

RANCANG BANGUN APLIKASI MANAJEMEN GANGGUAN INFRASTRUKTUR TEKNOLOGI INFORMASI KANTOR BERBASIS WEB DENGAN NOTIFIKASI REAL-TIME

¹Atang Supena

²Anindito Kusumo Birowo

Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta,
atang@itbu.ac.id , anindito@itbu.ac.id

Abstrak

Transformasi digital menuntut instansi beralih dari paradigma pencatatan konvensional menuju sistem terintegrasi guna menjamin keandalan infrastruktur Teknologi Informasi (TI). Penelitian ini dilatarbelakangi oleh inefisiensi tata kelola pelaporan gangguan infrastruktur yang berjalan secara konvensional melalui lisan atau pesan WhatsApp personal. Masalah tersebut menyebabkan tidak adanya perekaman data terpusat, sulitnya mengukur KPI teknis, lambatnya respons, asimetri informasi bagi pelapor, serta hilangnya Tacit Knowledge perbaikan. Penelitian ini bertujuan untuk merancang dan membangun aplikasi helpdesk terpadu bernama INFRADESK guna mendigitalisasi pelaporan, mereduksi latensi respons, memberikan transparansi pelacakan tiket, serta mengamankan pengetahuan teknis. Metodologi yang digunakan adalah Penelitian dan Pengembangan (Research and Development / R&D) dengan model siklus hidup klasik Waterfall menurut Pressman, yang mencakup tahap requirement, design, Implementation, Verification, dan Maintenance. Aplikasi dikembangkan berbasis web menggunakan framework PHP Laravel, basis data relasional MySQL, protokol WebSockets (Pusher/Laravel Reverb) untuk pembaruan dasbor instan, serta integrasi WhatsApp Gateway API sebagai peringatan darurat ke ponsel teknis. Modul Knowledge Management System (KMS) dirancang secara sekuensial dengan mewajibkan teknis mendokumentasikan spesifikasi aset dan catatan perbaikan sebelum tiket diselesaikan. Hasil pengujian fungsional menggunakan metode Black Box Testing dengan teknik Equivalence Partitioning pada seluruh formulir utama membuktikan bahwa sistem berjalan sukses dan berstatus valid tanpa adanya kegagalan program (bug).

Kata Kunci : INFRADESK, Waterfall, WebSockets, WhatsApp Gateway, Knowledge Management System.

1. PENDAHULUAN

Transformasi digital menuntut setiap instansi beralih dari paradigma pencatatan konvensional menuju sistem terintegrasi guna menjamin keandalan infrastruktur teknologi informasi (TI). Stabilitas jaringan internet, perangkat keras komputer, dan kelistrikan menjadi pilar sentral penggerak produktivitas instansi. Namun, tata kelola pelaporan dan penanganan gangguan infrastruktur pada objek penelitian ini masih bersifat konvensional, di mana karyawan melaporkan kendala teknis secara lisan atau melalui aplikasi pesan singkat (WhatsApp) secara acak tanpa standar format yang baku. Pendekatan ad-hoc ini menyebabkan data historis gangguan tidak terdokumentasi dalam satu basis data terpusat, menyulitkan teknis dalam memprioritaskan urgensi penanganan, dan memicu penundaan perbaikan yang

mengganggu kesinambungan operasional kerja.

Kondisi tersebut bermuara pada empat disfungsi operasional. Pertama, pencatatan manual berbasis WhatsApp dan lisan menyebabkan ketiadaan repositori sentral, sehingga manajemen tidak memiliki landasan valid untuk mengevaluasi Indikator Kinerja Utama (Key Performance Indicator / KPI) para teknis. Kedua, ketiadaan mekanisme notifikasi secara real-time membuat teknis lambat menyadari masuknya insiden kritis, yang memperpanjang waktu henti (downtime). Ketiga, pelapor mengalami asimetri informasi karena tidak dapat melacak progres perbaikan fasilitas secara mandiri. Keempat, ketiadaan sistem manajemen pengetahuan (Knowledge Management System / KMS). Knowledge Management System (KMS) adalah sistem berbasis teknologi informasi yang

dikembangkan untuk mendukung proses penciptaan, penangkapan, penyimpanan, dan penyebaran pengetahuan di dalam suatu organisasi. Teori penciptaan pengetahuan yang paling fundamental dikemukakan oleh Ikujiro Nonaka dan Hirotaka Takeuchi (Nonaka, I. & Takeuchi, H, 1995). Membuat rekam jejak penyelesaian masalah (troubleshooting notes) hilang, memaksa teknisi melakukan diagnosis ulang dari awal saat menghadapi kerusakan serupa di masa depan.

