P-ISSN : 2580-4316
E-ISSN : 2654-8054
https://doi.org/10.37817/ikraith-informatika.v10i3

Fishbone dan PIECES untuk Perancangan Aplikasi Mobile Konsolidasi Notifikasi Teknologi Informasi Perbankan Berbasis Flutter

[1]Anggy Firmansyah, [2]Malabay
[1]Sistem Informasi, Universitas Esa Unggul, Jakarta Barat

E-mail: [1]anggyfirmansyah@student.esaunggul.ac.id, [2]malabay@esaunggul.ac.id

ABSTRAK

Notifikasi operasional teknologi informasi di PT Bank XYZ kerap terlambat direspons. Penyebabnya lima kanal yang berdiri sendiri-sendiri, yaitu Grafana, e-Centrix, Jira, GitHub, dan GitHub Actions. Setiap peringatan pun hanya bisa ditindaklanjuti dari komputer kerja. Penelitian terdahulu mengenai persoalan serupa umumnya bertumpu pada satu metode analisis saja, Fishbone untuk menelusuri akar penyebab atau PIECES untuk mengevaluasi dampaknya, sehingga akar masalah dan dampaknya pada sistem berjalan jarang dihubungkan secara eksplisit. Studi ini menjawab celah itu dengan memadukan kedua metode lewat pemetaan kausal-efek. Setiap akar penyebab hasil telusuran Fishbone dipetakan dampaknya lewat PIECES, sehingga kebutuhan aplikasi yang dirumuskan dapat dirunut balik ke sumber masalahnya. Pendekatan ini diterapkan untuk merancang sekaligus mengimplementasikan frontend aplikasi mobile Employee Access sebagai pusat notifikasi terpadu, memakai framework Flutter dengan metode Waterfall. Kebutuhan aplikasinya lahir dari wawancara terhadap lima responden lintas fungsi teknologi informasi, menghasilkan empat belas kebutuhan fungsional dan nonfungsional yang berwujud menjadi tujuh kelompok layar aplikasi. Pengujian Black Box atas dua puluh skenario yang mencakup dua belas use case berhasil sepenuhnya (100%), diperkuat hasil User Acceptance Testing terhadap kelima responden yang sama, yang menghasilkan skor System Usability Scale rata-rata 73,5 (kategori Grade C/OK). Rancangan antarmukanya sudah selaras dengan kebutuhan pengguna dan layak dijalankan secara fungsional. Pengaruhnya terhadap percepatan waktu respons insiden di lingkungan produksi belum terukur, dan itu menuntut kajian lanjutan.

Kata kunci : aplikasi mobile, Flutter, konsolidasi notifikasi, Fishbone, PIECES, Waterfall

ABSTRACT

Operational information technology notifications at PT Bank XYZ often receive late responses. The problem originates from five channels that stand on their own, namely Grafana, e-Centrix, Jira, GitHub, and GitHub Actions. Every alert can also be acted on only through a desktop computer. Prior research on similar problems generally relies on a single analysis method, Fishbone to trace root causes or PIECES to evaluate their impact, so the root causes and their impact on the running system are rarely connected explicitly. This study addresses that gap by combining both methods through causal-effect mapping. Each root cause traced through Fishbone is mapped to its impact through PIECES, so that the resulting application requirements can be traced back to their source. This approach was applied to design and implement the frontend of the Employee Access mobile application as a unified notification center, built on the Flutter framework with the Waterfall method. Application requirements emerged from interviews with five respondents across information technology functions, producing fourteen functional and nonfunctional requirements that took shape as seven groups of application screens. Black Box

https://doi.org/10.37817/ikraith-informatika.v10i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Testing on twenty scenarios covering twelve use cases succeeded entirely (100%), reinforced by User Acceptance Testing on the same five respondents, which produced an average System Usability Scale score of 73.5 (Grade C/OK category). The resulting interface design aligns with user needs and works functionally. Its influence on accelerating incident response time in a production environment remains unmeasured, and that demands further study.

Keyword : mobile application, Flutter, notification consolidation, Fishbone, PIECES, Waterfall

1. PENDAHULUAN

Divisi Teknologi Informasi bank umumnya mengoperasikan sistem pemantauan, pencatatan tiket, manajemen tugas, dan repositori kode secara paralel, sehingga makin banyak sistem yang berjalan, makin banyak pula kanal notifikasi yang harus dipantau personel teknologi informasi agar gangguan terdeteksi dan tertangani tepat waktu. Kecepatan pada tahap ini membawa konsekuensi regulatif: Peraturan Otoritas Jasa Keuangan Nomor 11/POJK.03/2022 tentang Penyelenggaraan Teknologi Informasi oleh Bank Umum mewajibkan bank menjaga ketahanan siber sekaligus menyampaikan notifikasi awal insiden kepada Otoritas Jasa Keuangan (Otoritas Jasa Keuangan, 2022).

