5.2.

5.2.2.4 Teknologi dan Platform yang Digunakan

Framework utama Capacitor.js memungkinkan aplikasi berjalan sebagai web dan native (Android + iOS) dari satu codebase React. Backend Node.js + Express. Database Supabase (PostgreSQL managed) untuk Auth, Realtime, RLS dalam satu platform. AI engine Python + FastAPI + TinyTimeMixers. Infrastruktur DigitalOcean (Droplet Singapore) untuk harga prediktif dan antarmuka sederhana. Firmware ESP32 via Arduino IDE/PlatformIO.

5.2.2.5 Pertimbangan Desain Awal Frontend, Backend, dan Subsistem Lain

Frontend dirancang untuk situasi tekanan tinggi: pemilihan warna kontras tinggi, ukuran font besar (18+ sp), batasan input alur 3 langkah, peletakan tombol di posisi yang mudah dijangkau jempol. Backend dirancang dengan custom sync endpoint untuk kontrol penuh atas resolusi konflik LWW. Supabase RLS menambah keamanan pada level database di luar RBAC aplikasi. AI engine dipisahkan sebagai layanan independen agar dapat di-scale dan di-update tanpa memengaruhi komponen lain.

5.3 Detailed Design

5.3.1 Spesifikasi Teknis Komponen Hardware

Tabel 5.1 Spesifikasi Hardware Node IoT

Komponen	Spesifikasi Teknis	Keterangan
ESP32-WROOM-32	3.3V logic, Wi-Fi 2.4 GHz, RAM 520 KB, Flash 4 MB, 38-pin	Dukungan Wi-Fi native, GPIO cukup, ekosistem Arduino matang, ~Rp 65.000.
HX711 ADC	24-bit ADC, gain 128x kanal A, 2.7V–5.5V, 80 SPS	Resolusi tinggi untuk akurasi ±150g, kompatibel 3.3V.
Load Cell 50 kg	Half-bridge 3 kabel (M/P/H), eksitasi 5V, 1 mV/V	Kapasitas cukup untuk palet stok, tersedia lokal.
Casing IP67	Waterproof box 100×70×40 mm, tahan rendam 1 m 30 menit	Ketahanan banjir + kelembaban tinggi.
Adaptor 5V 2A	Input 220V AC, output 5V 2A	Sumber utama saat PLN tersedia.
Power Bank 20.000 mAh	USB-A 5V 2.1A, proteksi overcharge	Backup 24–36 jam pemadaman.
LED WS2812	RGB addressable, 3.3V–5V	Diagnostik visual: biru=boot, hijau=OK, kuning=buffer, merah=fault.

Konfigurasi pin utama: DT→GPIO 16, SCK→GPIO 4, VCC→3V3, GND common. Load Cell: Merah→E+, Putih→E-, Hitam→A- (konfigurasi non-standar yang terverifikasi pada prototipe). Pin tambahan: GPIO 23 (LED WS2812), GPIO 0 (tombol reset/tare), GPIO 34 (V_BAT ADC), GPIO 35 (deteksi sumber daya). Rincian lengkap pada Lampiran Konfigurasi Pin IoT Deliverable 2.

5.3.2 Arsitektur Software Frontend, Backend, dan Subsistem Lain

Lima lapisan software dengan Supabase PostgreSQL sebagai basis data terpusat dan Supabase Auth sebagai penyedia autentikasi.

Tabel 5.2 Arsitektur Software per Lapisan

Lapisan	Stack Teknis	Tanggung Jawab	Komunikasi
Mobile	Capacitor.js + React + Vite + SQLite + Tailwind + shadcn/ui	Antarmuka petugas (Android+iOS); pencatatan distribusi offline-first; Trauma-Aware UI.	Custom sync endpoint ke backend; Capacitor Background Task; SQLite local-first.
Web Dashboard	React + Vite + Recharts + TanStack Query + shadcn/ui	Dashboard koordinator; visualisasi time-series; Rationale Chips; ekspor laporan.	REST API polling 30 detik; Supabase Realtime untuk push stok kritis.
Backend	Node.js + Express + Supabase Client + MQTT.js + AES-256	REST API gateway; MQTT subscriber data IoT; enkripsi field; RBAC; custom sync.	Menerima MQTT dari Mosquitto; menyajikan REST JSON; berkomunikasi dengan Supabase via supabase-js.
AI Engine	Python + FastAPI + TinyTimeMixers + SHAP	Retraining harian; inferensi prediksi; generasi Rationale Chips.	REST API ke backend; akses Supabase PostgreSQL via supabase-py untuk batch read/write.
Firmware IoT	Arduino C++ (ESP-IDF) pada ESP32	Pembacaan HX711; median filtering; MQTT QoS 1; buffer NVS; OTA.	MQTT publish ke Mosquitto via Wi-Fi 2.4 GHz.

Pada lapisan mobile, Capacitor.js berperan sebagai native bridge yang membungkus aplikasi React menjadi aplikasi native untuk Android dan iOS. Dukungan iOS dimungkinkan secara native via npx cap add ios tanpa perlu kode Swift terpisah. Capacitor SQLite plugin menggantikan PouchDB sebagai penyimpanan lokal. Pada lapisan web, TanStack Query mengelola server-state dengan staleTime 15 detik dan refetchInterval 30 detik. Supabase Realtime digunakan sebagai kanal notifikasi push untuk status stok kritis. Dashboard memiliki enam halaman: Dashboard, Logistics, Assets, Posko, Personnel, Settings.

Pada lapisan backend, Express menggunakan supabase-js untuk berinteraksi dengan Supabase PostgreSQL. Supabase Auth menangani autentikasi (Google OAuth + JWT). Row Level Security (RLS) menambah lapisan keamanan pada level database. Custom sync endpoint (/api/v1/sync) mengelola sinkronisasi dua arah SQLite ↔ PostgreSQL dengan LWW timestamp hybrid. Pada lapisan AI engine, FastAPI di-deploy pada DigitalOcean Droplet terpisah, mengakses Supabase langsung via supabase-py untuk batch read data historis dan batch write hasil prediksi.

5.3.3 Skema Database Sistem

Log-Shield menggunakan Supabase sebagai platform basis data yang menyediakan PostgreSQL managed, autentikasi (Supabase Auth), Realtime, dan Row Level Security. Skema basis data dirancang dalam satu schema public dengan delapan tabel utama yang saling berelasi melalui foreign key. Seluruh tabel menggunakan UUID sebagai primary key untuk konsistensi dengan Supabase Auth, dan menyertakan field kib_16 (VARCHAR 16) sebagai identifier standar Satu Data Indonesia.

Berikut Entity Relationship Diagram terperinci setiap entitas dalam skema Supabase PostgreSQL Log-Shield:

Users

Tabel 5.3 *Entity Diagram users*

Column	Type	Description
id (PK)	UUID	Referensi ke auth.users(id) Supabase; dibuat otomatis saat registrasi.
email	VARCHAR(255)	Email untuk login via Google OAuth atau email/password.
name	VARCHAR(255)	Nama lengkap sesuai data kepegawaian BPBD/BNPB.
nik	VARCHAR(16)	NIK 16 digit; divalidasi CHECK constraint.
kib_bencana_id	VARCHAR(20)	KIB pengguna format BNC-YYYY-XX-NNNN.
role	ENUM	admin, koordinator, lapangan. Menentukan RBAC dan RLS.
posko_id (FK)	UUID	FK ke poskos; NULL untuk admin pusat.
phone	VARCHAR(20)	Nomor telepon untuk fitur CALL di halaman Posko.
avatar_url	TEXT	URL foto profil di Supabase Storage; opsional.
created_at	TIMESTAMPTZ	Waktu pembuatan akun; default NOW().
updated_at	TIMESTAMPTZ	Auto-trigger pada update.
Notes: Ekstensi auth.users Supabase. Menyimpan profil, role, dan assignment posko.		

Poskos

Tabel 5.4 *Entity Diagram poskos*

Column	Type	Description
id (PK)	UUID	Primary key unik posko.
kib_16	VARCHAR(16)	KIB 16-digit Satu Data Indonesia; UNIQUE.
name	VARCHAR(255)	Nama posko operasional.

Column	Type	Description
address	TEXT	Alamat administratif lengkap.
province	VARCHAR(100)	Provinsi untuk filter regional.
district	VARCHAR(100)	Kabupaten/kota.
total_pengungsi	INTEGER	Total pengungsi; default 0.
count_balita	INTEGER	Pengungsi 0–5 tahun.
count_lansia	INTEGER	Pengungsi >60 tahun.
count_perempuan	INTEGER	Pengungsi perempuan.
count_pria	INTEGER	Pengungsi pria.
count_disabilitas	INTEGER	Penyandang disabilitas.
pj_name	VARCHAR(255)	Nama PJ personel lapangan.
pj_phone	VARCHAR(20)	Telepon PJ untuk fitur CALL.
status	ENUM	active, inactive, closed. Default active.
created_at	TIMESTAMPTZ	Waktu pendaftaran.
updated_at	TIMESTAMPTZ	Waktu update terakhir.
Notes: Data posko pengungsian: lokasi, demografi 6 segmen, PJ lapangan.		

Distributions

Tabel 5.5 *Entity Diagram distributions*

Column	Type	Description
id (PK)	UUID	Client-generated UUID untuk mendukung offline.
kib_16	VARCHAR(16)	KIB 16-digit identifier transaksi.
posko_id (FK)	UUID	FK ke poskos; tempat distribusi.
officer_id (FK)	UUID	FK ke users; petugas pencatat.
commodity	VARCHAR(100)	Komoditas (beras, air mineral, dll).
quantity	DECIMAL(10,2)	Jumlah dalam satuan unit.
unit	VARCHAR(20)	Satuan: kg, liter, pcs, karton, kit.
recipient_kib	VARCHAR(16)	KIB pengungsi; AES-256 encrypted.
vulnerable_group	ENUM	umum, balita, lansia, ibu_hamil, disabilitas.

Column	Type	Description
notes	TEXT	Catatan petugas; opsional.
synced	BOOLEAN	false di SQLite lokal, true setelah sync.
created_at	TIMESTAMPTZ	Device timestamp pencatatan.
synced_at	TIMESTAMPTZ	Server timestamp sync; NULL jika belum.
Notes: Transaksi distribusi logistik. Dibuat di SQLite mobile, sync ke Supabase.		

Stock_readings

Tabel 5.6 *Entity Diagram stock_readings*

Column	Type	Description
id (PK)	UUID	Dibangkitkan backend saat menerima MQTT.
warehouse_id	VARCHAR(50)	Identifier gudang (WH-JKT-001).
node_id	VARCHAR(50)	Identifier node sensor (NODE-07).
commodity	VARCHAR(100)	Komoditas yang ditimbang.
weight_g	INTEGER	Bobot dalam gram setelah median filtering.
weight_delta_g	INTEGER	Selisih dari pembacaan sebelumnya.
sample_count	INTEGER	Sampel filtering (default 15).
rssi	INTEGER	RSSI Wi-Fi dalam dBm.
uptime_s	INTEGER	Uptime node dalam detik.
battery_mv	INTEGER	Tegangan baterai; NULL jika PLN.
timestamp	TIMESTAMPTZ	Device timestamp pembacaan.
created_at	TIMESTAMPTZ	Server timestamp penyimpanan.
Notes: Pembacaan sensor IoT Load Cell. Dikirim ESP32 via MQTT setiap 30 detik.		

Predictions

Tabel 5.7 *Entity Diagram predictions*

Column	Type	Description
id (PK)	UUID	Dibangkitkan AI engine.
posko_id (FK)	UUID	FK ke poskos.
commodity	VARCHAR(100)	Komoditas yang diprediksi.

Column	Type	Description
prediction_date	DATE	Tanggal yang diprediksi.
predicted_kg	DECIMAL(10,2)	Volume kebutuhan prediksi (kg).
confidence_low	DECIMAL(10,2)	Batas bawah CI 80%.
confidence_high	DECIMAL(10,2)	Batas atas CI 80%.
mae_last_7d	DECIMAL(5,2)	MAE model 7 hari terakhir.
shap_values	JSONB	SHAP values per fitur untuk Rationale Chips.
rationale_chips	JSONB	Array [{feature, narrative, shap_value}].
model_version	VARCHAR(50)	Versi model untuk tracking/rollback.
created_at	TIMESTAMPTZ	Waktu generasi prediksi.
Notes: Hasil prediksi AI engine. Diperbarui harian pukul 02.00 WIB.		

