

# Pengembangan dan Evaluasi *Usability* Aplikasi Mobile MOOS Berbasis Jetpack Compose

Rangga Gumilang<sup>1</sup>, Ronggo Alit<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Universitas Negeri Surabaya

<sup>1</sup>[rangga.22150@mhs.unesa.ac.id](mailto:rangga.22150@mhs.unesa.ac.id)

<sup>2</sup>[rongoalit@unesa.ac.id](mailto:rongoalit@unesa.ac.id)

**Abstrak**— Sistem internal hotel MOOS (Mercure Online Order System) versi web yang berjalan di atas platform osTicket memiliki keterbatasan aksesibilitas dan antarmuka yang tidak ramah perangkat mobile, sehingga menghambat kinerja staf teknisi dan housekeeping yang memiliki mobilitas tinggi di lapangan. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi MOOS versi mobile berbasis Jetpack Compose yang mendukung aktivitas operasional staf hotel secara fleksibel dan real-time, serta mengukur tingkat usability aplikasi menggunakan metode System Usability Scale (SUS). Penelitian menggunakan pendekatan kuantitatif dengan jenis Research and Development (R&D). Pengembangan sistem dilaksanakan menggunakan metode V-Model yang mencakup tahap analisis kebutuhan, perancangan sistem, perancangan arsitektur, perancangan modul, implementasi, hingga pengujian. Penelitian dilaksanakan di Hotel Mercure Surabaya Grand Mirama pada periode Desember 2025 hingga Maret 2026. Subjek penelitian adalah 21 staf internal hotel yang dipilih menggunakan teknik purposive sampling. Instrumen penelitian terdiri atas kuesioner SUS, skenario tugas, dan lembar observasi. Data dianalisis menggunakan perhitungan skor SUS, statistik deskriptif, dan uji reliabilitas Cronbach's Alpha. Hasil penelitian menunjukkan bahwa seluruh 10 kebutuhan fungsional berhasil diimplementasikan dan semua skenario pengujian sistem menghasilkan status pass. Evaluasi usability menghasilkan skor SUS rata-rata sebesar 86,43 yang termasuk dalam kategori Excellent dan Acceptable berdasarkan pedoman Bangor et al. (2008), dengan overall task success rate 100% dan nilai Cronbach's Alpha 0,835. Aplikasi MOOS Mobile dinyatakan layak diimplementasikan sebagai alat kerja operasional staf hotel.

**Kata Kunci**— MOOS Mobile, Jetpack Compose, Usability, System Usability Scale, V-Model

## I. PENDAHULUAN

Perkembangan teknologi informasi dalam satu dekade terakhir telah mendorong transformasi digital di berbagai sektor industri, termasuk perhotelan. Digitalisasi operasional hotel terbukti meningkatkan efisiensi proses pelayanan, akurasi data, dan kecepatan respons terhadap kebutuhan tamu [1]. Konsep smart hotel yang mengadopsi teknologi mobile dan Internet of Things (IoT) tidak hanya mempercepat layanan tamu, tetapi juga menciptakan sistem operasional yang lebih adaptif dan hemat sumber daya [2].

Hotel Mercure Surabaya Grand Mirama menerapkan sistem internal bernama MOOS (Mercure Online Order System) berbasis web di atas platform osTicket untuk mendukung

operasional hotel, khususnya penanganan tiket perawatan fasilitas dan keluhan tamu. MOOS memiliki fitur utama seperti pembuatan tiket (work order), pelacakan status pekerjaan, pengaturan prioritas berdasarkan Service Level Agreement (SLA), dan pelaporan aktivitas. Meskipun fungsional, antarmuka web osTicket dirancang berorientasi pada penggunaan desktop sehingga tidak responsif ketika diakses melalui perangkat mobile. Tampilan menu yang padat, tombol navigasi berukuran kecil, dan penyajian data dalam format tabel statis mengharuskan pengguna melakukan pembesaran layar dan pengguliran berulang. Kondisi ini menghambat staf teknisi dan housekeeping yang memiliki mobilitas tinggi di lapangan, mengakibatkan keterlambatan pembaruan status pekerjaan dan peningkatan risiko kesalahan input.