PT Bank XYZ sudah mengoperasikan sejumlah sistem pendukung operasional: Grafana menangani alert, *e-Centrix* mencatat tiket, Jira mengelola tugas, serta *GitHub* dan *GitHub Actions* menopang repositori kode dan proses integrasi berkelanjutan. Secara individual semuanya bekerja baik, namun persoalan muncul pada distribusi notifikasinya yang hanya mengalir lewat kanal umum berupa surel dan *Google Chat*, tanpa satu pun kanal khusus yang mengonsolidasikannya.

Wawancara terhadap lima responden yang memegang fungsi kritis penanganan insiden di Kantor Pusat PT Bank XYZ memperlihatkan pola konsisten: notifikasi berprioritas tinggi kerap tenggelam di antara notifikasi berprioritas rendah, dengan pengalaman insiden yang beragam, mulai dari tertahannya deployment karena persetujuan pull request tidak segera diberikan, sampai alert yang nyaris terlewat saat pergantian shift. Bebannya bertambah karena tindak lanjut hanya mungkin lewat antarmuka desktop, ditambah profil kerja perangkat bergerak yang mengunci fungsi salin-tempel dan tangkapan layar.

Celah penelitian ini berangkat dari keterbatasan pendekatan terdahulu. Pemantauan infrastruktur dengan Grafana yang dipadukan notifikasi lewat aplikasi pesan instan (Saputra et al., 2025) maupun berbasis Zabbix dan Telegram (Husna & Rosyani, 2021) memang mempercepat penyampaian peringatan, tetapi notifikasinya bergerak satu arah dan tidak bisa ditindaklanjuti langsung dari perangkat bergerak — celah yang bahkan sudah disinggung Rahman et al. (2020) sebagai arah pengembangan berikutnya menuju aplikasi bergerak mandiri. Kajian push notification pada aplikasi bergerak (Naufal & Suartana, 2024; Wahyudin et al., 2024) juga umumnya berkutat pada satu sumber notifikasi saja, belum menyentuh agregasi lintas sistem pada konteks operasional teknologi informasi perbankan. Keterbatasan serupa muncul pada metode analisis: penelitian berkerangka PIECES umumnya berhenti pada evaluasi sistem berjalan tanpa menelusuri penyebabnya, sedangkan penelitian berdiagram Fishbone berhenti pada identifikasi akar masalah tanpa lanjut ke tahap perancangan. Sejauh penelusuran penulis, belum ada rancangan antarmuka bergerak yang mengonsolidasikan notifikasi lintas sistem sekaligus menyediakan aksi tindak lanjut, dengan kebutuhan yang

P-ISSN : 2580-4316
E-ISSN : 2654-8054
https://doi.org/10.37817/ikraith-informatika.v10i3

dirumuskan lewat penelusuran sebab-dampak yang terhubung.

Aplikasi Employee Access menjawab kebutuhan tersebut sebagai pusat notifikasi terpadu sekaligus sarana aksi cepat bagi personel teknologi informasi. Penelitian ini berfokus pada sisi frontend, yakni perancangan dan implementasi antarmuka dengan framework Flutter, sementara backend dan integrasi antarsistem masuk ranah pengembangan lanjutan. Sasarannya ada dua: merancang dan mengimplementasikan frontend aplikasi mobile Employee Access dengan metode Waterfall, lalu menguji fungsionalitas dan penerimaan penggunanya lewat Black Box Testing dan User Acceptance Testing.

2. LANDASAN TEORI

Metode *Waterfall* menempatkan tahapan pengembangan perangkat lunak dalam urutan ketat, meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian, dengan setiap tahap harus tuntas sebelum tahap berikutnya bergulir (Pressman & Maxim, 2020), dan cocok untuk pengembangan dengan kebutuhan yang sudah terbaca jelas sejak tahap analisis (Fahrezi et al., 2021). Penerapan metode *Prototype* pada pengembangan aplikasi web dan Android untuk kebutuhan operasional internal perusahaan (Andrias & Malabay, 2022; Prayoga & Malabay, 2024) turut memperkuat gambaran bahwa pendekatan SDLC terstruktur lazim dipakai untuk kebutuhan sistem internal semacam ini, sekalipun domain permasalahan ketiga kajian tersebut berbeda dari penelitian ini.