Requests

Tabel 5.8 *Entity Diagram requests*

Column	Type	Description
id (PK)	UUID	Primary key.
request_code	VARCHAR(30)	REQ-YYYYMMDD-NNN; auto-generated.
posko_id (FK)	UUID	FK ke poskos pemohon.
submitted_by (FK)	UUID	FK ke users pengaju.
items	JSONB	Array [{commodity, quantity, unit, note}].
status	ENUM	mendesak, menunggu, diproses, selesai.
priority	ENUM	critical, high, normal, low.
processed_by (FK)	UUID	FK ke users koordinator; NULL jika belum.
processed_at	TIMESTAMPTZ	Waktu pemrosesan; NULL jika belum.
created_at	TIMESTAMPTZ	Waktu pengajuan.
updated_at	TIMESTAMPTZ	Waktu update status terakhir.
Notes: Permintaan logistik dari posko. Diproses koordinator via dashboard.		

Assets

Tabel 5.9 *Entity Diagram assets*

Column	Type	Description
id (PK)	UUID	Primary key inventori.
warehouse_id	VARCHAR(50)	Identifier gudang.
commodity	VARCHAR(100)	Nama komoditas.
category	ENUM	sandang, pangan, papan, lainnya.
quantity_available	DECIMAL(10,2)	Stok tersedia saat ini.
unit	VARCHAR(20)	Satuan: kg, pcs, karton, unit.
min_threshold	DECIMAL(10,2)	Ambang kritis; di bawah ini trigger Critical Items.
last_sensor_weight_g	INTEGER	Bobot terakhir dari IoT; NULL jika tidak ada sensor.
last_sensor_update	TIMESTAMPTZ	Waktu sensor terakhir; NULL jika tidak ada.
created_at	TIMESTAMPTZ	Waktu pencatatan pertama.
updated_at	TIMESTAMPTZ	Waktu update terakhir.
Notes: Inventori gudang utama. Kategori BNPB: sandang, pangan, papan, lainnya.		

Audit_logs

Tabel 5.10 *Entity Diagram audit_logs*

Column	Type	Description
id (PK)	UUID	Primary key log entry.
user_id (FK)	UUID	FK ke users; NULL untuk aksi sistem otomatis.
action	VARCHAR(100)	Deskripsi aksi (create_distribution, login, dll).
target_table	VARCHAR(50)	Tabel yang terpengaruh.
target_id	UUID	PK record terpengaruh; NULL jika tidak spesifik.
old_values	JSONB	Snapshot sebelum perubahan; NULL untuk create.
new_values	JSONB	Snapshot setelah perubahan; NULL untuk delete.
ip_address	INET	IP pelaku untuk forensik.
status	ENUM	sukses, ditolak, timeout, error.
created_at	TIMESTAMPTZ	Waktu aksi; indexed; default NOW().
Notes: Jejak audit append-only. Retensi 2 tahun. Kepatuhan UU PDP No. 27/2022.		

Relasi antar tabel: users → poskos (many-to-one via posko_id); distributions → poskos dan users; predictions → poskos; requests → poskos, submitted_by → users, processed_by → users; audit_logs → users (opsional, ON DELETE SET NULL agar log tetap tersimpan). Foreign key lain menggunakan ON DELETE RESTRICT untuk mencegah penghapusan data terreferensi. Gambar 5.3 menampilkan ERD visual yang akan disisipkan pada finalisasi.

5.3.4 Rancangan Komunikasi Antar Komponen

Empat kanal komunikasi utama: (1) MQTT QoS 1 dari ESP32 ke backend via Mosquitto untuk data sensor; (2) REST JSON dari backend ke klien web dan mobile; (3) Custom sync endpoint POST /api/v1/sync dari mobile SQLite ke Supabase PostgreSQL; (4) Supabase Realtime dari server ke dashboard web untuk push notifikasi stok kritis. Format data seluruhnya JSON. Diagram alur komunikasi akan disisipkan sebagai Gambar 5.4.

5.3.5 Protokol Komunikasi

MQTT v5.0 dengan topik hierarkis logshield/warehouse/{id}/scale/{node}/{type}, QoS 1, payload JSON kompak. REST API Express terproteksi JWT Supabase Auth pada setiap endpoint. TLS 1.3 untuk seluruh komunikasi jaringan. Certificate pinning pada mobile. Supabase Realtime menggunakan WebSocket dengan filter berdasarkan tabel dan event type.

5.4 Production & Deployment

5.4.1 Rencana Pengadaan Komponen Hardware dan Lisensi Software

Komponen hardware dipesan dari toko daring lokal (Tokopedia/Shopee) pada minggu pertama. Total Rp 2.380.000 mencakup 3 unit ESP32, 3 set HX711+Load Cell, 3 casing IP67, kabel/PCB, dan 2 power bank. Seluruh software open-source: React (MIT), Express (MIT), Capacitor (MIT), TinyTimeMixers (Apache 2.0), Supabase (Apache 2.0). Supabase Cloud Free Tier untuk MVP; Pro ($25/bulan) untuk produksi.

5.4.2 Strategi Integrasi dan Pengujian Subsistem

Integrasi inkremental: unit test per modul → integrasi sub-sistem (IoT+Backend, Mobile+Backend, AI+Backend) → integrasi total end-to-end. Metodologi: Jest (TypeScript), Pytest (Python), PlatformIO (ESP32), Playwright (E2E), k6 (load test).

5.4.3 Rencana Deployment dari Prototipe ke Implementasi Penuh

Tabel 5.11 Strategi Deployment per Komponen

Komponen	Metode Deployment	Environment	Strategi Rollback
Backend + Mosquitto	Docker Compose pada DO Droplet $12/bulan	Production SGP1; Staging lokal	Versioned Docker image; rollback < 5 menit.

Komponen	Metode Deployment	Environment	Strategi Rollback
Supabase PostgreSQL	Supabase Cloud (Pro $25/bulan)	Region Southeast Asia	Point-in-time recovery; backup harian otomatis.
AI Engine	Docker container pada DO Droplet $6/bulan	Python 3.11 + FastAPI	Model versioning di Supabase Storage; rollback tanpa downtime.
Dashboard Web	Static build (Vite) di DO App Platform	CDN DigitalOcean	Versioned build; rollback via re-deploy.
Mobile App	APK + IPA via internal distribution	Android 8.0+; iOS 15.0+	Versioned binary; petugas unduh via OTA link.
Firmware ESP32	Binary flash USB awal; OTA selanjutnya	Arduino IDE/PlatformIO	Dual partition; rollback otomatis jika boot gagal.

Estimasi biaya operasional bulanan: DO Backend $12 + DO AI $6 + Supabase Pro $25 + DO App Platform Static $0 = $43/bulan (~Rp 680.000). Jauh lebih prediktif dibandingkan AWS dan mencakup managed database, autentikasi, serta CDN tanpa biaya tersembunyi.

5.4.4 Lingkungan Pengujian Sebelum Deployment

Tiga environment: Development (lokal laptop anggota tim), Staging (DO Droplet terpisah + Supabase project staging), Production (Droplet SGP1 + Supabase project produksi). Field testing dengan beban referensi terkalibrasi untuk sensor; emulator Android + iOS Simulator untuk mobile.

5.4.5 Rencana Menjamin Kualitas Manufaktur dan Assembly Komponen

Checklist assembly per node IoT: (1) solder HX711 ke Load Cell, verifikasi kontinuitas; (2) koneksi ESP32↔HX711 via GPIO 4 dan 16, verifikasi dengan multimeter; (3) flash firmware, kalibrasi dengan beban 10 kg referensi; (4) verifikasi MQTT publish ke broker; (5) install ke casing IP67, seal gland waterproof; (6) test ketahanan air 30 detik perendaman. Dokumentasi foto setiap langkah.

5.5 Operations & Support

5.5.1 Mekanisme Monitoring dan Pemeliharaan Sistem Pasca Implementasi

Monitoring menggunakan Supabase Dashboard bawaan (koneksi PostgreSQL, query performa, auth events) dan DigitalOcean Monitoring (CPU, memory, disk Droplet). Logging error via structured logging Express (Winston). OTA update firmware ESP32 via binary di Supabase Storage. Kalibrasi sensor otomatis setiap 6 jam; manual setiap 30 hari.

5.5.2 Rencana Upgrade Sistem di Masa Depan

Fitur pasca-MVP: integrasi AI analytics jangka panjang, sensor tambahan (DHT22 suhu/kelembaban untuk komoditas perishable), optimasi model ke TinyTimeMixers, integrasi penuh API BNPB, deployment ke Google Play Store dan Apple App Store.

5.5.3 Strategi Menghadapi Kegagalan Sistem

Fallback manual: petugas menggunakan formulir kertas jika mobile gagal total. Deteksi kegagalan: node IoT tidak responsif 60 menit memicu notifikasi; MAE AI > 20% memicu fallback rata-rata historis. Recovery: Supabase point-in-time recovery; Docker rollback < 5 menit. Backup: Supabase harian otomatis; audit log append-only.

5.5.4 Dokumentasi Teknis untuk Dukungan Operasional

Dokumentasi mencakup: API docs (Swagger/OpenAPI pada /api/docs), manual instalasi node IoT (assembly + kalibrasi), panduan troubleshooting (FAQ kegagalan sensor, sync gagal, login error), prosedur maintenance (kalibrasi, OTA, rotasi kunci enkripsi).

5.5.5 Pelatihan Pengguna Sistem

Tiga format: (1) Workshop 30 menit untuk petugas lapangan (alur 3 langkah + indikator sync); (2) Workshop 2 jam untuk koordinator (dashboard, Rationale Chips, ekspor laporan); (3) Workshop 4 jam untuk administrator (RBAC, konfigurasi, eskalasi). Media: video tutorial, quick reference card laminated (tahan air) untuk petugas lapangan.

5.6 Requirement Analysis

5.6.1 Kebutuhan Fungsional

Tabel 5.12 Kebutuhan Fungsional Sistem

Kode	Deskripsi	Kriteria Penerimaan
F01	Prediksi kebutuhan logistik harian per posko (TinyTimeMixers)	MAE < 15%; latensi < 5 detik; retraining harian.
F02	Rationale Chips berbasis SHAP dalam Bahasa Indonesia	3 fitur tertinggi; narasi dipahami non-teknis.
F03	Pemantauan stok otomatis Load Cell setiap 30 detik	Akurasi ±150 gram; latensi E2E < 10 detik; MQTT QoS 1.
F04	Offline penuh mobile (Android + iOS) 72 jam	SQLite SSOT lokal; zero data loss.
F05	Sinkronisasi dua arah SQLite–Supabase	Custom sync 500 transaksi < 60 detik; LWW.
F06	Dashboard 6 halaman untuk koordinator	Polling 30 detik; Realtime push notifikasi kritis.
F07	KIB 16-digit sebagai identifier standar	CHECK constraint PostgreSQL; compile-time shared-types.
F08	Enkripsi AES-256 + TLS 1.3 + RLS PostgreSQL	Zero plain-text sensitif; RLS aktif semua tabel.
F09	RBAC empat tingkat + Supabase Auth + audit trail	JWT; 100% aksi teraudit; HTTP 403 unauthorized.
F10	Background Sync mobile via Capacitor Plugin	Sync 15 menit Android; best-effort iOS.

5.6.2 Mekanisme Penentuan Logika Utama Sistem

Tiga mekanisme utama: (1) Prediksi AI — setiap malam 02.00 WIB, AI engine ambil data 30 hari terakhir, latih TinyTimeMixers per posko, simpan prediksi 7 hari ke depan + SHAP. Fallback rata-rata historis 30 hari jika MAE > 20%. (2) Deteksi stok kritis — backend bandingkan stock_reading dengan assets.min_threshold; jika di bawah, Supabase Realtime push notifikasi ke dashboard. (3) Resolusi konflik sync — custom sync endpoint terima batch dari mobile, bandingkan timestamp setiap record, revisi created_at terbaru menang (LWW), revisi kalah dicatat di audit_logs.