Pengembangan aplikasi mobile merupakan solusi yang memungkinkan staf mengakses sistem secara real-time dari perangkat smartphone. Aplikasi mobile terbukti meningkatkan kecepatan komunikasi dan efisiensi kolaborasi antar-departemen di industri perhotelan, khususnya pada fungsi maintenance dan guest service [3]. Dalam pengembangan Android modern, Jetpack Compose sebagai framework UI deklaratif dari Google mampu meningkatkan produktivitas pengembang dan menghasilkan antarmuka yang konsisten sesuai prinsip Material Design 3 dengan performa rendering yang kompetitif [4].

Penelitian ini menerapkan metode V-Model untuk pengembangan sistem karena menawarkan keterkaitan langsung antara setiap tahap perancangan dan tahap pengujian, memastikan verifikasi berlapis sebelum produk digunakan pengguna akhir [5]. Tingkat kelayakan aplikasi diukur menggunakan System Usability Scale (SUS), instrumen evaluasi usability yang telah teruji pada berbagai antarmuka teknologi [6][7][8]. Penelitian ini bertujuan: (1) merancang dan mengembangkan aplikasi MOOS versi mobile berbasis Jetpack Compose; dan (2) mengukur tingkat usability aplikasi menggunakan SUS untuk menilai kelayakan implementasinya sebagai alat kerja operasional staf hotel.

## II. METODE PENELITIAN

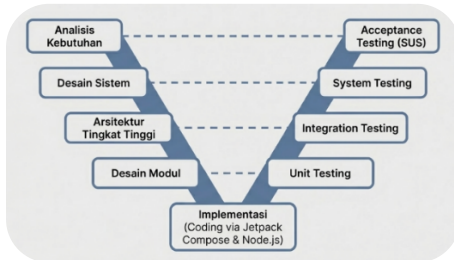
### A. Gambaran Umum Penelitian

Penelitian menggunakan pendekatan kuantitatif dengan jenis Research and Development (R&D). Penelitian dilaksanakan di Hotel Mercure Surabaya Grand Mirama, Jalan Raya Darmo, Surabaya, pada periode Desember 2025 hingga Maret 2026.

Teknik pengambilan sampel yang digunakan adalah purposive sampling dengan kriteria: terlibat langsung dalam penggunaan MOOS, telah bekerja minimal tiga bulan, dan bersedia mengikuti sesi pengujian secara penuh. Jumlah sampel adalah 21 responden dari berbagai departemen (Engineering/Teknisi, Housekeeping, Front Office, Food & Beverage, IT/Sistem, dan lainnya). Variabel utama penelitian adalah usability aplikasi MOOS Mobile yang diukur menggunakan SUS mengacu pada standar ISO 9241-11 [6][7].

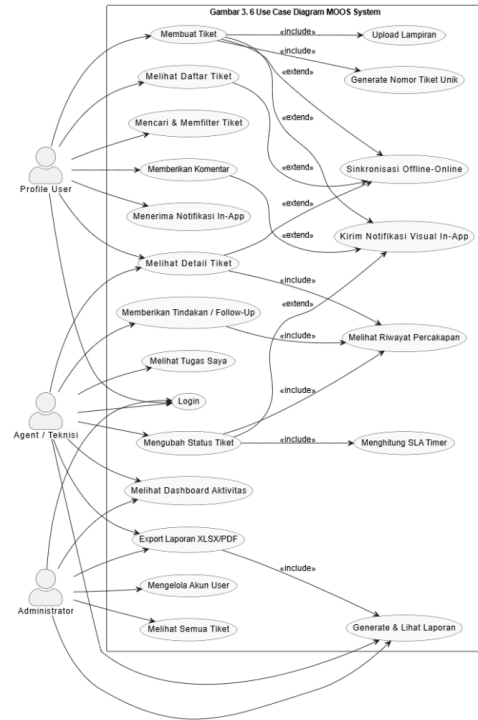
**B. Metode Pengembangan (V-Model)**

Pengembangan sistem menggunakan metode V-Model yang menekankan keterkaitan langsung antara setiap fase perancangan (sisi kiri) dan fase pengujian yang bersesuaian (sisi kanan) [5]. Ilustrasi penerapan V-Model pada penelitian ini ditampilkan pada Gbr. 1.