Diagram *Fishbone,* juga dikenal sebagai diagram Ishikawa, mengidentifikasi dan mengelompokkan penyebab suatu akibat ke dalam lima faktor: manusia, metode, teknologi, informasi, dan lingkungan (Rahayu & Oktarina, 2024). Kerangka PIECES bekerja pada sisi berbeda, yakni mengevaluasi sistem lewat enam aspek: *performance, information, economics, control, efficiency,* dan *service* (Lestari et al., 2021; Mustikasari et al., 2022).

Flutter merupakan *framework* pengembangan aplikasi lintas platform berbasis bahasa Dart, yang memungkinkan satu basis kode membangun aplikasi Android dan iOS sekaligus (Ariyanto et al., 2025). Manajemen status pada penelitian ini bersandar pada pola Bloc/Cubit, yang memisahkan logika bisnis dari lapisan antarmuka lewat konsep *state* dan *event*.

Push notification merupakan mekanisme penyampaian pesan dari peladen (*server*) ke perangkat pengguna tanpa memerlukan permintaan aktif dari perangkat, berbeda dengan *polling* yang mengharuskan aplikasi memeriksa pembaruan secara berkala sehingga kurang efisien dan menimbulkan jeda informasi.

Firebase Cloud Messaging (FCM) adalah layanan Google untuk mengirimkan notifikasi ke aplikasi pada platform Android, iOS, maupun web, yang penerapannya telah dibuktikan pada berbagai konteks, antara lain aplikasi pemesanan (Naufal & Suartana, 2024) dan sistem notifikasi *real-time* untuk manajemen persuratan multiplatform yang memadukan FCM dengan *Application Programming Interface*/API (Wahyudin et al., 2024)

Black Box Testing menguji perangkat lunak dari sisi fungsionalitas tanpa menyentuh struktur kode internalnya (Hendri et al., 2020), analisis tematik mengidentifikasi pola tema yang berulang dari data wawancara (Braun & Clarke, 2006), dan triangulasi sumber memeriksa konsistensi temuan antarresponden (Sugiyono, 2022).

Penerimaan pengguna terhadap antarmuka diukur melalui *User Acceptance Testing* (UAT), yaitu pengujian yang melibatkan pengguna

https://doi.org/10.37817/ikraith-informatika.v10i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

akhir langsung untuk memvalidasi kesesuaian perangkat lunak dengan kebutuhan hasil analisis — berbeda dari *Black Box Testing* yang menilai kesesuaian fungsional terhadap rancangan. *System Usability Scale* (SUS), instrumen umum dalam UAT untuk mengukur usability, terdiri atas sepuluh pernyataan berskala Likert 1–5 yang menghasilkan skor 0–100 lengkap dengan kategori interpretasi standar dari *Not Acceptable* hingga *Best Imaginable* (Sukma et al., 2024).

3. METODOLOGI

Penelitian ini berpijak pada pendekatan kualitatif konstruktif yang dipadukan dengan metode pengembangan *Waterfall*, dengan objek personel teknologi informasi PT Bank XYZ Kantor Pusat pada empat unit kerja: IT Arsitektur dan Engineer, *Production Support* (Core Banking dan Delivery Channel), IT Data Management, serta IT Data Center Management.

Wawancara semi terstruktur terhadap lima responden terpilih secara *purposive* menjadi sumber data primer, dilengkapi observasi sistem berjalan. Analisis tematik menyaring pola tema berulang dari transkrip wawancara dan menghasilkan sebelas tema permasalahan, yang kekuatannya diuji lewat triangulasi sumber; delapan di antaranya disebut oleh seluruh responden, pertanda permasalahan bersifat sistemik.

Diagram *Fishbone* memetakan akar penyebab permasalahan pada lima faktor, lalu kerangka PIECES membaca dampaknya terhadap sistem berjalan, dan pemetaan kausal-efek menautkan keduanya secara eksplisit sehingga setiap kebutuhan aplikasi berpangkal pada akar permasalahan. Empat belas kebutuhan fungsional dan nonfungsional lahir dari proses ini.

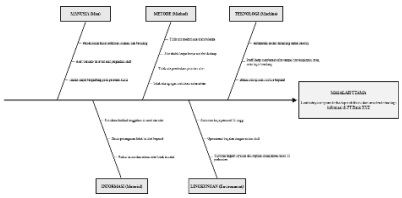
Pengembangan sistem menempuh empat tahapan *Waterfall*. Tahap analisis kebutuhan menghasilkan kebutuhan fungsional dan nonfungsional seperti terurai di atas; tahap perancangan menuangkan proses ke diagram UML (*Use Case, Activity, Class,* dan *Sequence Diagram*) ditambah rancangan visual lewat Figma; dan tahap implementasi berjalan di atas *framework Flutter* versi 3.44.6 dengan Dart versi 3.12.2 serta manajemen status pola Bloc/Cubit, dengan data tampilan bersifat tiruan karena pembangunan *backend* berada di luar lingkup penelitian.