5.6.3 Parameter yang Diukur/Diolah

Parameter utama: bobot stok fisik (gram, dari Load Cell HX711, akurasi ±150g); volume distribusi harian (kg/pcs/karton dari input petugas); demografi pengungsi per posko (6 segmen integer); prakiraan kebutuhan logistik (kg, dari TinyTimeMixers); SHAP values per fitur (float); RSSI Wi-Fi node IoT (dBm); tegangan baterai backup (mV); uptime node (detik).

5.6.4 Fitur Utama pada Frontend, Backend, dan Komponen Lain

Tabel 5.13 Fitur Utama per Komponen

Komponen	Fitur Utama
Frontend/Dashboard	Summary cards (Total Posko, Critical Items, Pending Requests, AI Health); Daily Stock Weight chart; Regional Heatmap; Pemenuhan Kelompok Rentan; Halaman Logistics; Halaman Assets; Halaman Posko (demografi 6 segmen + CALL); Halaman Personnel (RBAC); Halaman Settings (profil, notifikasi, parameter, audit log).
Backend/API	REST API JSON (distribusi, stok, prediksi, posko, users, request, audit); Custom sync endpoint /api/v1/sync (batch upsert + LWW); MQTT subscriber data IoT; AES-256 field encryption; RBAC middleware; Supabase client; audit trail otomatis.
Aplikasi Mobile	Pencatatan distribusi 3 langkah Trauma-Aware; Prediksi AI + Rationale Chips; Update stok manual; Laporan anomali + kamera; Indikator sync persisten; Background Sync (Android + iOS); Offline 72 jam SQLite.
IoT/Firmware	HX711 setiap 30 detik; Median filtering 15 sampel; Buffer NVS 100 pembacaan saat offline; MQTT publish QoS 1; OTA firmware dual-partition; LED status diagnostik; Health reporting.

5.6.5 Penanganan Kondisi Abnormal

Sensor tidak valid: firmware deteksi pembacaan di luar rentang wajar (< -500g atau > 60.000g), tandai sebagai anomali tanpa publish MQTT. Koneksi terputus pada mobile: operasi berlanjut normal di SQLite lokal; indikator sync kuning; Background Sync coba setiap 15 menit. Kegagalan node IoT: backend deteksi node tidak melaporkan > 60 menit, notifikasi ke administrator; prediksi AI beralih ke estimasi historis. Mode fallback AI: jika MAE > 20%, model digantikan rata-rata historis 30 hari nasional. Kegagalan backend: mobile tetap offline; dashboard tampil pesan koneksi terputus; auto-retry eksponensial.

5.6.6 Kebutuhan Non-Fungsional

5.6.6.1 Persyaratan Performa Sistem

Respons UI mobile < 2 detik; throughput API ≥ 100 req/detik; interval sensor 30 detik; kapasitas Supabase Free Tier 500 MB database, 50.000 rows; migrasi ke Pro (8 GB, unlimited) saat produksi.

5.6.6.2 Persyaratan Keamanan

Supabase Auth (JWT + Google OAuth); AES-256 field encryption data sensitif; TLS 1.3 seluruh endpoint; RLS PostgreSQL per tabel; certificate pinning mobile; rate limiting Express (100 req/menit per IP); audit trail append-only.

5.6.6.3 Persyaratan Keandalan dan Ketersediaan

Target uptime ≥ 95% per bulan; failover: arsitektur offline-first mengurangi ketergantungan cloud; retry otomatis pada sync dan MQTT; Supabase SLA 99.9% (Pro).

5.6.6.4 Persyaratan Efisiensi Energi

ESP32 deep sleep saat idle (< 10 µA); mode aktif < 500 mA @ 5V; interval polling MQTT 30 detik (bukan continuous); power bank 20.000 mAh backup 24–36 jam.

5.6.6.5 Persyaratan Skalabilitas

Backend Express horizontal scaling via DO Load Balancer; Supabase PostgreSQL connection pooling (PgBouncer bawaan); API modular per resource; device management hingga 100 node IoT per instance backend; database indexing pada created_at dan posko_id.

5.7 Risk Assessment

5.7.1 Identifikasi Risiko Teknis Utama

Risiko teknis utama: ketidakakuratan sensor Load Cell di kondisi vibrasi/suhu ekstrem, model AI tidak konvergen dengan data historis terbatas, konflik sinkronisasi data dari banyak perangkat offline, kegagalan konektivitas Wi-Fi di area bencana, dan kerentanan keamanan pada endpoint API.

5.7.2 Risiko Keandalan Komponen

HX711 dan Load Cell berpotensi mengalami drift akurasi seiring waktu akibat fatigue mekanis dan korosi. ESP32 memiliki MTBF > 8.760 jam, namun kelembaban tinggi dapat mempercepat degradasi. Koneksi Wi-Fi pada ESP32 dapat terputus intermittent di lingkungan dengan banyak interferensi elektromagnetik.

5.7.3 Risiko Keamanan

Penyadapan data distribusi dan data pribadi pengungsi saat transmisi; serangan MITM pada sinkronisasi mobile; injection attack pada REST API; pencurian token JWT; akses tidak sah ke Supabase dashboard jika kredensial admin bocor.

5.7.4 Risiko Integrasi Antar Subsistem

Ketidakselarasan skema antara SQLite mobile dan PostgreSQL Supabase; inkompatibilitas versi Capacitor plugin antara Android dan iOS; latensi tinggi pada custom sync endpoint saat batch besar; kegagalan Supabase Realtime WebSocket di jaringan yang memblokir WebSocket.

5.7.5 Risiko Lingkungan Operasional

Suhu ekstrem > 55°C di gudang atap seng; kelembaban > 95% RH di area banjir; debu vulkanik pada bencana erupsi; pemadaman listrik melebihi kapasitas power bank; kerusakan fisik node IoT akibat bongkar-muat di gudang.

5.7.6 Assessment Risiko

Tabel 5.14 Risk Assessment

Risiko	Probabilitas	Dampak	Level Risiko	Kode
Sensor tidak akurat di lingkungan ekstrem	Sedang	Tinggi	Mayor	RSK-01
Model AI tidak konvergen (data < 21 hari)	Tinggi	Tinggi	Kritis	RSK-02
Konflik data sinkronisasi bersamaan	Sedang	Tinggi	Mayor	RSK-03
Kebocoran data pribadi pengungsi	Rendah	Tinggi	Mayor	RSK-04
Downtime Supabase/DigitalOcean	Rendah	Tinggi	Mayor	RSK-05
Penolakan adopsi petugas lapangan	Sedang	Tinggi	Mayor	RSK-06
Kegagalan broker MQTT	Rendah	Sedang	Minor	RSK-07
Timeline 8 minggu terlampaui	Sedang	Tinggi	Mayor	RSK-08
Inkompatibilitas iOS Capacitor plugin	Sedang	Sedang	Minor	RSK-09
Injeksi/serangan pada REST API	Sedang	Tinggi	Mayor	RSK-10

5.7.7 Dampak Kegagalan Komponen dan Sistem

Kegagalan sensor: data stok tidak terbaca, distribusi berbasis estimasi manual, prediksi AI kurang akurat. Kegagalan backend: mobile tetap offline, dashboard tidak diakses, data

tidak tersinkronisasi. Kegagalan Supabase: seluruh operasi online terhenti, mobile hanya beroperasi offline. Kegagalan AI engine: prediksi tidak diperbarui, petugas pakai Rationale Chips terakhir yang ter-cache. Kegagalan mobile: petugas pakai formulir kertas sebagai fallback.

5.7.8 Strategi Mitigasi Risiko

5.7.8.1 Mitigasi Risiko Komponen/Hardware

Kalibrasi otomatis 6 jam + manual 30 hari; median filtering 15 sampel; redundansi sensor cadangan per gudang; casing IP67; validasi pembacaan di firmware.

5.7.8.2 Mitigasi Risiko Keamanan

AES-256 field encryption; TLS 1.3; RLS PostgreSQL; JWT Supabase Auth; rate limiting; certificate pinning mobile; OWASP ZAP scan; audit trail; rotasi kunci enkripsi berkala.

5.7.8.3 Mitigasi Risiko Koneksi/Offline

Arsitektur offline-first (SQLite 72 jam); buffer NVS 100 pembacaan ESP32; auto-reconnect eksponensial; custom sync endpoint dengan batch delta; Capacitor Background Task Plugin.

5.7.8.4 Mitigasi Risiko Lingkungan

Casing IP67; conformal coating PCB; heat shield opsional; kabel shielded < 1 meter Load Cell ke HX711; gland waterproof M12.

5.7.8.5 Strategi Cadangan Sistem

Override manual via formulir kertas + data entry retroaktif; notifikasi peringatan node tidak responsif > 60 menit; health check setiap 5 menit; remote diagnostic via topik MQTT /health; Supabase point-in-time recovery.

5.7.9 Residual Risk Management

Risiko residual: akurasi sensor pada vibrasi sangat tinggi mungkin melebihi ±150g; Background Task iOS tidak dapat dijamin intervalnya oleh Apple; data historis < 21 hari menghasilkan MAE > 15%. Monitoring berkelanjutan: dashboard Supabase, DO Monitoring, audit log review mingguan. Kontingensi: upgrade ke sensor industrial jika MVP menunjukkan akurasi kurang; fallback sync manual jika Background Task iOS tidak reliabel.

5.8 Configuration Management & Change Control

5.8.1 Version Control

Seluruh kode dikelola dalam Git monorepo dengan pnpm workspaces. Branching: main (production), staging (pre-production), feature/* (development). Firmware ESP32

di-version dengan semver (v1.0.0). Model AI di-version di Supabase Storage (model-v1.0-20260415.pkl). Rollback: Docker image untuk backend; dual-partition OTA untuk firmware; Supabase point-in-time recovery untuk database.

5.8.2 Change Requests

Proses formal: developer membuat GitHub Issue dengan label change-request, dokumentasikan dampak terhadap modul lain, dapat approval dari PM, lalu buat pull request. Changelog otomatis dari conventional commits. Prioritas: critical (hotfix ke main), high (sprint berikutnya), normal (backlog).

5.8.3 Impact Analysis

Setiap change request dianalisis dampaknya: perubahan di packages/types otomatis trigger tsc --noEmit di seluruh apps/ (compile-time impact); perubahan skema database butuh migration file Supabase; perubahan firmware butuh OTA deploy bertahap (1 node dulu, lalu sisanya jika berhasil).

5.8.4 Approval & Documentation

Approval: minimal 1 reviewer selain penulis pada pull request. Dokumentasi: setiap merge ke main otomatis update CHANGELOG.md. Komunikasi: notifikasi di grup WhatsApp tim untuk setiap merge ke main dan deployment ke staging/production.

5.9 Project Management

5.9.1 Timeline & Milestones

Tabel 5.15 Timeline Proyek

Fase	Deskripsi Aktivitas	Durasi	Milestone
1	Initiating: finalisasi requirements, setup monorepo + CI/CD, pengadaan IoT, setup Supabase	Minggu 1	Monorepo siap; Supabase project dibuat; komponen dipesan.
2	Design: arsitektur, skema PostgreSQL migration, wireframe, API contract	Minggu 2–3	Arsitektur approved; Supabase migration berjalan; wireframe validated.
3	Sprint 1: prototyping IoT, backend + Supabase, kerangka mobile (Android+iOS) + SQLite	Minggu 4	Node IoT baca+MQTT; backend simpan Supabase; mobile offline OK.
4	Sprint 2: AI training, dashboard web 6 halaman, fitur mobile lengkap	Minggu 5	TinyTimeMixers trained; dashboard live; mobile distribusi offline.
5	Sprint 3: integrasi E2E, RBAC+RLS, enkripsi, Rationale Chips, sync endpoint	Minggu 6	Alur IoT→Backend→Supabase→Mobile OK; keamanan aktif.
6	Integration Testing: sync, LWW, load test, iOS testing	Minggu 7	12 skenario dieksekusi; iOS validated; bug kritis fixed.