Untuk mengeliminasi permasalahan tersebut, dibangun Aplikasi Manajemen Gangguan Infrastruktur Kantor Berbasis Web dengan Notifikasi Real-time. Sistem ini dikembangkan menggunakan framework Laravel dan MySQL, serta menerapkan *Role-based access control (RBAC)* guna memisahkan hak akses pelapor, teknisi, dan administrator/ manajer TI. Aplikasi ini mengintegrasikan teknologi *WebSockets* adalah protokol komunikasi dua arah (full-duplex) yang persisten, berjalan melalui koneksi Transmission Control Protocol (TCP) tunggal yang established antara browser dan server (Pusher Development Team, Pusher. 2026) dan WhatsApp Gateway untuk memancarkan notifikasi ganda secara seketika, sekaligus menanamkan modul KMS untuk mendokumentasikan rekam jejak solusi. Melalui sinergi ini, diharapkan latensi penanganan dapat direduksi, pengawasan KPI menjadi objektif, dan repositori pengetahuan institusi dapat terbangun secara tangguh.

Berdasarkan permasalahan diatas, penelitian ini memiliki tujuan dan manfaat sebagai berikut:

1. Merancang dan membangun aplikasi manajemen gangguan infrastruktur kantor berbasis web menggunakan basis data relasional MySQL guna mendigitalisasi serta mensentralisasi kan proses dokumentasi pelaporan agar lebih terstruktur.

2. Mengimplementasikan sistem notifikasi dual-channel yang memadukan teknologi *WebSockets* untuk pembaruan dashboard web secara real-time dan WhatsApp Gateway/API sebagai peringatan instan ke perangkat seluler, guna meminimalkan latensi response time.
3. Mendeskripsikan dan membangun mekanisme antarmuka pelacakan yang komprehensif, memungkinkan entitas karyawan untuk melakukan tracking status perkembangan tiket secara transparan dari fase Pending hingga Done.
4. Menerapkan arsitektur *Knowledge Management System (KMS)* pragmatis melalui modul otomatisasi system log dan rekam jejak troubleshooting notes untuk mengekstraksi pengetahuan taksit (Tacit Knowledge) teknisi menjadi aset repositori perusahaan.

2. METODOLOGI PENELITIAN

2.1 Metode Penelitian

Sebuah penelitian dapat dilihat bentuk penelitiannya dari jenis penelitian yang dilakukan. Jenis penelitian yang digunakan dalam penyusunan skripsi ini adalah Metode Penelitian dan Pengembangan (Research and Development / R&D). Metode Penelitian dan Pengembangan (R&D) adalah strategi atau metode penelitian yang cukup ampuh untuk memperbaiki produk/desain, yaitu rangkaian proses atau langkah-langkah dalam rangka mengembangkan suatu produk/desain baru atau menyempurnakan produk/desain yang telah ada agar dapat dipertanggungjawabkan.

Pemilihan jenis penelitian ini didasarkan pada luaran utama dari skripsi ini, yaitu menghasilkan sebuah produk berupa "Aplikasi Manajemen Gangguan Infrastruktur Kantor Berbasis Web dengan Notifikasi Real-time". Produk ini dikembangkan secara komprehensif menggunakan kerangka

kerja Laravel dan basis data MySQL, serta mengintegrasikan Knowledge Management System (KMS) pragmatis dan teknologi notifikasi *dual-channel* (*WhatsApp Gateway* dan *WebSockets*) untuk menyempurnakan mekanisme pelaporan yang sebelumnya berjalan secara konvensional.

2.2 Kerangka Berpikir

Kerangka berpikir merupakan representasi pola pemecahan masalah secara utuh dari hulu ke hilir. Mengacu pada pengintegrasian antara tahapan rekayasa perangkat lunak dengan metode pembahasan hasil analisis, kerangka berpikir dalam penelitian ini digambarkan pada alur kerja (flow) berikut:

Berdasarkan pemodelan kerangka berpikir di atas, alur penelitian dieksekusi secara terstruktur melalui lima tahapan utama:

1. Identifikasi Masalah: Tahap pengidentifikasian inefisiensi pelaporan berbasis WhatsApp dan lisan, tingginya latensi waktu tanggap, serta ketiadaan repositori Knowledge Management System (KMS).
2. Pengumpulan Data dan Perumusan Solusi: Tahap penarikan data empiris melalui observasi lapangan, wawancara tiga entitas (manajerial, eksekutor, pelapor), studi dokumen, hingga merumuskan solusi berupa aplikasi berbasis web.
3. Realisasi dan Pengembangan Sistem: Tahap konstruksi perangkat lunak menggunakan arsitektur MVC pada framework Laravel dan basis data MySQL, menerapkan pembatasan hak akses (RBAC), serta mengintegrasikan WebSockets dan API WhatsApp Gateway.
4. Penyajian dan Pengujian Sistem: Tahap menjabarkan antarmuka (User Interface) serta fungsionalitas modul utama (Triage Admin, Form KMS, dan Pelacakan Tiket),

dilanjutkan dengan verifikasi logika internal menggunakan metode Black-Box Testing.