Tahap pengujian memakai dua metode yang saling melengkapi. *Black Box Testing* dijalankan atas dua puluh skenario yang mencakup dua belas *use case*, menyoroti kesesuaian fungsionalitas dan navigasi antarmuka terhadap rancangan, bukan struktur kode internal. User Acceptance Testing dijalankan pada kelima responden yang sama dengan tahap analisis kebutuhan, terdiri atas empat skenario tugas inti (*acknowledge* notifikasi, menambah catatan kerja, eskalasi ke tiket kerja, dan membuka tools eksternal) serta kuesioner *System Usability Scale* sepuluh pernyataan untuk memvalidasi penerimaan pengguna.

4. HASIL DAN PEMBAHASAN

a. Analisis Akar Masalah dengan Diagram Fishnone

Masalah utama yang diidentifikasi adalah lambatnya respons terhadap notifikasi dan insiden teknologi informasi di PT Bank XYZ. Diagram *Fishbone* mengelompokkan akar penyebabnya ke dalam lima faktor sebagaimana tergambar pada Gambar 1 dan dirinci pada Tabel 1.



Gambar 1. Diagram Fishbone

P-ISSN : 2580-4316
E-ISSN : 2654-8054
https://doi.org/10.37817/ikraith-informatika.v10i3

Tabel 1. Akar penyebab berdasarkan diagram Fishbone

Faktor	*Akar Penyebab*
Manusia (*Man*)	Tindak lanjut insiden bergantung pada personel kunci yang tidak selalu berada di depan komputer kerja; perpindahan tanggung jawab pada pergantian *shift* berisiko menyebabkan *alert* tidak tertangani; pemeriksaan kanal notifikasi dilakukan secara manual dan berulang
Metode (*Method*)	Tidak tersedia mekanisme agregasi notifikasi dari Grafana, e-Centrix, Jira, GitHub, dan GitHub Actions; tidak terdapat pembedaan prioritas antara notifikasi kritikal dan notifikasi rutin; alur tindak lanjut hanya tersedia melalui antarmuka *desktop*; tidak tersedia mekanisme *acknowledge* yang terlihat lintas personel dan *shift*
Teknologi (*Machine*)	Belum tersedia aplikasi bergerak terpusat untuk notifikasi operasional teknologi informasi; profil kerja pada perangkat bergerak membatasi fungsi salin-tempel dan tangkapan layar serta mengharuskan proses masuk berulang; antarmuka sistem yang ada dirancang untuk *desktop* dan kurang optimal pada perangkat bergerak
Informasi (*Material*)	Notifikasi kritikal tenggelam dalam volume surel dan pesan yang tinggi; status penanganan tidak terlihat secara terpusat; waktu penerimaan dan penyelesaian *alert* tidak tercatat
Lingkungan (*Environment*)	Operasional Divisi Teknologi Informasi berjalan dengan pola berbeda antarunit sehingga insiden dapat terjadi kapan saja; mobilitas kerja personel tergolong tinggi; terdapat tuntutan pemenuhan tingkat layanan (*service level*) dan regulasi manajemen risiko teknologi informasi

b. Analisis Dampak dengan Kerangka PIECES
Dampak setiap akar penyebab di atas terhadap sistem yang berjalan dianalisis menggunakan kerangka PIECES sebagaimana disajikan pada Tabel 2.

Tabel 2. Analisis PIECES terhadap sistem berjalan

Aspek	*Kondisi Sistem Berjalan*	*Indikator Kelemahan*
Performance	Waktu respons insiden memanjang karena aksi hanya dapat dilakukan melalui komputer kerja	Proses *deployment* tertahan menunggu persetujuan *pull request*, penanganan *alert* kanal terlambat
Information	*Alert* kritikal tidak menonjol karena tercampur notifikasi berprioritas rendah	Tiket eskalasi tertimbun surel lain, *alert* nyaris terlewat pada pergantian *shift*
Economics	Waktu kerja terserap untuk memeriksa banyak kanal dan berpindah perangkat	Pemeriksaan surel dilakukan berkali-kali setiap hari, potensi kerugian akibat gangguan layanan kanal
Control	Status penanganan tidak tercatat dan tidak terlihat terpusat	Tidak tersedia jejak audit, kinerja respons tidak dapat diukur
Efficiency	Proses respons memerlukan langkah berulang (membuka profil kerja, masuk ke setiap sistem)	Proses masuk berulang memakan waktu, pembatasan salin-tempel dan tangkapan layar memperlambat eskalasi
Service	Keterlambatan respons internal berdampak pada layanan eksternal	Keluhan pengguna kantor cabang, keluhan dari sisi kanal ATM dan *mobile banking*

c. Pemetaan Kausal Efek dan Kebutuhan Solusi
Karena penelitian ini memakai lebih dari satu metode analisis, hubungan sebab-akibat antara Fishbone dan PIECES dipetakan secara eksplisit sebagaimana disajikan pada Tabel 3, sehingga setiap kebutuhan aplikasi yang dirumuskan dapat ditelusuri kembali ke akar permasalahannya, bukan sekadar gejala yang tampak di permukaan.