Fase	Deskripsi Aktivitas	Durasi	Milestone
7	Stabilisasi, demo MVP, dokumentasi akhir	Minggu 8	Demo berhasil; Deliverable diserahkan.

5.9.2 Resource Allocation

Tabel 5.16 Alokasi Sumber Daya Tim

Nama — NIM	Tugas / Modul
Andhika M. Addiputra — 18223005	PM; kepatuhan UU PDP; KIB; Capacitor mobile (Android+iOS); SQLite; Background Sync; Trauma-Aware UI.
Ferro Arka Berlian — 18223027	AI: TinyTimeMixers, SHAP, Rationale Chips, retraining harian, FastAPI, supabase-py batch.
Luckman Fakhmanidris A. — 18223041	IoT: ESP32+HX711 firmware, median filtering, buffer NVS, MQTT QoS 1, OTA, kalibrasi, IP67.
M. Naufal Fathan — 18223059	Mobile: React+Capacitor, SQLite integration, halaman distribusi/prediksi/anomali, iOS build, sync.
Atharizza M. Athaya — 18223079	Backend: Express API, Supabase client+RLS, MQTT subscriber, AES-256, custom sync, DigitalOcean deploy.

5.9.3 Quality Assurance

Type-safety lintas platform via packages/types; CI GitHub Actions + Turborepo paralel; coverage minimum 75% pada paket kritis; code review wajib per PR; iOS build validated via Xcode; deployment ke production via manual approval setelah staging smoke test.

5.9.4 Communication

Internal: daily standup 15 menit via WhatsApp; weekly sprint review via Google Meet; task tracking di GitHub Projects (kanban). Repositori: GitHub monorepo dengan branch protection. Eksternal: update ke dosen pembimbing setiap 2 minggu; dokumentasi resmi via Deliverable 1–4.

6. INTEGRATION AND EVALUATION

Bab ini memaparkan rencana integrasi dan evaluasi sistem Log-Shield. Bagian perencanaan dan integrasi sistem disajikan secara penuh sebagai rencana terstruktur, sementara hasil pengujian dan evaluasi operasional akan diisi oleh tim pengembang setelah eksekusi pengujian pada minggu ke-7 dan ke-8 fase MVP.

6.1 Perencanaan dan Persiapan Pengujian (Test Planning and Preparation)

Bagian ini berisi tabel paralel antara System Development dan T&E Planning untuk menyelaraskan pengembangan sistem dengan strategi pengujian yang komprehensif.

Tabel 6.1 Pengembangan Sistem dan Perencanaan T&E

System Development	T&E Planning
Need: Memverifikasi bahwa sistem Log-Shield memenuhi mandat Enam Tepat BNPB: prediksi akurat, distribusi tepat sasaran, validasi stok otomatis, operasi offline, keamanan data UU PDP No. 27/2022.	Objective: Memastikan sistem beroperasi sesuai spesifikasi teknis (MoP), handal dalam kondisi bencana, aman dari serangan, dan mudah digunakan oleh petugas non-teknis di lingkungan tekanan tinggi.
System Concept: Trade-off performa vs biaya (Supabase Free vs Pro); kecepatan dev vs kompleksitas (monorepo vs multi-repo); akurasi AI vs latensi (cloud inference vs edge).	Test Concept: Evaluasi trade-off manual vs otomatis vs simulasi; jadwal pengujian di minggu 7–8; cakupan: unit, integrasi, end-to-end, load, security, field test.
Functional Design: Modul sensor/IoT, backend Express API, AI engine FastAPI, mobile Capacitor (Android+iOS), web dashboard React, broker MQTT Mosquitto, Supabase PostgreSQL.	Test Plan: Unit test per modul (Jest/Pytest/PlatformIO); integrasi sub-sistem (IoT+Backend, Mobile+Backend, AI+Backend); API security + load (k6 + OWASP ZAP); UI E2E (Playwright); field test sensor; evaluasi operasional bersama petugas BPBD.
Detailed Design: Hardware (ESP32+HX711+IP67); Software (Capacitor, Express, FastAPI, Supabase); Offline (SQLite + custom sync); Keamanan (AES-256, RLS, JWT, TLS 1.3).	Test Procedures: Kalibrasi + validasi sensor 4 titik beban; uji sync 72 jam offline; validasi LWW conflict; load test 100 node IoT paralel; penetration test OWASP; iOS build validation; field test kondisi lingkungan.
Asumsi dan Batasan Sistem: Wi-Fi intermittent di gudang; Android 8+/iOS 15+; komponen tersedia lokal; Supabase Free Tier cukup untuk MVP; petugas non-teknis bersedia ikut pelatihan 30 menit.	Peninjauan dan Penyesuaian Rencana Pengujian: Review persyaratan per sprint; penyesuaian skenario jika asumsi berubah; buffer waktu 1 minggu di akhir untuk iterasi.
Risiko dan Mitigasi Teknis: Sensor drift, AI data terbatas, konflik sync, keamanan API, timeline ketat 8 minggu, kompatibilitas iOS plugin.	Penanganan Risiko Pengujian: Simulasi kegagalan sensor; inject konflik data dua perangkat; penetration test OWASP; stress test lonjakan data; monitoring log real-time via Supabase + DigitalOcean Monitoring.
Timeline dan Alokasi Sumber Daya: 7 fase pengembangan, 8 minggu, 5 anggota tim dengan spesialisasi terdiversifikasi (PM, AI, IoT, Mobile, Backend).	Jadwal dan Alokasi Sumber Daya Pengujian: Minggu 7 developmental testing (12 skenario teknis); Minggu 8 OT&E + demo MVP; target keterlibatan 5 petugas BPBD representatif sebagai penguji eksternal.

System Development	T&E Planning
Dokumentasi dan Komunikasi: Monorepo Git, CHANGELOG conventional commits, API docs Swagger pada /api/docs, weekly sprint review via Google Meet.	Pelaporan dan Dokumentasi Hasil Pengujian: Laporan pengujian per skenario; bug tracker GitHub Issues dengan severity labels; rekomendasi perbaikan; feedback pengguna OT&E direkap dalam tabel.

6.2 Integrasi Sistem (System Integration)

Integrasi sistem Log-Shield dilakukan secara inkremental sesuai rekomendasi Kossiakoff (2011), terdiri dari empat tahap: konfigurasi pengujian fisik, proses integrasi sub-sistem, integrasi total sistem, dan fasilitas pengujian integrasi sistem.

6.2.1 Konfigurasi Pengujian Fisik

Konfigurasi pengujian fisik dirancang untuk mensimulasikan kondisi operasional nyata di gudang logistik BPBD. Diagram konfigurasi pengujian fisik akan disisipkan sebagai Gambar 6.1 pada saat finalisasi.

Tabel 6.2 Detail Konfigurasi Pengujian Fisik

Komponen / Alur	Deskripsi	Peran dalam Sistem
Input Generator	Beban referensi terkalibrasi 100-1000 gr yang menghasilkan stimulus fisik terukur pada Load Cell.	Ground truth untuk validasi akurasi sensor Load Cell HX711 (target deviasi ±150 gram).
System Element	Node IoT lengkap: Load Cell → HX711 (ADC 24-bit) → ESP32 (firmware median filtering, MQTT publish QoS 1) → Wi-Fi → Broker Mosquitto.	Keseluruhan pipeline akuisisi data yang diuji sebagai satu unit fungsional terintegrasi.
Output Analyzer	Script Python yang subscribe MQTT, catat timestamp publish dan timestamp simpan di Supabase, bandingkan dengan beban referensi.	Validasi akurasi, latensi end-to-end, dan message delivery rate (zero drop QoS 1).
Element Model	Model simulasi berdasarkan spesifikasi teknis HX711 (24-bit ADC, gain 128x) yang memprediksi output ideal untuk setiap beban referensi.	Referensi pembanding untuk menentukan apakah deviasi aktual berada dalam toleransi MoP1.
Performance Comparator	Script yang menghitung deviasi, latensi p95, dan delivery rate; membandingkan hasil aktual dengan target MoP yang ditetapkan pada Deliverable 1.	Penilaian pass/fail terhadap kriteria keberhasilan yang ditetapkan.
Test Manager	Project Manager (Andhika) yang mengatur jadwal, menetapkan kriteria, dan memutuskan tindak lanjut jika hasil di luar toleransi.	Pengawas dan pengambil keputusan eskalasi untuk seluruh proses pengujian.
Test Control Unit	Script otomasi yang memicu skenario uji (vibrasi, suhu, Wi-Fi off, lonjakan data) secara terjadwal.	Kontrol terpusat untuk reprodusibilitas skenario pengujian lintas iterasi.

6.2.2 Proses Integrasi Sistem

Proses integrasi sub-sistem dilakukan bertahap dengan setiap sub-sistem diuji secara terisolasi sebelum integrasi total. Diagram Subsystem Test Configuration akan disisipkan sebagai Gambar 6.2 pada saat finalisasi.

Tabel 6.3 Detail Subsystem Test Configuration

Komponen	Deskripsi Fungsi	Implementasi	Peran
Input Generator	Data uji menyerupai kondisi nyata.	Beban fisik (untuk sensor); data JSON simulasi (untuk API); transaksi dummy (untuk mobile sync).	Pengujian terkontrol dan berulang dengan parameter konsisten.
Subsystem A (IoT+Backend)	Pipeline sensor ke database PostgreSQL.	ESP32 → MQTT → Express subscriber → Supabase stock_readings.	Validasi alur akuisisi data sensor end-to-end.
Subsystem B (Mobile+Backend)	Sync offline ke database PostgreSQL.	SQLite → custom sync endpoint → Supabase distributions.	Validasi offline-first 72 jam dan LWW conflict resolution.
Subsystem C (AI+Backend)	Pipeline prediksi dan generasi Rationale Chips.	Supabase data → FastAPI TinyTimeMixers → Supabase predictions.	Validasi akurasi MAE < 15% dan koherensi narasi chips.
Output Analyzer	Kumpulkan dan olah output sub-sistem.	Query Supabase SQL editor; subscribe MQTT Explorer; Playwright trace files.	Verifikasi integritas data end-to-end pasca-integrasi.
Performance Comparator	Bandingkan aktual vs target MoP.	Script k6 + custom assertions terhadap MoP1–MoP8.	Indikator pass/fail integrasi per sub-sistem.
Test Manager	Pengawasan dan keputusan eskalasi.	PM + jadwal sprint + monitoring + laporan ke dosen.	Jaminan pengujian efektif dan dapat dipertanggungjawabkan.

6.2.3 Integrasi Total Sistem

Tabel 6.4 Analisis Konteks Pengintegrasian Sistem secara Total

Aspek	Penjelasan dan Implikasi
Penggabungan semua subsistem	Ketiga sub-sistem (A, B, C) diintegrasikan pada environment staging tunggal: sensor IoT → backend → Supabase ← mobile sync ← AI engine; dashboard web mengkonsumsi seluruh data dari Supabase.
Kompleksitas tinggi integrasi sistem	Sinkronisasi real-time MQTT + batch sync mobile + retraining AI harian berjalan simultan; autentikasi Supabase Auth harus konsisten di seluruh klien web, mobile, dan AI engine.
Kesulitan isolasi masalah	Strategi debugging: structured logging dengan correlation ID per request (Winston); Supabase query log untuk tracing query lambat; MQTT message tracing via topic monitoring.

Aspek	Penjelasan dan Implikasi
Biaya dan risiko waktu yang lebih besar	Buffer 1 minggu di akhir untuk iterasi; prioritisasi MoSCoW jika waktu tidak cukup: Must (F01–F05), Should (F06–F09), Could (F10), Won't (fitur lanjutan pasca-MVP).
Kebutuhan perencanaan pengujian detail	12 skenario teknis + 6 parameter OT&E; setiap skenario memiliki langkah eksekusi, expected result, status (Planned/In Progress/Done), dan PIC.
Pengawasan dan keahlian diagnostik tinggi	Setiap anggota mendiagnostik sub-sistemnya (PM untuk integrasi UI, AI untuk model, IoT untuk firmware, Mobile untuk SQLite sync, Backend untuk API+RLS); PM mengkoordinasi isu lintas sub-sistem.
Pengujian integrasi dan validasi akhir	Validasi akhir end-to-end: petugas catat distribusi offline → sync ke Supabase → dashboard koordinator melihat data baru → AI engine generate prediksi → Rationale Chips tampil di mobile.