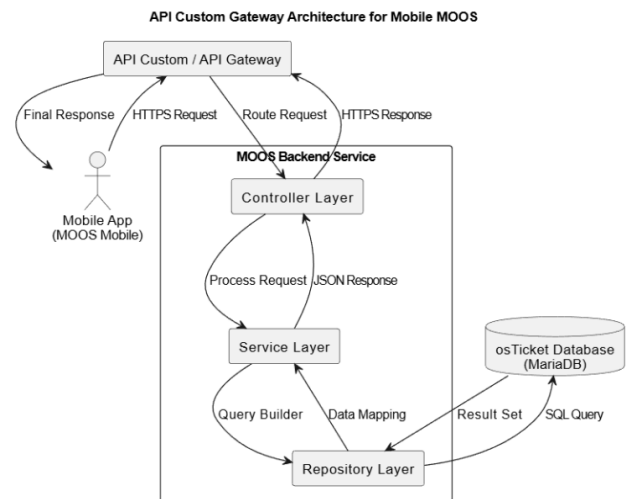
Gbr. 1 Ilustrasi V-Model MOOS Mobile.

1) *Requirement Analysis*: Analisis kebutuhan dilakukan melalui observasi langsung terhadap MOOS Web, wawancara semi-terstruktur dengan Supervisor IT, dan kajian dokumen MOOS Manual Policies & Technical Procedures (2021). Hasilnya: 10 kebutuhan fungsional (F1-F10) dan 7 kebutuhan non-fungsional (NF1-NF7), disertai User Flow dan Use Case Diagram (Gbr. 2) yang menggambarkan interaksi tiga peran pengguna: User/Pelapor, Staff/Teknisi, dan Admin.



Gbr. 2 Use Case Diagram MOOS Mobile.

2) *System Design*: Perancangan mencakup: (a) Entity Relationship Diagram (ERD) berbasis 9 tabel inti (ostTicket, ostTicketCdata, ostThread, ostThreadEntry, ostAttachment, ostFile, ostFileChunk, ostStaff, ostUserAccount) tanpa mengubah struktur tabel asli; dan (b) arsitektur API Gateway sebagai single entry point yang meneruskan permintaan ke MOOS Backend Service melalui tiga lapisan: Controller Layer, Service Layer, dan Database Access Layer (Gbr. 3).



Gbr. 3 API Custom Gateway Architecture for Mobile MOOS.

3) *Architectural Design*: Arsitektur Android mengimplementasikan Clean Architecture + MVVM dengan Repository Pattern menggunakan Hilt sebagai dependency

injection. Lapisan arsitektur terdiri dari UI Layer (Jetpack Compose Material 3), ViewModel Layer (StateFlow + Kotlin Coroutines), dan Repository Layer (safeApiCall → sealed class NetworkResult). Komponen kritis: AuthInterceptor (menyisipkan JWT header dan mendeteksi respons 401 untuk auto-logout) dan AuthEventBus (SharedFlow penghubung Network Layer ke UI Layer tanpa circular dependency). Sesi pengguna disimpan di DataStore Preferences terenkripsi oleh Android Keystore.

4) *Module Design & Implementasi*: Lima modul utama dirinci spesifikasinya: Autentikasi & User, Ticket Management, Thread & Komentar, Attachment Management, dan Notification & SLA. Implementasi menghasilkan 9 layar berbasis role menggunakan Kotlin, Jetpack Compose Material Design 3 (Android), dan Node.js Express (backend API).

C. *Instrumen dan Pengumpulan Data*

Instrumen penelitian terdiri dari tiga komponen. Pertama, Kuesioner SUS: 10 butir pernyataan dengan skala Likert 1–5; item ganjil berupa pernyataan positif dan item genap berupa pernyataan negatif [6]. Kedua, Skenario Tugas: Task Set A (5 tugas) untuk role User/Pelapor dan Task Set B (5 tugas) untuk role Staff/Admin, merepresentasikan alur kerja operasional harian. Ketiga, Lembar Observasi: mencatat tingkat keberhasilan tugas (Mandiri / Dengan Kesulitan / Dengan Bantuan / Gagal) dan pain points secara non-partisipan. Validitas instrumen diverifikasi melalui expert judgment oleh dosen pembimbing dan pilot test kepada 3 responden.