5. Evaluasi Pemecahan Masalah: Tahap akhir untuk menginterpretasikan dan membandingkan kesesuaian produk dengan tujuan penelitian; difokuskan pada validasi sentralisasi data (system log), efektivitas pemangkasan latensi respons, dan pengamanan aset pengetahuan (Tacit Knowledge) ke dalam KMS.

2.3 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan penulis untuk mendapatkan data dan informasi sehingga dapat mendukung penyempurnaan hasil penelitian ini adalah sebagai berikut:

1. Observasi (Observation): Observasi merupakan teknik pengumpulan data dengan cara melakukan pengamatan secara langsung di lokasi penelitian, yaitu di PT Lumbung Riang Communication yang berlokasi di Pondok Kelapa Dua, Jl. Camar Blok MD No. 1, RT.5/RW.05, Pasir Gunung Selatan, Cimaggis, Kota Depok, Jawa Barat 16451, sebuah perusahaan yang bergerak di bidang jasa telekomunikasi. Pada tahap ini, penulis mengamati secara menyeluruh aktivitas operasional karyawan terkait tata kelola penanganan gangguan infrastruktur teknologi informasi (IT), mulai dari munculnya kendala teknis hingga perbaikan tersebut selesai dieksekusi. Berdasarkan pengamatan langsung di lapangan, diperoleh temuan riil bahwa mekanisme pelaporan masih bersifat konvensional; di mana setiap masalah dilaporkan secara manual melalui aplikasi pesan singkat personal (WhatsApp) atau instruksi lisan, tanpa didukung oleh

sistem pemantauan (monitoring) dan pelacakan tiket yang terpusat

2. Wawancara (InterView): Pengambilan data melalui wawancara atau lisan secara langsung dari sumber data, baik secara tatap muka maupun melalui alat komunikasi. Wawancara dilakukan dengan pihak manajemen operasional, teknisi IT, dan pengguna fasilitas (karyawan) untuk mengidentifikasi kebutuhan spesifik sistem manajemen pengetahuan (KMS) dan alur perbaikan Service Level Agreement (SLA) Pilar utama penjaminan mutu dalam manajemen insiden adalah penerapan *Service Level Agreement (SLA)* (Whitten, J. L. & Bentley, L. D.,2007).

Berikut adalah dokumentasi dan penjabaran detail hasil wawancara dari ketiga entitas pengguna:

1.) Wawancara Entitas Manajerial (Kepala IT / Team Leader)

- Narasumber: Teguh Hardiansyah (Ex Team Leader Project ESNB Internasional)
- Lokasi & Waktu: Ruang Meeting PT Lumbung Riang Communication (Jumat, 10 April 2026, 16.30 WIB)
- Bukti Wawancara :



Gambar 1. Wawancara Dengan Narasumber Teguh Hardiansyah Sebagai Software Engineer
Sumber: Penelitian Mandiri 2026

Wawancara pada entitas ini difokuskan pada mekanisme pengawasan kinerja teknisi. Dari hasil diskusi, diperoleh informasi bahwa kendala utama manajemen adalah ketiadaan rekam jejak historis gangguan yang terpusat, sehingga menyulitkan perhitungan Indikator Kinerja Utama (Key Performance Indicator / KPI) teknisi. Selain itu, narasumber menekankan perlunya regulasi pembatasan hak akses (Role-based access control) yang tegas pada sistem baru.

2.) Wawancara Entitas Eksekutor (Teknisi / IT Support)

- Narasumber 1: Togar Pandaoni (Member Team Project Instalasi Antena MW, PT Lumbung Riang Communication) - Halaman Belakang Kantor, 10 April 2026, 19.00 WIB.
- Narasumber 2: Kharisma Muhammad Reihan (Staff & Team Engineer Project Metro Sisindokom, PT Lumbung Riang Communication) - Ruang Admin, 10 April 2026, 20.03 WIB.
- Narasumber 3: Daffa Ramadhan (IT Support, SMK Taruna Bangsa Bekasi) - Online Google Meet, 5 April 2026, 12.22 WIB.
- Bukti wawancara:



Gambar 2. Wawancara dengan Narasumber Kharisma Sebagai Teknisi Lapangan
Sumber: Penelitian Mandiri 2026

Penggalan data pada kelompok eksekutor menyoroiti kendala teknis

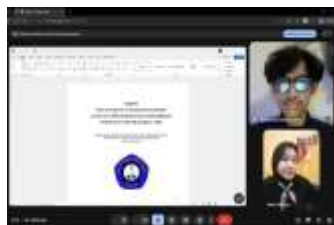
di lapangan. Para teknisi menyatakan bahwa pelaporan gangguan melalui pesan singkat (WhatsApp) pribadi sering kali tertumpuk oleh obrolan lain, memicu keterlambatan penanganan. Mereka membutuhkan adanya peringatan notifikasi seketika serta kemudahan dalam mendokumentasikan catatan perbaikan (troubleshooting notes).