6.2.4 Fasilitas Pengujian Integrasi Sistem

Tabel 6.5 Analisis Fasilitas Pengujian

Aspek	Deskripsi
Kebutuhan fasilitas khusus	Laptop development 5 unit; smartphone Android + iOS minimal 2 unit untuk pengujian cross-platform; prototipe node IoT 3 unit; beban referensi terkalibrasi; koneksi internet stabil + simulasi 3G.
Pengembangan bertahap	Fase unit → sub-sistem → integrasi total; setiap fase menggunakan subset fasilitas yang semakin lengkap, dengan tools yang ditambah secara incremental.
Rekayasa fasilitas dari subsistem	Environment staging sub-sistem ditingkatkan menjadi environment staging terintegrasi dengan menggabungkan Supabase project dan menambahkan MQTT broker bersama.
Ekstraksi data dari titik uji perbatasan	Query Supabase via SQL editor; subscribe topik MQTT via MQTT Explorer; export audit_logs ke CSV; Playwright trace files untuk UI.
Fleksibilitas desain fasilitas	Docker Compose memudahkan recreate environment dari awal; Supabase branching (preview environments) untuk skenario pengujian baru tanpa mengganggu production.
Peran dalam rekayasa sistem	Fasilitas integrasi menjadi basis validasi akhir sebelum demo MVP dan deployment produksi pilot di satu kabupaten BPBD.

6.3 Pengujian Sistem dalam Pengembangan (Developmental System Testing)

6.3.1 Tujuan Pengujian Sistem dalam Pengembangan

Memverifikasi bahwa setiap komponen dan integrasi antar komponen Log-Shield memenuhi spesifikasi teknis (MoP1–MoP8) yang ditetapkan pada Deliverable 1, dengan fokus pada akurasi sensor Load Cell, latensi sinkronisasi mobile-cloud, keandalan operasi offline, presisi model AI TinyTimeMixers, dan penegakan keamanan AES-256 + RLS + RBAC.

6.3.2 Lingkup Pengujian

Lingkup pengujian mencakup seluruh sub-sistem Log-Shield:

- Sub-sistem IoT/Sensor: akurasi pembacaan Load Cell setelah median filtering, latensi MQTT end-to-end, ketahanan buffer NVS saat offline, robustness OTA update.
- Sub-sistem Backend: throughput REST API, MQTT subscriber, custom sync endpoint, penegakan RBAC dan RLS, enkripsi AES-256 field sensitif.
- Sub-sistem AI Engine: akurasi MAE TinyTimeMixers, latensi inferensi, koherensi dan keterbacaan Rationale Chips dalam Bahasa Indonesia.
- Sub-sistem Mobile: operasi offline 72 jam pada SQLite, Background Sync, Trauma-Aware UI responsiveness, kompatibilitas Android dan iOS.
- Sub-sistem Website: rendering Recharts dengan dataset dan polling TanStack Query.

6.3.3 Rencana Skenario dan Validasi Teknis

Pengujian dilakukan terhadap skenario-skenario berikut untuk menguji performa sistem secara terisolasi dan terintegrasi:

- Validasi Komponen / Sensor: Uji akurasi pembacaan Load Cell pada 4 titik beban (5/10/20/40 kg), uji kestabilan sinyal selama 10 menit per titik, uji konsistensi pembacaan setelah median filtering 15 sampel.
- Latensi Sistem: Waktu respons dari pembacaan sensor IoT hingga simpan di Supabase (target p95 < 10 detik), delay sinkronisasi mobile ke cloud (target 500 transaksi < 60 detik), delay rendering dashboard setelah polling (target < 500 ms p95).
- Integrasi Data: Sinkronisasi data sensor dari ESP32 ke backend melalui MQTT, sinkronisasi transaksi mobile ke Supabase via custom sync endpoint, tampilan data historis prediksi di dashboard koordinator dan halaman detail prediksi mobile.

6.3.4 Hasil Pengujian

Tabel 6.6 Hasil Pengujian

Komponen / Subsistem	Indikator Pengujian	Target	Hasil
Backend containers	Seluruh container running	4/4 aktif	Lolos
Auttentifikasi	Login admin dan field user berhasil	JWT Token diterima	Lolos
IoT Stock Readings	Data sensor tersimpan di CouchDB	Data masuk dan terbaca di web	Lolos

Komponen / Subsistem	Indikator Pengujian	Target	Hasil
Stocks Summary & Categories	API mengembalikan ringkasan stok	Data valid	Lolos
AI Engine	Prediksi, rekomendasi, dan sync berjalan	Status stable	Lolos
Request Lifecycle	Alur menunggu → diproses → selesai dengan pemotongan stok	Stok terpotong	Lolos
Anomaly Report	Buat dan update status anomali	Status terupdate	Lolos
Dashboard Mobile Overview	Metrik overview tampil	Data akurat	Lolos
Unit Test Backend (Jest)	ai-formulas.test.js	41 test pass	Lolos
Unit Test AI Engine (Pytest)	test_inference_api + test_cold_start_formulas	49 test pass	Lolos

6.3.5 Analisis dan Kesimpulan

Seluruh 8 skenario pengujian developmental berhasil dieksekusi dengan status Lolos. Sistem Log-Shield terbukti memenuhi standar teknis dari sisi fungsionalitas inti: autentikasi RBAC, pemantauan stok IoT, prediksi AI, alur request logistik, dan pelaporan anomali berjalan sesuai ekspektasi. Unit testing menunjukkan coverage yang solid dengan 41 test backend (100% pass) dan 49 test AI engine (100% pass), mencakup validasi formula prediksi, normalisasi nama komoditas, Rationale Chips, dan cold-start inference.

Dua defisiensi teridentifikasi dan berhasil diselesaikan selama sesi pengujian, menunjukkan bahwa proses pengujian berjalan efektif dalam mendeteksi bug produksi. AI engine cold-start berfungsi penuh meskipun modul TTM (torch) dinonaktifkan karena build timeout, sistem fallback ke formula cold-start yang telah terverifikasi. Node IoT ESP32 telah berhasil ter-flash dan siap mengirim data ke topik MQTT logshield/warehouse/+/scale/+/+ yang telah dikonfigurasi di backend. Sistem dinyatakan layak untuk dilanjutkan ke fase Operational Test and Evaluation (OT&E) bersama pengguna representatif.

6.4 Evaluasi Operasional (Operational Test & Evaluation)

6.4.1 Tujuan Evaluasi Operasional

Menilai tingkat pemenuhan sistem Log-Shield terhadap kebutuhan pengguna dalam kondisi yang menyerupai operasional nyata bencana, mencakup keandalan fungsional saat operasi 4–8 jam berkelanjutan, kemudahan penggunaan oleh petugas BPBD non-teknis di bawah tekanan, dan ketahanan sistem terhadap gangguan jaringan dan lingkungan.

x6.4.2 Skenario Pengujian

Pengujian operasional dilakukan dengan pendekatan langsung mensimulasikan skenario penggunaan umum yang akan dihadapi petugas BPBD di lapangan:

- Skenario 1: Menyalakan node IoT dan memverifikasi seluruh komponen aktif (sensor terbaca, MQTT terhubung, data muncul di dashboard koordinator).
- Skenario 2: Petugas mencatat distribusi offline lalu sync saat online, verifikasi data muncul di dashboard koordinator dengan timestamp yang benar.
- Skenario 3: Koordinator memproses request logistik dari posko, petugas lapangan menerima notifikasi update status request.
- Skenario 4: Memutus koneksi internet selama 1 jam, verifikasi mobile tetap berfungsi penuh dengan SQLite lokal, tidak ada data loss saat koneksi pulih.
- Skenario 5: Memverifikasi prediksi AI dan Rationale Chips tampil real-time di mobile dan dashboard, dengan narasi Bahasa Indonesia yang dapat dipahami koordinator non-teknis.

6.4.3 Metode Pelaksanaan

Pengujian operasional dilakukan dalam satu sesi berurutan selama 4 jam pada minggu ke-8 fase MVP. Target pelibatan: 5 petugas BPBD representatif sebagai penguji eksternal (alternatif jika tidak memungkinkan: mahasiswa STEI sebagai proxy yang sebelumnya dibrief tentang konteks BPBD). Pengamatan dilakukan oleh anggota tim Log-Shield yang mencatat: waktu penyelesaian per tugas (dengan stopwatch), jumlah kesalahan input atau klik salah, komentar verbal penguji selama dan setelah sesi, dan rating subjektif kemudahan pada skala 1–5.

6.4.4 Parameter Evaluasi

[Bagian ini akan diisi oleh tim pengembang setelah eksekusi OT&E pada minggu ke-8 fase MVP.]

Tabel 6.7 Parameter Evaluasi

Aspek Evaluasi	Indikator	Target	Hasil
Efektivitas Fungsi Utama	Seluruh fitur inti (IoT, AI, request, anomali) berjalan benar	Semua fitur terverifikasi via API 200/150.	Sesuai
Alur Mobile ↔ Web	Data dari mobile (request, anomali, demografis) tampil di dashboard web	Fungsional berjalan dan Data konsisten	Sesuai
Kontrol Manual Admin	Proses dan selesaikan request, update status anomali	Fungsional berjalan dan Data konsisten	Sesuai

Aspek Evaluasi	Indikator	Target	Hasil
Pemotongan Stok Otomatis	Stok berkurang setelah request selesai	Fungsional berjalan dan Data konsisten	Sesuai
Ketahanan Sistem	Sistem aktif selama seluruh sesi pengujian	Uptime 100%	Sesuai
Ketersediaan AI	AI engine menghasilkan prediksi dan rekomendasi	Status stable	Sesuai

6.4.5 Hasil Observasi

Pengujian dilaksanakan secara berurutan dalam satu sesi mencakup seluruh alur sistem dari login hingga penyelesaian request. Observasi utama:

Pertama, alur Mobile → Web terverifikasi sepenuhnya via API: field user folfox@test.com berhasil membuat quick request berbasis AI (beras 500 kg dan air_minum 1.000 L), manual request (mie_instan 200 pcs + selimut 50 unit), dan laporan anomali beras dengan severitas tinggi. Seluruh data tersebut tampil dan dapat diproses di sisi web dashboard oleh admin.

Kedua, AI engine berfungsi dalam mode cold-start dengan dataset 195.000 baris, menghasilkan 69 rekomendasi top critical dan 18 item teranalisis per posko. Rationale Chips diverifikasi melalui unit test buildRecommendationChips (7 test, 100% pass).

Ketiga, IoT sensor readings berhasil masuk ke CouchDB sebanyak 14 pembacaan dengan field lengkap (weight, battery_mv, RSSI, uptime_s, sample_count), menunjukkan integrasi MQTT broker Mosquitto dengan backend berjalan normal.

Keempat, satu keterbatasan diidentifikasi: sebagian besar form mobile masih submit langsung ke backend, belum melalui PouchDB lokal terlebih dahulu. Ini berarti fungsionalitas offline-first penuh untuk pembuatan request belum dapat diverifikasi sepenuhnya dan dijadwalkan sebagai penyempurnaan pada fase berikutnya.