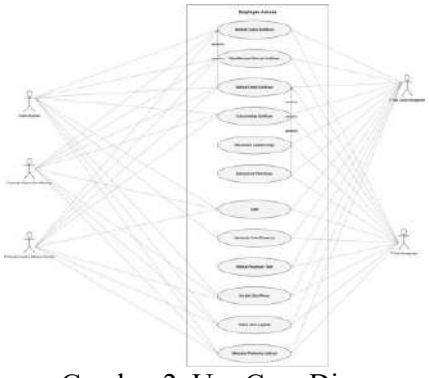
https://doi.org/10.37817/ikraith-informatika.v10i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Tabel 3. Pemetaan Kausal-Efek dan Kebutuhan Solusi

Akar Penyebab (Fishbone)	*Dampak (PIECES)*	*Kebutuhan Solusi (Requirement)*
Notifikasi tersebar tanpa pembedaan prioritas (*Method*, *Material*)	*Information*, *Performance*	Agregasi notifikasi lintas sistem dengan penanda prioritas
Aksi tindak lanjut hanya melalui *desktop* dan terhambat profil kerja (*Machine*)	*Performance*, *Efficiency*	Aksi cepat dari perangkat bergerak, yaitu *acknowledge*, komentar, tindak lanjut pada tools eksternal terintegrasi (termasuk *merge pull request* dan *re-run* proses CI/CD), serta pembuatan dan pembaruan tiket
Status penanganan tidak terlihat lintas personel (*Man*, *Material*)	*Control*	Status *acknowledge* yang tersinkron beserta riwayat penanganan
Volume notifikasi tinggi tanpa penyaringan (*Material*)	*Economics*, *Efficiency*	Penyaringan dan pengelompokan notifikasi
Mobilitas tinggi dan operasional berkelanjutan (*Environment*)	*Service*	Notifikasi *push* dengan rancangan antarmuka *mobile-first*
Kebutuhan pengamanan data perbankan	*Control*	Pembatasan tampilan informasi sensitif dan pengendalian akses

Pemetaan tersebut memperlihatkan bahwa fragmentasi kanal notifikasi tanpa pembedaan prioritas menyebabkan *alert* kritikal tenggelam, yang kemudian diperberat oleh keterbatasan tindak lanjut pada perangkat bergerak, sehingga waktu respons insiden memanjang dan pada akhirnya berdampak pada layanan nasabah serta pengendalian operasional. Hasil pemetaan kausal-efek antara *Fishbone* dan PIECES ini melahirkan empat belas kebutuhan fungsional dan nonfungsional, yang kemudian diterjemahkan menjadi tujuh kelompok layar aplikasi, yaitu portal masuk, dasbor notifikasi terkonsolidasi, detail notifikasi dan *acknowledge*, integrasi tools eksternal, daftar tiket kerja, pembuatan tiket kerja, serta pengaturan akun dan peran, sebagaimana tergambar pada Gambar 2.

Gambar 2 memperlihatkan lima aktor independen sesuai unit kerja masing-masing, yaitu System Engineer, Production Support (Core Banking), Production Support (Delivery Channel), IT Data Management, dan IT Data Center Management, yang masing-masing terhubung langsung ke *use case* yang relevan tanpa aktor umum. Delapan dari dua belas *use case*, yaitu Login, Melihat Dasbor Notifikasi, Memfilter dan Mencari Notifikasi, Melihat Detail Notifikasi, Acknowledge Notifikasi, Beralih Aktor/Peran, Keluar Sesi, dan Mengatur Preferensi Aplikasi, dapat diakses oleh seluruh aktor. *Use case* Membuka Tools Eksternal hanya diakses System Engineer, IT Data Management, dan IT Data Center Management, karena Production Support menangani notifikasi bersumber Grafana yang tidak memiliki mekanisme buka tools eksternal, sedangkan *use case* Menambah Catatan Kerja, Eskalasi ke Tiket Kerja, dan Melihat Registrasi Tiket hanya diakses IT Data Management dan IT Data Center Management sesuai peran keduanya sebagai fungsi triase dan eskalasi. Kedua belas *use case* itu merangkum seluruh kebutuhan fungsional yang telah dirumuskan, mulai dari melihat dasbor notifikasi sampai mengatur preferensi aplikasi.