3.) Wawancara Entitas Pelapor (Karyawan / Pengguna Fasilitas)

- Narasumber 1: Kharisma Muhammad Reihan (Staff & Team Engineer Project Metro Sisindokom, PT Lumbung Riang Communication)
- Narasumber 2: Ulfah Wadhhah Fakhirah (Sales and Operasional Administrator, PT Sunway Asia Industry) - Online Google Meet, 7 April 2026, 21.25 WIB.
- Bukti wawancara



Gambar 3. Wawancara dengan narasumber Kharisma
Sumber: Penelitian Mandiri 2026



Gambar 4 Wawancara dengan narasumber Ulfah Wadhhah F Sebagai Administrator Operasional

Sumber: Penelitian Mandiri 2026

Diskusi dengan perwakilan pelapor menunjukkan adanya asimetri informasi, di mana karyawan tidak dapat mengetahui sejauh mana progres perbaikan fasilitas yang mereka laporkan. Kebutuhan mendasar dari kelompok ini adalah ketersediaan fitur pelacakan tiket (ticket tracking) mandiri.

3. Dokumen: Pengambilan data dalam bentuk dokumen tertulis atau elektronik dari lembaga/institusi atau pihak pemilik data. Berdasarkan hasil pengamatan di lokasi instansi penelitian, ditemukan fakta bahwa belum ada dokumen historis pelaporan atau Knowledge Management System (KMS) yang baku. Oleh karena itu, pengumpulan data dokumen pada penelitian ini mengambil sampel dari arsip-arsip informal yang digunakan secara ad-hoc oleh tim operasional. Dokumen tersebut meliputi:

- Rekapitan catatan pribadi teknisi (format Microsoft Excel) yang dibuat secara mandiri untuk mencatat riwayat perbaikan
- Tangkapan layar (screenshot) instruksi pekerjaan dan pelaporan gangguan melalui pesan singkat (WhatsApp)
- Dokumentasi foto aktivitas perbaikan (troubleshooting) yang dilakukan oleh teknisi di lapangan

Pengumpulan dokumen internal yang tidak tersentralisasi ini menjadi bukti empiris yang memperkuat urgensi perlunya dibangun aplikasi pelaporan terintegrasi. Berikut adalah salah satu sampel dokumen berupa catatan riwayat perbaikan pribadi (manual) dari tim teknisi :

No	Tanggal Laporan	Nama Pengisi	Jumlah Masalah	Diserang Perangkat	Nama Timbul	Media Laporan
1	20-Jan-2026	Prima Mulya	10	Perangkat jaringan	Prima Mulya	Printed
2	20-Jan-2026	Prima Mulya	10	Perangkat jaringan	Prima Mulya	Printed
3	20-Jan-2026	Prima Mulya	10	Perangkat jaringan	Prima Mulya	Printed
4	20-Jan-2026	Prima Mulya	10	Perangkat jaringan	Prima Mulya	Printed
5	20-Jan-2026	Prima Mulya	10	Perangkat jaringan	Prima Mulya	Printed

Gambar 5 Catatan pribadi tim teknis
Sumber: Penelitian Mandiri 2026



Gambar 6 Dokumentasi intruksi dari pesan singkat

Sumber: Penelitian Mandiri 2026



Gambar 7 Dokumentasi pengerjaan Troubleshooting

Sumber: Penelitian Mandiri 2026

Data yang diperoleh terdiri dari 2 (dua) macam, yaitu:

1. Data Primer: Data yang diperoleh dari survei/pengamatan/pengawasan, wawancara langsung dengan pihak terkait di lokasi penelitian.
2. Data Sekunder: Data yang diperoleh dari pihak lain seperti literatur akademik, referensi pustaka mengenai kerangka kerja pengembangan web, dan dokumentasi pengelolaan basis data.

2.3 Metode Analisis Data

Analisis data adalah kegiatan mengolah data, dari data yang sudah diperoleh, melakukan kajian dengan teori dan data-data yang diperoleh. Analisis Data

juga dilakukan dengan mengorganisasikan data, menjabarkan kedalam unit-unit, melakukan sintesis, menyusun kedalam pola, memilih mana yang penting dan yang akan dipelajari dan membuat kesimpulan.