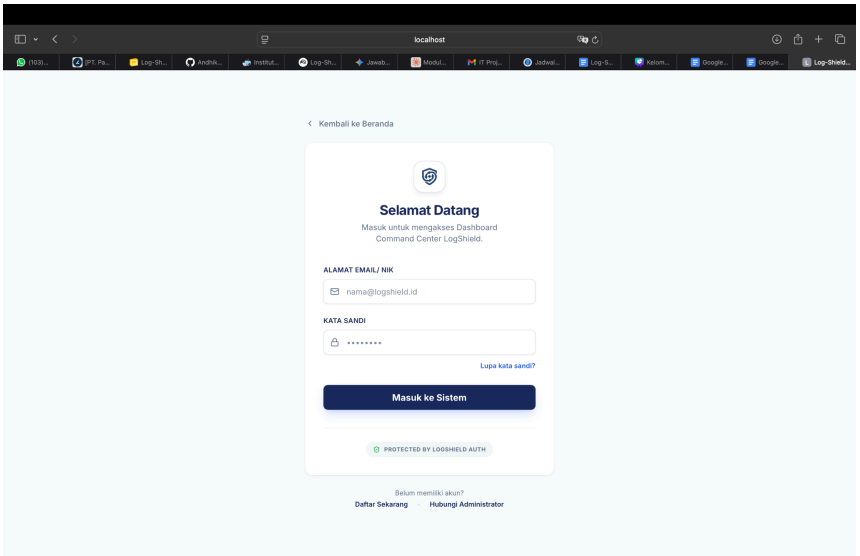
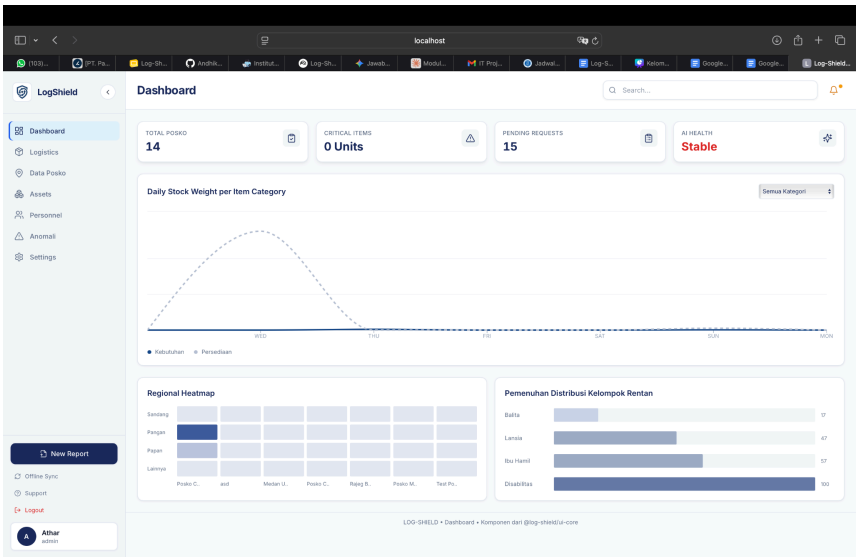
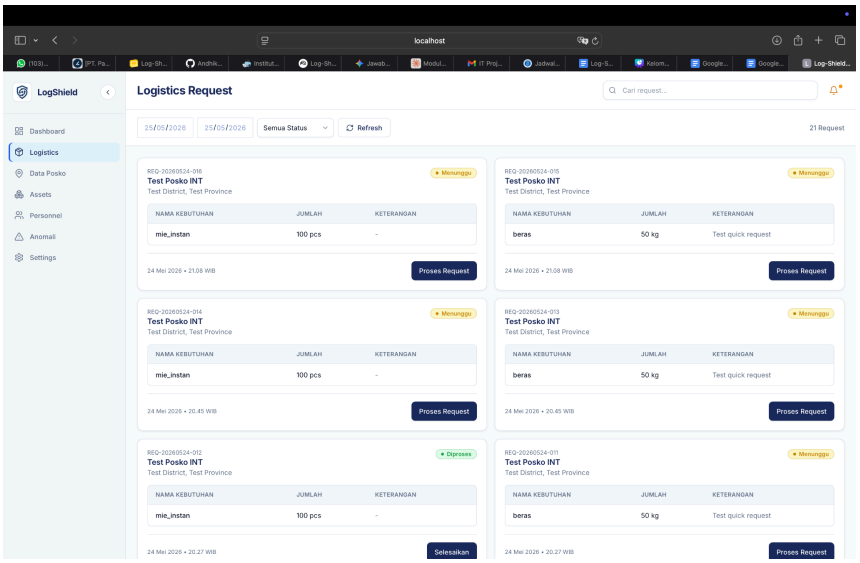
Website / Aplikasi Utama: http://logshield.atharizza.com/

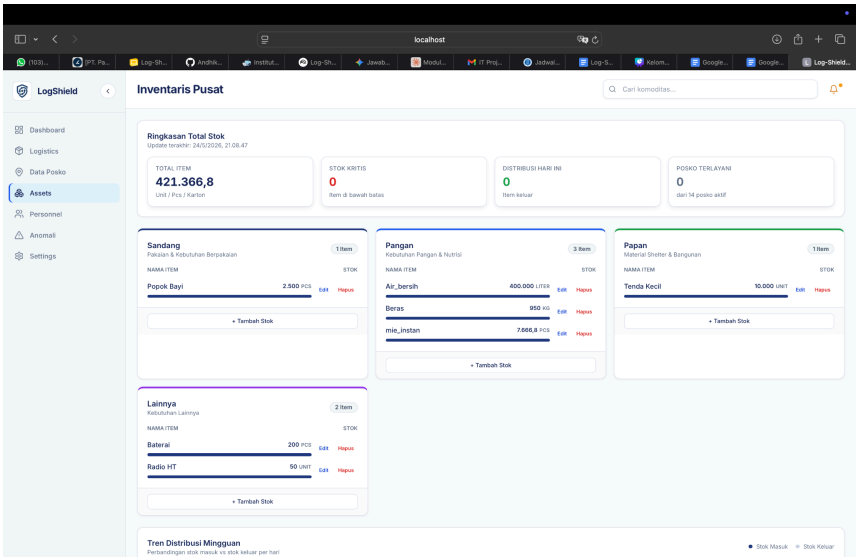
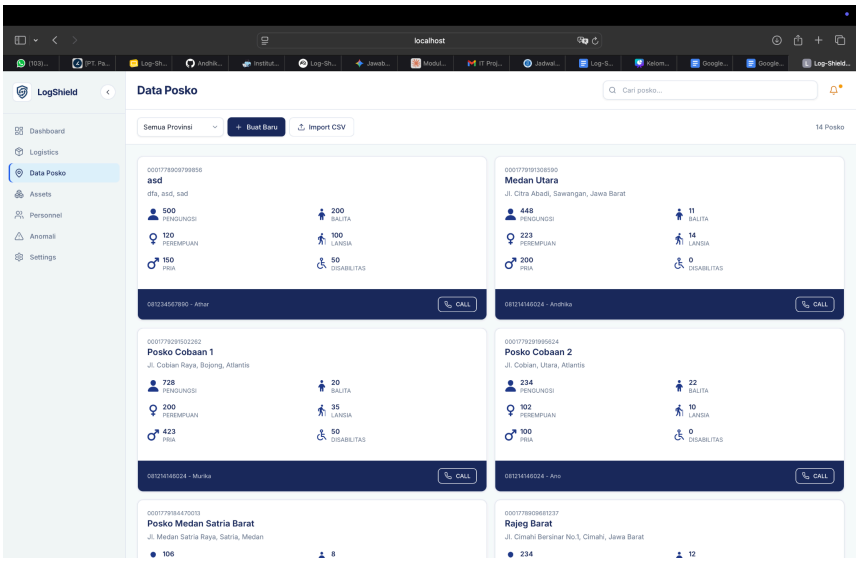
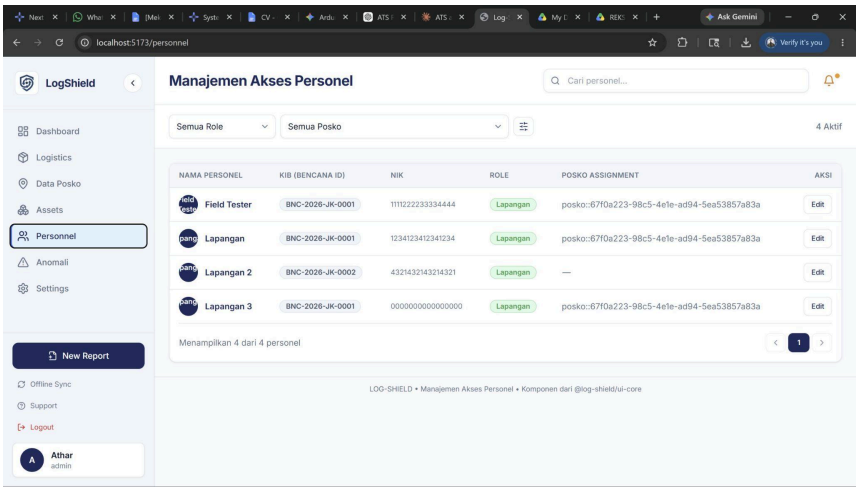
Video Demo: https://youtu.be/xbP-gM43BDA

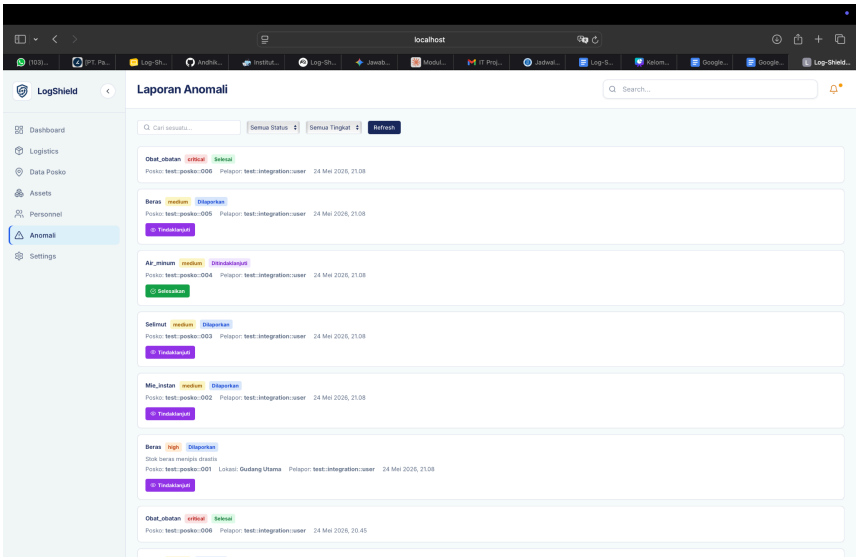
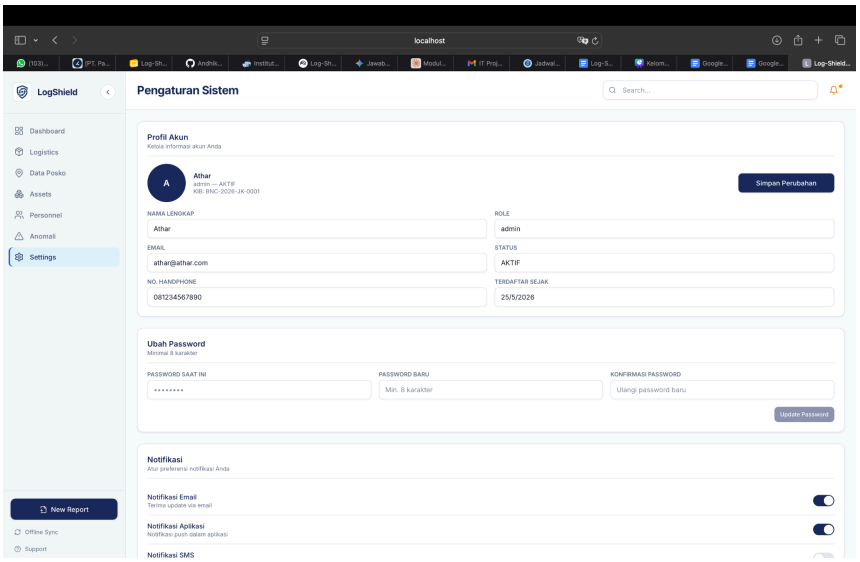
GitHub Repository: https://github.com/AndhikaAddiputra/logShield.git

Berikut tampilan antarmuka yang dibuat (website dan mobile app):