Gambar 2. Use Case Diagam

P-ISSN : 2580-4316
E-ISSN : 2654-8054
https://doi.org/10.37817/ikraith-informatika.v10i3

Dasbor Notifikasi memikul peran utama sebagai perwujudan kebutuhan mengonsolidasikan notifikasi dari Grafana, e-Centrix, Jira, GitHub, dan GitHub Actions, lengkap dengan penanda prioritas, sebagaimana tampak pada Gambar 3. Rancangan ini menyasar langsung akar masalah pada faktor metode dan informasi, yakni ketiadaan agregasi notifikasi serta pembedaan prioritas (lihat pemetaan pada Tabel 3). Layar Detail Notifikasi melengkapinya lewat mekanisme *acknowledge* dan catatan kerja kolaboratif, yang mengisi celah pencatatan status penanganan *alert* sebagaimana teridentifikasi pada analisis PIECES (Tabel 2).



Gambar 3. Implementasi Dasbor

Kebutuhan menindaklanjuti notifikasi tanpa berpindah aplikasi terwujud lewat pola antarmuka pembungkus bernama *External Tool Simulator*, yang menangani empat sumber tools eksternal secara konsisten, yaitu Jira, GitHub, GitHub Actions, dan e-Centrix. Rancangan peninjauan *pull request*, lengkap dengan status persetujuan peninjau serta mekanisme *branch protection*, bertahan utuh sejak eksplorasi desain awal hingga hasil implementasi akhir. Rancangan itulah yang menjawab kebutuhan inti responden untuk menyetujui *pull request* langsung dari perangkat bergerak.

Aplikasi *Employee Access* terimplementasi dengan Flutter dan berjalan pada versi 1.0.4 ALPHA. Struktur objek pada sisi *frontend* memodelkan dua lapisan. Lapisan model domain terdiri atas kelas AppUser, NotificationItem, CommentModel, dan TicketModel, sedangkan lapisan manajemen status berisi TriageState yang dikelola oleh TriageCubit. Struktur semacam ini merupakan wujud pola Bloc/Cubit yang memisahkan logika bisnis dari lapisan antarmuka.

Pengujian *Black Box* atas dua puluh skenario yang mencakup dua belas *use case* menghasilkan tingkat keberhasilan seperti tersaji pada Tabel 4.

Tabel 4. Hasil pengujian Black Box per kelompok use case

Kelompok Use Case	*Jumlah Skenario*	*Berhasil*	*Persentase*
Login (UC-01)	2	2	100%
Notifikasi (UC-02 s.d. UC-06)	6	6	100%
Tools Eksternal (UC-07)	4	4	100%
Tiket (UC-08 s.d. UC-09)	4	4	100%
Akun dan Preferensi (UC-10 s.d. UC-12)	4	4	100%
Total	**20**	**20**	**100%**

Seluruh dua puluh skenario (100%) dinyatakan berhasil. Sub-fungsi menjalankan ulang proses CI/CD pada simulasi GitHub Actions sempat gagal pada iterasi pengujian sebelumnya, karena aksi konfirmasi belum terhubung dengan pembaruan status di lapisan manajemen *state*. Perbaikan sudah dilakukan, dan sub-fungsi tersebut lolos pada pengujian akhir. Angka keberhasilan ini mencerminkan kesesuaian fungsional

https://doi.org/10.37817/ikraith-informatika.v10i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

antarmuka terhadap rancangan, bukan ukuran penerimaan pengguna akhir maupun bukti percepatan waktu respons insiden di lingkungan kerja nyata, mengingat pengujian berjalan di atas data tiruan dan pembangunan *backend* berada di luar lingkup penelitian.

Penerimaan pengguna terhadap antarmuka yang telah diimplementasikan turut diuji melalui *User Acceptance Testing* terhadap kelima responden yang sama dengan tahap analisis kebutuhan. Empat skenario tugas representatif, yaitu *acknowledge* notifikasi, menambah catatan kerja, eskalasi ke tiket kerja, dan membuka tools eksternal, seluruhnya berhasil diselesaikan secara mandiri oleh kelima responden sebagaimana disajikan pada Tabel 5.