Berpijak pada jenis penelitian R&D, pendekatan yang digunakan adalah Analisis Data Kualitatif. Dalam rekayasa perangkat lunak, proses ini difokuskan pada Analisis Kebutuhan Sistem (*System Requirement Analysis*). Langkah-langkah analisis yang dilakukan meliputi:

1. Analisa Kebutuhan (*Requirement Analysis*): Tahap awal yang berfokus pada pengumpulan data terkait disfungsi pelaporan operasional, identifikasi kebutuhan hak akses (Admin, Teknisi, Karyawan), serta pendefinisian parameter modul pelaporan dan *Knowledge Management System (KMS)*
2. *Desain Sistem (System Design)*: Proses penerjemahan syarat kebutuhan ke dalam perancangan perangkat lunak, meliputi perancangan arsitektur basis data relasional (*ERD*) *MySQL*, permodelan interaksi dan alur sistem menggunakan *UML (Unified Modeling Language)*, serta desain antarmuka (*User Interface*) responsif berbasis web.
3. Implementasi (*Implementation*): Tahap pengkodean (*coding*) di mana desain dikonversi menjadi baris program perangkat lunak menggunakan arsitektur *Model-View-Controller (MVC)* pada framework *Laravel*, serta eksekusi integrasi *Application Programming Interface (API)* untuk notifikasi *WhatsApp Gateway* dan *WebSockets*.
4. Pengujian (*Testing*): Tahap verifikasi sistem menggunakan metode pengujian *Black Box*

Testing untuk memastikan seluruh fungsionalitas aplikasi berjalan sesuai dengan spesifikasi yang diharapkan. Fokus pengujian dilakukan dengan mengamati hasil eksekusi dari sisi fungsional pengguna, meliputi pengujian kesesuaian rekam jejak system log, perubahan status siklus hidup tiket (Pending, Process, Done), serta keakuratan pengiriman notifikasi real-time tanpa adanya error.

5. Pemeliharaan (Maintenance): Tahap akhir di mana aplikasi yang telah lolos uji diserahkan kepada institusi untuk didistribusikan dalam lingkungan intranet guna mengelola pelaporan gangguan infrastruktur secara berkesinambungan.

2.4 Metode Penelitian

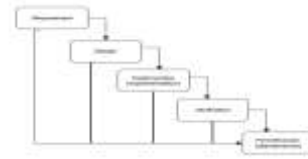
Jenis penelitian yang dipilih dan digunakan dalam penelitian ini adalah penelitian kuantitatif. Fokus dari penelitian ini adalah untuk mempelajari dan mengeksplorasi secara mendalam dan menyeluruh suatu kasus spesifik dalam konteks tertentu. Metode yang digunakan dalam pengumpulan data pada penelitian kuantitatif ini meliputi kuesioner dan dokumentasi.

2.5 Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang diterapkan dalam penelitian ini menggunakan pendekatan sekuensial linier, yaitu Model Waterfall menurut Pressman (2015). Metode ini mengharuskan setiap fase diselesaikan secara utuh sebelum beralih ke fase berikutnya.

Menurut Roger S. Pressman (2015), model Waterfall merupakan pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial (linier), di mana pengerjaannya mengalir ke bawah melewati fase-fase terstruktur. Dalam metode ini, setiap fase harus diselesaikan secara penuh

sebelum tim pengembang dapat beralih ke fase berikutnya.



Gambar 8 Waterfall Model

Sumber : Penelitian Mandiri 2026

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Sistem

Tahap perancangan sistem merupakan fase transisi kritis dalam rekayasa perangkat lunak yang bertugas menerjemahkan spesifikasi kebutuhan pengguna (user requirement specification) ke dalam bentuk representasi model arsitektur sebelum memasuki tahap pengkodean (construction). Pada tahap ini, pemodelan logika perangkat lunak dikembangkan secara terstruktur menggunakan metodologi *Unified Modeling Language (UML)* berstandar industri. Tujuan utama dari perancangan ini adalah menyediakan cetak biru (blueprint) komprehensif yang memvisualisasikan batasan fungsionalitas aktor, dinamika aliran kerja program, serta mekanisme sekuensial pertukaran data antar objek di dalam sistem. Perancangan pemodelan pada penelitian ini difokuskan pada tiga diagram perilaku (behavioral diagram), yaitu Use Case Diagram, Activity Diagram, dan Sequence Diagram, yang kemudian dilanjutkan dengan perancangan arsitektur basis data relasional.

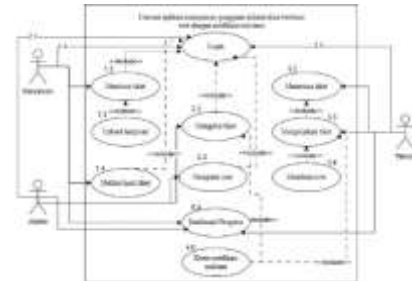
1.) Use Case Diagram

Use Case Diagram dirancang untuk memetakan batasan fungsionalitas sistem secara makro sekaligus mendeskripsikan interaksi eksternal antara para pengguna dengan perangkat lunak yang dibangun. Pada aplikasi manajemen gangguan infrastruktur ini, sistem

mengadopsi pendekatan Role-based access control (RBAC) yang membagi kewenangan operasional ke dalam tiga aktor fungsional dengan hak akses yang terisolasi secara tegas, yaitu:

1. Karyawan (Pelapor): Aktor yang merepresentasikan pengguna fasilitas kantor. Kewenangan aktor ini dibatasi pada fungsionalitas pembuatan tiket laporan baru (create ticket), pengunggahan bukti foto kerusakan (attachment), dan pemantauan status perbaikan secara privat (tracking). Karyawan tidak memiliki hak akses untuk mengubah status pengerjaan tiket.
2. Administrator (Single Point of Contact): Aktor manajerial yang bertindak sebagai gerbang kontrol utama. Administrator memiliki kewenangan penuh melakukan tahapan verifikasi awal (Incident Triage) untuk memvalidasi kategori gangguan, menentukan batas waktu penanganan (Service Level Agreement / SLA), mendelegasikan tiket kepada teknisi lapangan, serta mengelola data master pengguna.
3. Teknisi (IT Support): Aktor eksekutor yang bertugas menangani kendala teknis di lapangan. Kewenangan teknisi meliputi penerimaan delegasi tugas, pemutakhiran status siklus hidup tiket (Process hingga Done), pengajuan eskalasi insiden, serta kewajiban mengisi modul rekam jejak solusi (Knowledge Management System / KMS).

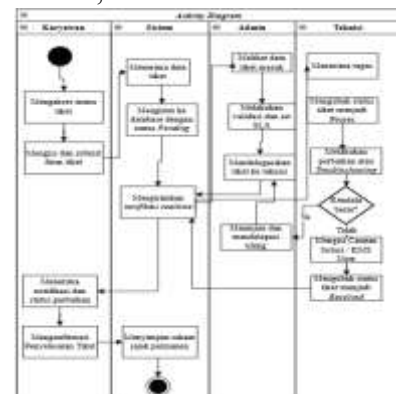
Pemodelan interaksi antara ketiga aktor tersebut terhadap sistem usulan direpresentasikan pada Gambar berikut:



Gambar 9 Use Case Diagram Usulan
Sumber : Penelitian Mandiri 2026

2.) Activity Diagram

Jika Use Case Diagram berfokus pada apa yang dilakukan oleh sistem, maka Activity Diagram dirancang untuk memodelkan dinamika alur kerja (workflow) logika prosedural sistem secara kronologis dari kondisi awal (initial node) hingga kondisi akhir (final node). Guna memperjelas perpindahan tanggung jawab data, pemodelan ini dibagi ke dalam empat partisi vertikal (swimlane) yang mewakili Karyawan, Sistem, Admin, dan Teknisi.



Gambar 10 Activity Diagram Usulan
Penelitian Mandiri 2026

Diagram diatas Alur aktivitas diinisiasi pada partisi Karyawan yang mengakses menu tiket untuk melakukan pengisian dan pengiriman formulir pelaporan. Setelah masukan (submit) dilakukan, eksekusi berpindah ke partisi Sistem. Sistem

menerima data tiket tersebut, menyimpannya ke dalam basis data dengan status awal Pending, dan secara otomatis memicu pemancaran notifikasi real-time untuk memberikan peringatan dini akan adanya tiket baru.

Peringatan tersebut menggeser alur kerja menuju partisi Admin. Pada tahap ini, Admin melihat data tiket masuk untuk melakukan tahap verifikasi awal (triage Process), yang meliputi validasi kerusakan dan penetapan tenggat waktu kepatuhan (Service Level Agreement / SLA). Setelah tervalidasi, Admin mendelegasikan tiket tersebut kepada teknisi terkait. Perintah pendelegasian ini kembali memicu Sistem untuk menembakkan notifikasi real-time kepada Teknisi lapangan.

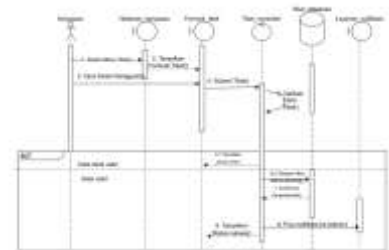
Selanjutnya, alur memasuki fase eksekusi pada partisi Teknisi. Teknisi merespons penugasan dengan memperbarui status sistem menjadi Proses sesaat sebelum melakukan tindakan perbaikan (troubleshooting). Di tengah proses perbaikan, sistem memodelkan sebuah percabangan logika (decision node) untuk mengevaluasi tingkat kepelikan insiden dengan kondisi "Kendala Berat?". Apabila hasil evaluasi bernilai 'Ya', alur akan berbalik menuju partisi Admin (Eskalasi) agar Admin dapat meninjau dan mendelegasikan ulang tiket tersebut. Namun, jika hasil evaluasi bernilai 'Tidak', Teknisi diwajibkan untuk melengkapi formulir rekam jejak aset pada modul Knowledge Management System (KMS) dan mengubah status tiket menjadi Resolved.