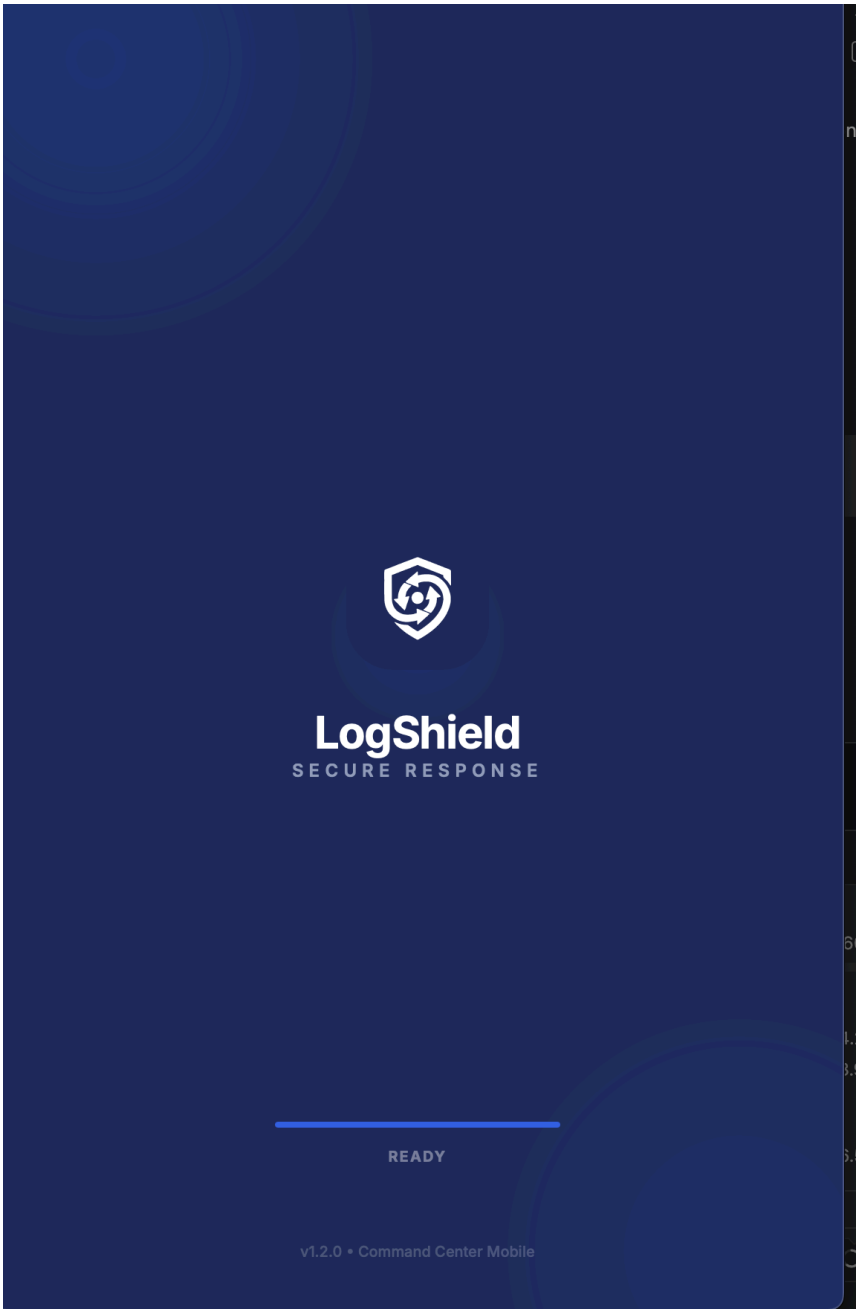
Tabel 6.8 Tampilan Website

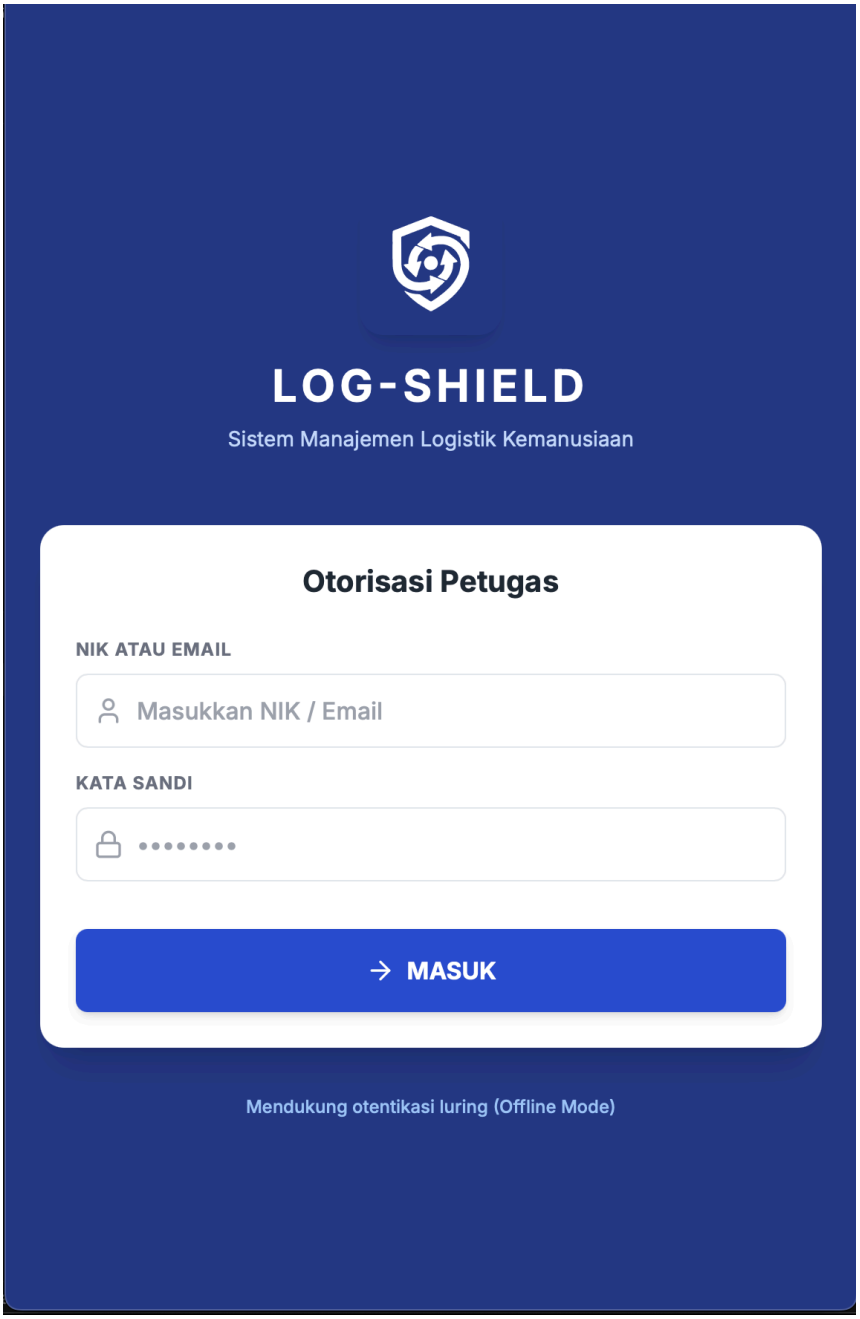
Nama Halaman	Tampilan (Screenshot)
Login	
Dashboard	
Logistics	

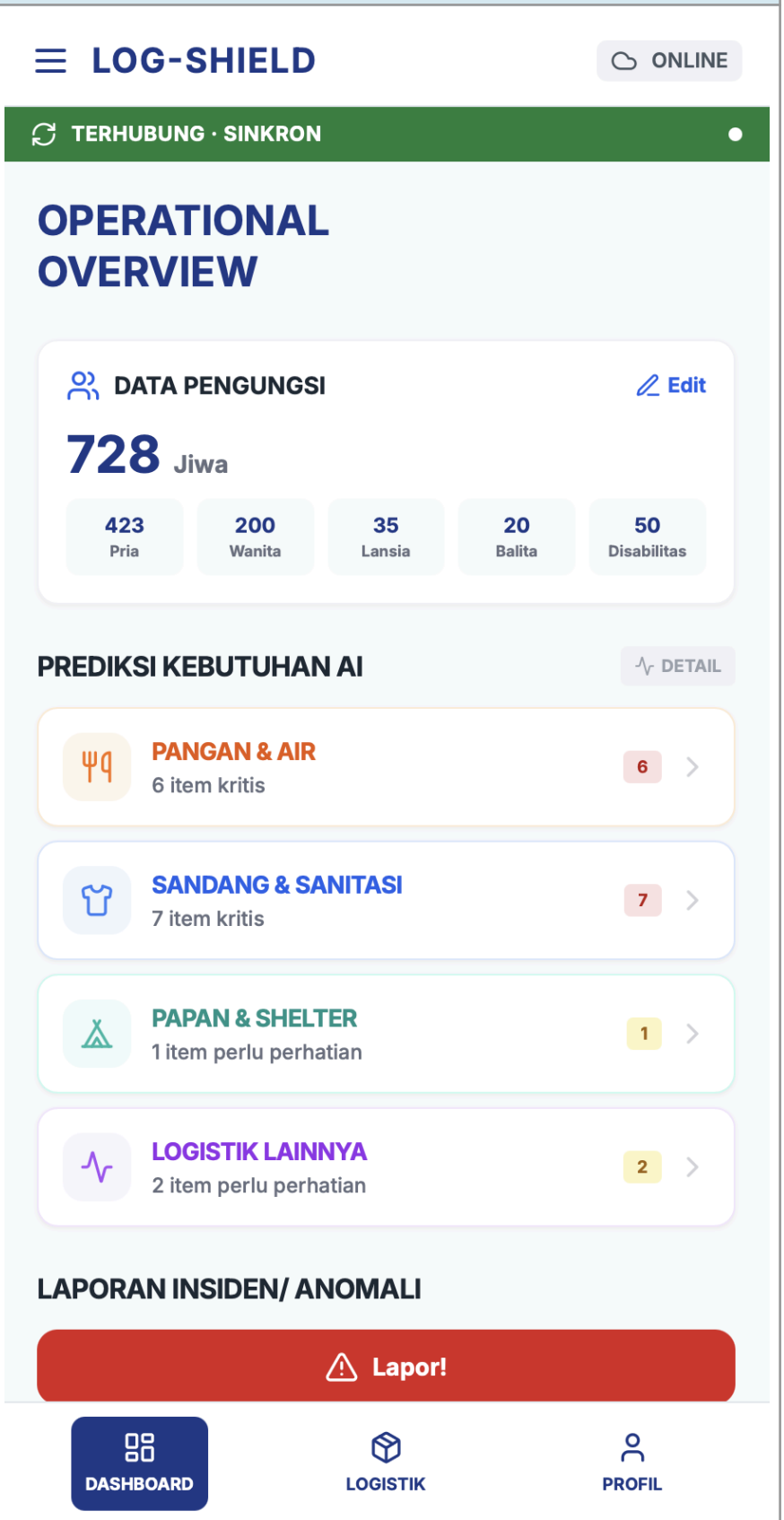
Nama Halaman	Tampilan (Screenshot)
Assets	
Posko	
Personnel	