D. *Teknik Analisis Data*

Skor SUS dihitung menggunakan metode standar Brooke [6]: skor kontribusi item ganjil = (nilai jawaban - 1), item genap = (5 - nilai jawaban). Total dikali 2,5 untuk rentang 0–100, sebagaimana Persamaan (1). Interpretasi mengacu pada pedoman Bangor et al. [7]: skor ≥68 = Acceptable; skor >80,3 = Excellent. Analisis mencakup juga statistik deskriptif dan uji Cronbach’s Alpha ( $\alpha \geq 0,70$ ) [9].

$$S = 2.5 \times \left[ \sum_{i \in \text{Ganjil}} (x_i - 1) + \sum_{j \in \text{Genap}} (5 - x_j) \right]$$

III. HASIL DAN PEMBAHASAN

A. *Hasil Pengembangan Aplikasi*

Seluruh 10 kebutuhan fungsional (F1–F10) berhasil diimplementasikan sebagaimana dirangkum pada Tabel I. Satu catatan pada F5: push notification berbasis FCM digantikan oleh indikator visual in-app (banner OVERDUE, badge isanswered, dan OfflineBanner) karena konfigurasi server-side FCM berada di luar cakupan penelitian.

TABEL I REKAPITULASI IMPLEMENTASI KEBUTUHAN FUNGSIONAL

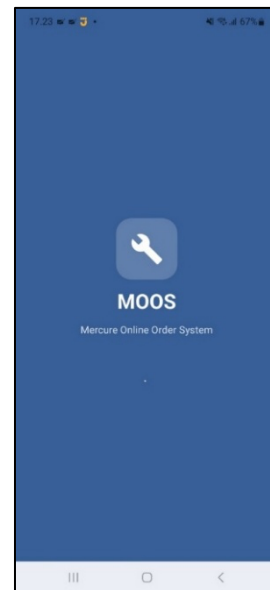
ID	Kebutuhan Fungsional	Prioritas	Status
F1	Autentikasi Multi-Role (JWT)	Tinggi	Selesai

F2	Pembuatan Tiket (Create Ticket)	Tinggi	Selesai
F3	Pembaruan Status Tiket	Tinggi	Selesai
F4	Nomor Tiket Unik	Tinggi	Selesai
F5	Notifikasi & Alert (Visual In-App)	Tinggi	Selesai *
F6	SLA Timer & Penandaan Overdue	Tinggi	Selesai
F7	Pencarian & Filter Tiket	Sedang	Selesai
F8	Ekspor Laporan (XLSX / PDF)	Sedang	Selesai
F9	Riwayat Percakapan & Logging	Tinggi	Selesai
F10	Dashboard Statistik (Role-based)	Sedang	Selesai

\* F5: FCM digantikan indikator visual in-app (di luar cakupan penelitian).

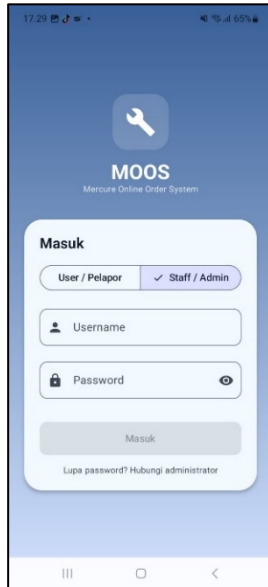
Aplikasi MOOS Mobile terdiri dari 9 layar utama yang diorganisasikan melalui bottom navigation bar berbasis role. Arsitektur API Gateway Node.js berfungsi sebagai middleware di atas database osTicket, memungkinkan sinkronisasi data dua arah secara real-time antara aplikasi mobile dan panel web osTicket. Berikut ini diuraikan masing-masing layar beserta fungsinya.

1) *SplashScreen*: Layar pembuka yang ditampilkan saat aplikasi pertama kali dijalankan. Menampilkan logo MOOS dan nama aplikasi dengan animasi fade-in selama ±1 detik. Secara bersamaan, SplashViewModel memeriksa keberadaan token di DataStore Preferences untuk menentukan arah navigasi: jika sesi aktif tersedia, pengguna langsung diarahkan ke DashboardScreen; jika tidak, diarahkan ke LoginScreen. Layar ini berperan sebagai gate keeper sesi pengguna.