Tabel 5. Hasil pengujian skenario User Acceptance Testing

No	*Skenario*	*Hasil yang Diharapkan*	*Hasil Aktual*
1	Acknowledge Notifikasi	Status notifikasi berubah menjadi "ditangani" tanpa kendala	Berhasil
2	Menambah Catatan Kerja	Catatan tersimpan dan tampil di riwayat penanganan	Berhasil
3	Eskalasi ke Tiket Kerja	Form tiket terisi otomatis, tiket berhasil terkirim	Berhasil
4	Membuka Tools Eksternal	Simulator tools eksternal terbuka sesuai sumber notifikasi	Berhasil

Kuesioner *System Usability Scale* yang diisi kelima responden menghasilkan skor rata-rata 73,5, berada pada kategori Grade C (OK) menurut interpretasi Sukma et al. (2024), sebagaimana disajikan pada Tabel 6.

Tabel 6. Skor System Usability Scale per responden

Kode Responden	*Skor SUS*
R1 (System Engineer)	70,0
R2 (PS Core Banking)	75,0
R3 (PS Delivery Channel)	75,0
R4 (IT Data Management)	72,5
R5 (IT Data Center Management)	75,0
Rata-rata	**73,5**

Skor ini mengonfirmasi bahwa rancangan antarmuka Employee Access secara umum diterima baik oleh pengguna dari segi kemudahan penggunaan pada perangkat bergerak, melengkapi hasil pengujian fungsional Black Box di atas dengan bukti penerimaan pengguna secara langsung.

Angka 100% pada pengujian Black Box tersebut sejalan dengan capaian Mariyana dan Salim (2026), yang metodenya paling berdekatan dengan penelitian ini (*frontend* Flutter dengan Waterfall) dan juga menyentuh tingkat keberhasilan 100% pada sembilan skenario pengujian antarmuka. Capaian penelitian ini pun melampaui tingkat keberhasilan 91% (73 dari 80 skenario) yang diraih Naufal et al. (2025) pada pengujian *Black Box* aplikasi kesehatan bergerak. Perbedaan lain terletak pada pendekatan analisisnya. Mustikasari et al. (2022) memakai kerangka PIECES sebagai alat evaluasi tunggal, sementara Rahayu dan Oktarina (2024) berhenti pada diagram *Fishbone* tanpa melanjutkan ke tahap perancangan dan implementasi. Penelitian ini memadukan keduanya lewat pemetaan kausal-efek yang eksplisit, sehingga setiap kebutuhan aplikasi yang diuji bisa ditelusuri kembali sampai ke akar penyebabnya.

5. KESIMPULAN

Frontend aplikasi *mobile Employee Access* berhasil dirancang dan diimplementasikan sebagai pusat notifikasi operasional teknologi informasi bagi personel PT Bank XYZ, dibangun dengan *framework* Flutter melalui metode *Waterfall*. Kebutuhan aplikasinya bersumber dari wawancara terhadap lima responden, yang kemudian ditelaah lewat diagram *Fishbone* dan kerangka PIECES, lalu ditautkan melalui pemetaan kausal-efek. Empat belas kebutuhan fungsional dan nonfungsional lahir dari proses itu, dan berwujud menjadi tujuh kelompok layar aplikasi yang diakses oleh lima aktor sesuai unit kerja masing-masing. Pengujian *Black Box* atas dua puluh skenario yang mencakup dua belas *use case* berhasil sepenuhnya (100%), diperkuat oleh hasil *User Acceptance Testing* yang menghasilkan skor *System Usability Scale* rata-rata 73,5 (kategori Grade C/OK) serta seluruh skenario tugas yang diselesaikan secara mandiri oleh kelima responden, pertanda rancangan antarmuka sudah selaras dengan kebutuhan pengguna sekaligus berjalan secara fungsional. Nilai lebih penelitian ini terletak pada pemaduan *Fishbone* dan PIECES lewat pemetaan kausal-efek eksplisit dalam merumuskan kebutuhan aplikasi konsolidasi notifikasi pada konteks operasional teknologi informasi perbankan, sebuah jalur yang berbeda dari penelitian terdahulu yang umumnya bersandar pada satu metode analisis saja. Pengukuran dampak operasional terhadap percepatan waktu respons insiden secara nyata masih menunggu penelitian lanjutan, yang melibatkan integrasi *backend* serta pengujian pada lingkungan kerja sesungguhnya.

6. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing atas arahan dan bimbingan selama penelitian ini, kepada pihak manajemen dan staf Divisi Teknologi Informasi PT Bank XYZ atas kesempatan dan data yang diberikan selama proses penelitian, serta kepada kelima responden yang telah berpartisipasi dalam wawancara maupun pengujian penerimaan pengguna.