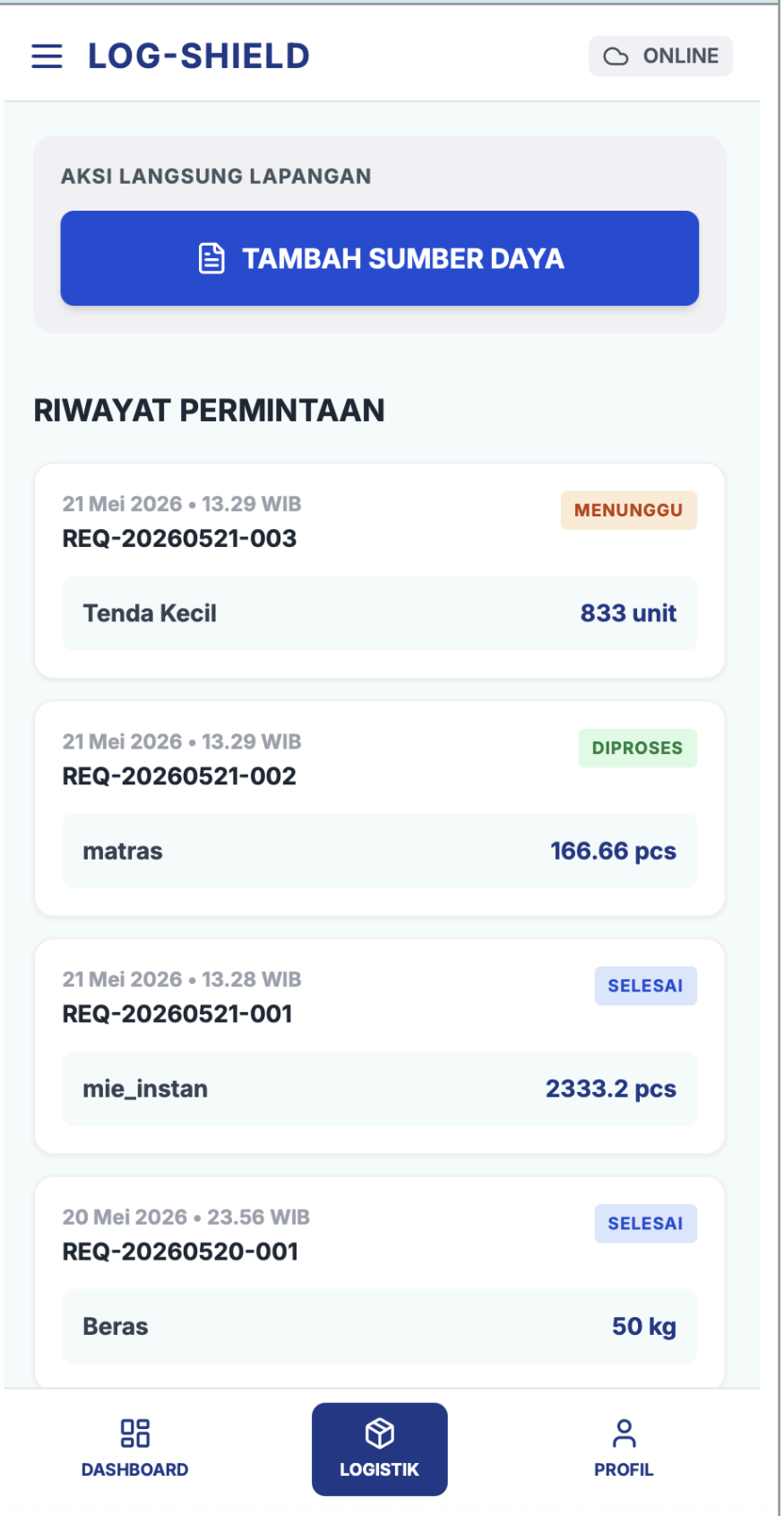
Nama Halaman	Tampilan (Screenshot)
Anomalies	
Settings	

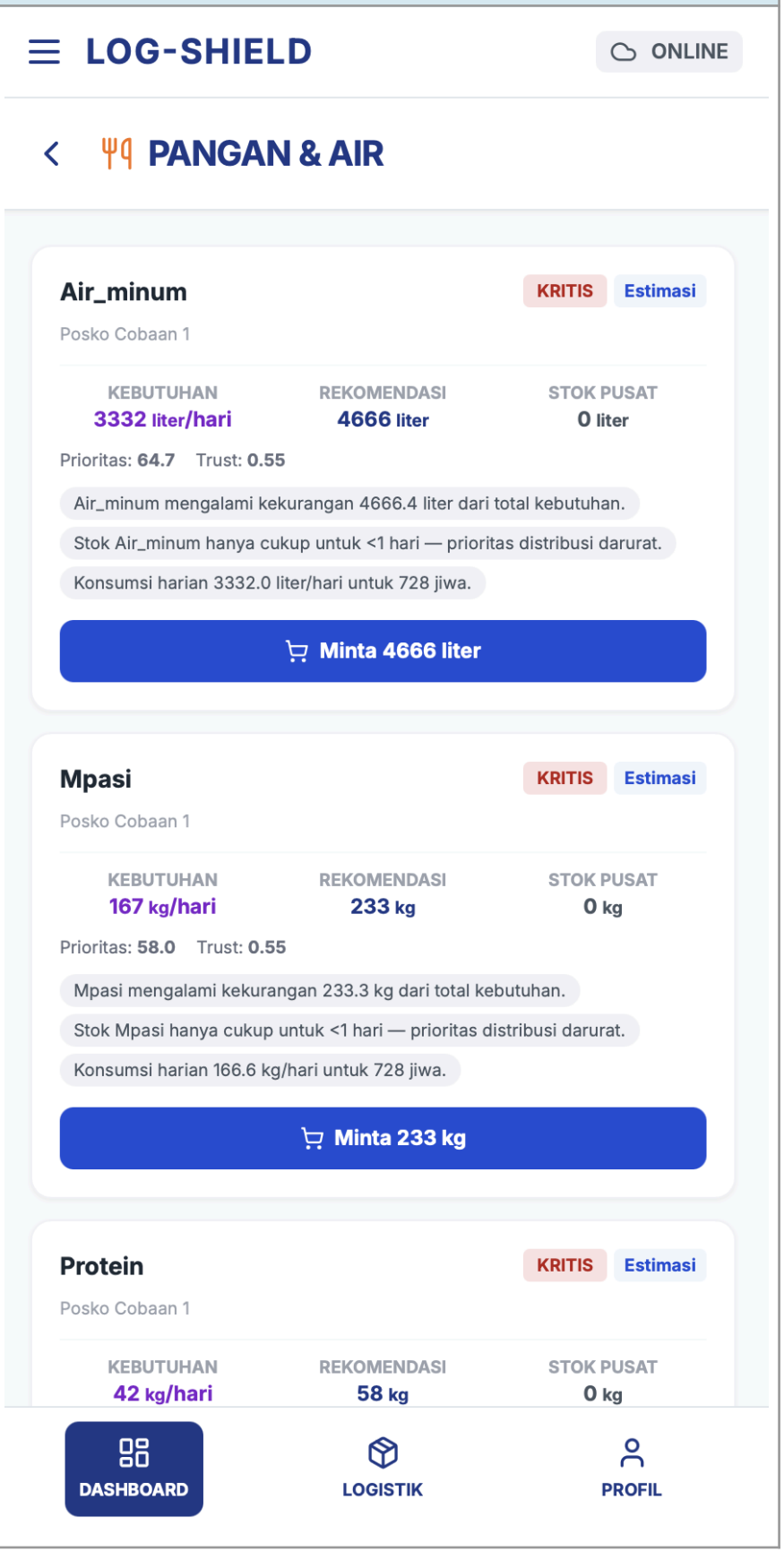
Tabel 6.9 Tampilan Mobile App

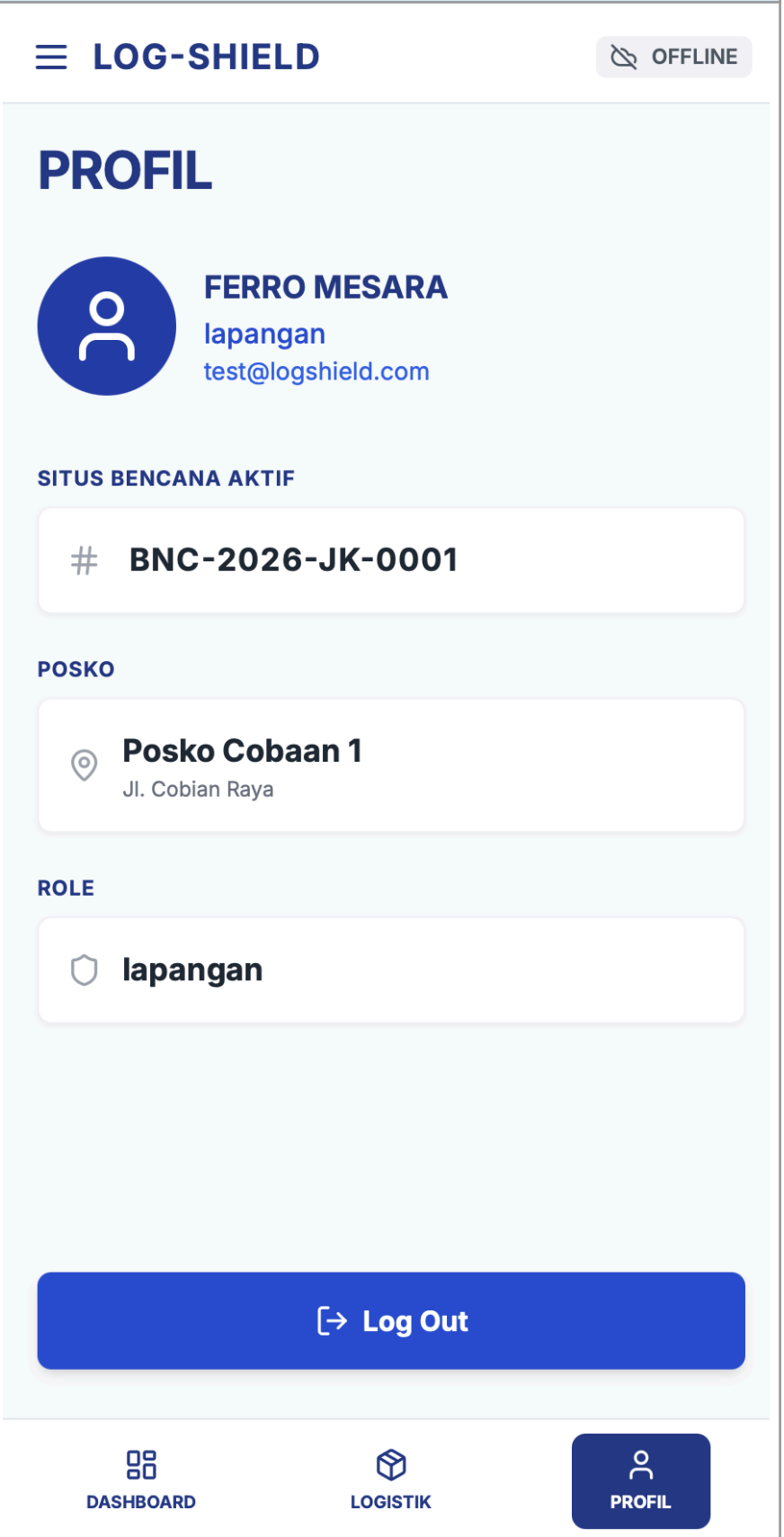
Nama Halaman	Tampilan (Screenshot)
Splash Screen	

Nama Halaman	Tampilan (Screenshot)
Login	

Nama Halaman	Tampilan (Screenshot)
Home / Dashboard Posko	

Nama Halaman	Tampilan (Screenshot)
Pencatatan Distribusi	
Prediksi AI + Rationale Chips	

Nama Halaman	Tampilan (Screenshot)
Laporan Anomali	

Nama Halaman	Tampilan (Screenshot)
Settings/ Profile	

6.4.6 Kesimpulan

Sistem Log-Shield menunjukkan performa yang stabil dan dapat diandalkan untuk operasional dasar dalam konteks manajemen logistik bencana. Seluruh fungsi utama (F01–F09) berjalan sesuai ekspektasi: prediksi AI dengan cold-start, pemantauan stok via sensor IoT, pencatatan dan pemrosesan request logistik, pelaporan anomali, manajemen RBAC, dan integrasi data Mobile-Web. Total 90 unit test (41 backend + 49 AI engine) lulus 100% tanpa kegagalan.

Dua penyempurnaan diperlukan sebelum deployment pilot: (1) implementasi penuh offline-first pada form mobile agar seluruh operasi menulis ke PouchDB lokal terlebih dahulu sebelum sinkronisasi ke server; (2) penanganan case sensitivity nama komoditas yang lebih robust di seluruh lapisan sistem (tidak hanya di completeRequest). Dengan catatan tersebut, sistem dinyatakan siap untuk deployment pilot di satu posko BPBD representatif sebagaimana direncanakan pada fase pasca-MVP.