Pemutakhiran status Resolved tersebut menginstruksikan Sistem untuk ketiga kalinya memancarkan notifikasi real-time yang diarahkan kembali ke Karyawan pelapor (melewati fork node sinkronisasi). Karyawan menerima notifikasi, melihat status hasil perbaikan, dan melakukan verifikasi akhir dengan mengubah status tiket menjadi Done. Pada tahap terminasi, Sistem akan

menangkap konfirmasi penyelesaian tersebut dan menyimpannya sebagai rekam jejak permanen (System Log) ke dalam basis data sebelum mengakhiri seluruh siklus alur kerja operasional.

3.) Sequence Diagram

Rancangan selanjutnya adalah Sequence Diagram, yang memodelkan interaksi prosedural antar objek sistem secara kronologis berdasarkan waktu. Diagram ini sangat penting untuk menggambarkan logika pemrosesan dan pertukaran informasi yang terjadi antara antarmuka pengguna (user interface), kontrol sistem (control layer), dan basis data (database). Pada penelitian ini, dirancang Sequence Diagram untuk fungsionalitas utama, yaitu Pembuatan Tiket Gangguan oleh Karyawan (Pelapor). Diagram ini memperlihatkan bagaimana sistem usulan memastikan proses validasi data dilakukan dengan ketat sebelum rekam jejak tiket disimpan secara terpusat di dalam basis data internal aplikasi. Adapun rancangan Sequence Diagram sistem usulan dapat dilihat pada Gambar berikut:



Gambar 11 Sequence Diagram
Pembuatan Tiket

Sumber : Penelitian Mandiri 2026

3.2 Hasil Implementasi Sistem

Tahap implementasi sistem merupakan proses penerjemahan hasil perancangan (desain) ke dalam bentuk

antarmuka perangkat lunak yang sesungguhnya menggunakan bahasa pemrograman. Pada tahap ini, sistem sudah dapat dioperasikan. Berikut adalah penjelasan mengenai tampilan antarmuka pengguna (User Interface) beserta petunjuk cara penggunaan aplikasi manajemen gangguan infrastruktur:

1) Implementasi *Login dan Dashboard*

Pada gambar dibawah menjelaskan tampilan dasbor utama untuk hak akses karyawan. Halaman ini memuat ringkasan status tiket gangguan yang pernah dilaporkan oleh karyawan yang bersangkutan, serta menyajikan informasi singkat mengenai tiket yang sedang dalam antrian atau sudah diproses.:



Gambar 12 Implementasi *Login*
Sumber : Penelitian Mandiri 2026



Gambar 13 Implementasi *Dashboard Karyawan*
Sumber : Penelitian Mandiri 2026

2) Implementasi Halaman Pembuatan Tiket Karyawan

Pada gambar di bawah menjelaskan antarmuka formulir pelaporan gangguan infrastruktur. Karyawan harus mengisi kolom yang tersedia seperti subjek laporan,

kategori gangguan, deskripsi detail kerusakan, serta mengunggah foto bukti. Setelah menekan tombol simpan, tiket akan otomatis masuk ke antrian administrator.



Gambar 14 Implementasi Halaman Antarmuka Formulir Pembuatan Tiket
Sumber : Penelitian Mandiri 2026

3) Implementasi Halaman Utama Administrator

Pada gambar dibawah menjelaskan tampilan halaman utama dasbor administrator. Halaman ini menampilkan ringkasan data statistik secara menyeluruh, berupa total tiket masuk, tiket yang sedang dikerjakan, tiket selesai, serta persentase tingkat kepatuhan Service Level Agreement (SLA).



Gambar 15 Implementasi Antarmuka Dashboard Administrator
Sumber : Penelitian Mandiri 2026

4) Implementasi Halaman Antarmuka Pendelegasian Tiket

Pada gambar di bawah menjelaskan antarmuka manajemen tiket masuk. Admin menggunakan halaman ini untuk melakukan proses verifikasi. Admin dapat menetapkan tingkat prioritas penanganan dan memilih teknisi lapangan yang akan

ditugaskan untuk memperbaiki kerusakan tersebut.



Gambar 16 Implementasi halaman Antarmuka Pendelegasian Tiket
Sumber : Penelitian Mandiri 2026

5) Implementasi Halaman Antarmuka Pengaturan parameter SLA

Pada gambar di bawah menjelaskan tampilan menu pengaturan parameter sistem. Pada antarmuka ini, administrator diberikan hak untuk mengonfigurasi batas waktu penanganan (SLA) untuk setiap tingkat prioritas.