DAFTAR PUSTAKA

Andrias, A., & Malabay, M. (2022). Implementasi Aplikasi Web Sertifikat Vaksin Karyawan Berbasis Radio-Frequency Identification (Studi Kasus : PT Megatech Bio Energi). *Ikraith-Informatika*, *6*(3), 107–115. https://doi.org/10.37817/ikraith-informatika.v6i3.2212

Ariyanto, A. S. S., Mangkunegara, I. S., Triwibowo, D. N., & Feriyanto, Y. (2025). Perancangan Aplikasi Mobile Berbasis Flutter untuk Pemantauan Data Sensor IoT: Solusi bagi Manajemen Suhu dan Kelembapan. *JSAI: Journal Scientific and Applied Informatics*, *8*(1), 204–211. https://jurnal.umb.ac.id/index.php/JSAI/article/view/7724

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Fahrezi, K., Mulana, A. R., Melinda, S., Nurhaliza, N., & Mulyati, S. (2021). Penerapan Model Waterfall dalam Pengembangan Sistem Informasi Akademik Berbasis Web sebagai Sistem Pengolahan Nilai Siswa. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, *4*(2), 98. https://doi.org/10.32493/jtsi.v4i2.10196

Hendri, H., Hasiholan Manurung, J. W., Ferian, R. A., Hanaatmoko, W. F., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi

Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, *3*(2), 107. https://doi.org/10.32493/jtsi.v3i2.4694

Lestari, M., Haryani, E., & Wahyono, T. (2021). Analisis Kelayakan Sistem Informasi Akademik Universitas Menggunakan PIECES dan TELOS. *Jurnal Teknik Informatika Dan Sistem Informasi*, *7*(2), 373–380. https://doi.org/10.28932/jutisi.v7i2.3612

Mustikasari, D., Karaman, J., & Cobantoro, A. F. (2022). Analysis of A PIECES Framework of A Localhost Web-Based Income Statement EPOSAL Application. *INTENSIF: Jurnal Ilmiah Penelitian Teknologi Dan Penerapan Sistem Informasi*, *6*(2), 271–284. https://doi.org/10.29407/intensif.v6i2.17465

Naufal, M. A. F. A., & Suartana, I. M. (2024). Penerapan Push Notification Booking Perlengkapan Outdoor Menggunakan Firebase Cloud Messaging. *JINACS: Journal of Informatics and Computer Science*, *6*(2), 573–580. https://doi.org/10.26740/jinacs.v6n02.p573-580

Otoritas Jasa Keuangan. (2022). *Peraturan Otoritas Jasa Keuangan Nomor 11/POJK.03/2022 tentang Penyelenggaraan Teknologi Informasi oleh Bank Umum*. https://www.ojk.go.id/id/regulasi/Pages/Penyelenggaraan-Teknologi-Informasi-Oleh-Bank-Umum.aspx

Prayoga, Y., & Malabay, M. (2024). Aplikasi Pemesanan Minuman Berbasis Android Di Kedai Kopi (Studi Kasus : Kedai Kopi Ngopidulu). *IKRA-ITH Informatika : Jurnal Komputer Dan Informatika*, *8*(2), 99–108. https://doi.org/10.37817/ikraith-informatika.v8i2.2964

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach*.

Rahayu, I. P., & Oktarina, A. B. (2024). Analisis Fishbone pada Program Aplikasi Pelayanan Publik SIPRAJA di Desa Tambak Sumur Kecamatan Waru Kabupaten Sidoarjo. *Jurnal Media Akademik (JMA)*, *2*(5). https://doi.org/10.62281/v2i5.323

Rahman, D., Amnur, H., & Rahmayuni, I. (2020). Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, *1*(4), 133–138. https://doi.org/10.30630/jitsi.1.4.19

Sugiyono. (2022). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*.

Wahyudin, A., Anisyah, A., & Ahmaddifa, D. (2024). Pengembangan Sistem Notifikasi Real-Time untuk Aplikasi Manajemen Persuratan Multiplatform Menggunakan Firebase Cloud Messaging dan Application Programming Interface. *Jurnal Teknik Informatika UNIKA St. Thomas (JTIUST)*, *9*(2), 284–300. https://ejournal.ust.ac.id/index.php/JTIUST/article/view/4437

Mariyana, M., & Salim, A. (2026). Perancangan Aplikasi Mobile Ternak Bareng Menggunakan Framework Flutter. *Sistematis: Jurnal Ilmiah Sistem Informasi*, *2*(2), 173–185. https://doi.org/10.69533/y2mwyj60

Naufal, A., Nuryasin, I., Muthohirin, B. F., Titani, M., Akrom, A., & Fadlil, A. (2025). User Acceptance Testing Melalui Evaluasi Blackbox dan ISO 9241-11 Terhadap Aplikasi Kesehatan Mobile: MAHATI. *JIPI: Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika*, *10*(4), 4296–4304. https://doi.org/10.29100/jipi.v10i4.7377