Gambar 17 Implementasi Halaman Antarmuka Pengaturan Parameter SLA

Sumber : Penelitian Mandiri 2026

6) Implementasi Antarmuka Dashboard Teknisi

Pada gambar di bawah menjelaskan tampilan dasbor utama teknisi lapangan. Halaman ini difokuskan untuk menyajikan rekapitulasi penugasan perbaikan yang sedang berstatus aktif, serta riwayat pekerjaan yang sudah berhasil diselesaikan.



Gambar 18 Implementasi Antarmuka Dashboard Teknisi

Sumber : Penelitian Mandiri 2026

7) Implementasi Formulir Knowledge Management System(KMS)

Pada gambar di bawah menjelaskan antarmuka pencatatan rekam jejak solusi teknis. Setelah perbaikan selesai, teknisi diwajibkan mengisi formulir KMS dengan memasukkan rincian jenis perangkat keras/lunak, nomor seri, serta penjabaran kronologi perbaikan yang telah dilakukan.



Gambar 19 Implementasi Halaman Formulir KMS pada Teknisi Sumber : Penelitian Mandiri 2026

8) Implementasi Antarmuka Data Tiket

Pada gambar di bawah menampilkan antarmuka data tiket yang merupakan kumpulan tiket yang sudah masuk sistem beserta statusnya



Gambar 20 Implementasi Halaman Antarmuka Data Tiket

Sumber : Penelitian Mandiri 2026

4. KESIMPULAN

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang telah dilakukan pada Aplikasi Manajemen

Gangguan Infrastruktur Kantor Berbasis Web dengan Notifikasi Real-time (Infradesk), maka dapat ditarik beberapa kesimpulan yang merupakan jawaban langsung atas rumusan masalah penelitian ini:

1. Penelitian ini telah berhasil merancang dan membangun arsitektur aplikasi manajemen gangguan infrastruktur kantor berbasis web (*Infradesk*) menggunakan framework PHP Laravel dengan arsitektur *Model-View-Controller* (MVC) dan sistem manajemen basis data relasional *MySQL*. Melalui pengujian fungsionalitas, sistem ini terbukti mampu mengotomatisasi dan mensentralisasikan seluruh pencatatan pelaporan gangguan infrastruktur kantor ke dalam satu basis data terpadu, sehingga berhasil mengeliminasi inefisiensi dokumentasi manual dan pelaporan konvensional (melalui lisan atau WhatsApp personal) yang sebelumnya bersifat ad-hoc dan tidak terstruktur.
2. Sistem notifikasi real-time dual-channel telah berhasil diimplementasikan dengan memadukan protokol WebSockets (seperti Pusher/Laravel Reverb) untuk pembaruan instan pada dasbor peramban (browser) karyawan, admin, dan teknisi, serta integrasi WhatsApp Gateway API untuk pengiriman pesan peringatan penugasan otomatis ke perangkat seluler teknisi. Berdasarkan hasil pengujian fungsional, sistem ini secara proaktif memancarkan sinyal peringatan seketika saat tiket didelegasikan oleh admin, yang secara signifikan berhasil mereduksi latensi respons teknisi dan meminimalkan waktu henti (*downtime*) operasional kantor.
3. Desain antarmuka pelacakan tiket (*live ticket tracking*) telah berhasil didesain dan diintegrasikan dengan baik pada dasbor karyawan pelapor. Fitur ini

memberikan transparansi penuh secara langsung mengenai tahapan siklus hidup penanganan tiket gangguan, mulai dari status Pending, Process, Resolved, hingga Done.s

4. *Elemen Knowledge Management System (KMS)* pragmatis telah berhasil diterapkan pada aplikasi dengan difokuskan pada modul pengisian spesifikasi perangkat (jenis, merek, tipe, dan nomor seri) serta catatan solusi (troubleshooting notes) oleh teknisi sebagai syarat mutlak sebelum tiket dapat ditutup (*Done*), yang diiringi dengan perekaman riwayat aktivitas otomatis (system log). Mekanisme ini terbukti efektif dalam mengekstraksi dan mengonversi pengetahuan taksi (*Tacit Knowledge*) dari teknisi lapangan menjadi aset intelektual digital perusahaan, guna membangun repositori pengetahuan historis institusi untuk mempercepat diagnosis dan penanganan kendala serupa di masa mendatang.

DAFTAR PUSTAKA

- Peraturan Gubernur Nomor 351 Tahun 2016
- Nonaka, I. & Takeuchi, H., *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, New York: Oxford University Press, 1995.
- Whitten, J. L. & Bentley, L. D., *System Analysis and Design Methods*, 7th ed., New York: McGraw-Hill, 2007.
- Pusher Development Team, *Pusher WebSockets and Push Notifications API*, Tersedia: <https://pusher.com>, 2026. [Akses Terakhir: 14 Juli 2026].
- Oracle Corporation, *MySQL Reference Manual*, Tersedia: <https://dev.mysql.com/doc>, 2026. [Akses Terakhir: 14 Juli 2026].