Gbr. 4 SplashScreen MOOS Mobile.

2) *LoginScreen*: Layar autentikasi yang menyediakan dua mode login melalui segmented control: mode User/Pelapor dan mode Staff/Admin. Pemilihan mode menentukan endpoint API yang digunakan, yaitu POST /auth/login/user untuk pelapor dan POST /auth/login/staff untuk staf. Setelah login berhasil, token JWT disimpan ke DataStore Preferences beserta data sesi (role, username, dept\_id) dan pengguna diarahkan ke DashboardScreen. Field password dilengkapi ikon toggle visibilitas untuk kenyamanan penggunaan.



Gbr. 5 LoginScreen dengan segmented control User/Staff.

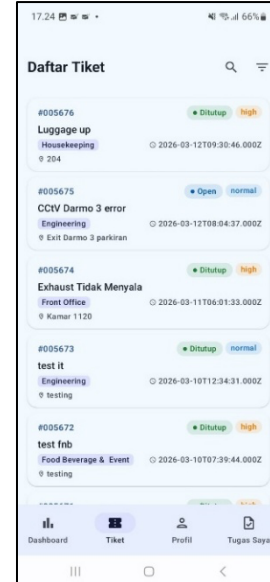
3) *DashboardScreen*: Layar beranda yang menampilkan informasi operasional berbeda-beda sesuai role pengguna. Untuk role User: ringkasan 5 kartu (total, open, closed, overdue, unanswered) dan daftar tiket terbaru. Untuk role Staff: counter cards yang dapat digulir horizontal, LineChart tren 7 hari pembuatan dan penutupan tiket, serta daftar tiket yang relevan dengan departemen. Untuk role Admin: tambahan section byDept dan staffPerformance. Semua data dimuat melalui endpoint GET /dashboard dan dirender secara reaktif menggunakan StateFlow.



Gbr. 6 DashboardScreen (role staff).

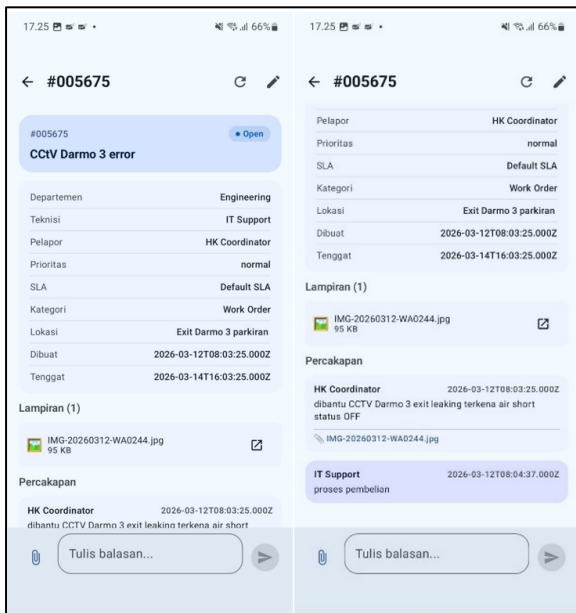
4) *TicketListScreen*: Layar manajemen daftar tiket yang menampilkan seluruh tiket sesuai hak akses role pengguna. Dilengkapi search bar dengan mekanisme smart search: query berupa angka dikirim sebagai ?number= (pencarian

berdasarkan nomor tiket), sedangkan query teks dikirim sebagai ?subject= (LIKE). Filter chips di bawah search bar memungkinkan penyaringan berdasarkan status: All, Open, Closed, Overdue, Perlu Dibalas, dan Tugas Saya. Setiap TicketCard menampilkan nomor, subjek, departemen, prioritas, dan badge overdue merah. Implementasi infinite scroll memuat 20 item per halaman.



Gbr. 7 TicketListScreen dengan filter chips dan smart search.

5) *TicketDetailScreen*: Layar detail tiket yang menyajikan informasi lengkap tiket beserta riwayat percakapan dalam format chat bubble yang intuitif dan mudah dibaca. Bagian atas menampilkan kartu info tiket (pelapor, departemen, lokasi, SLA, prioritas, due date, dan badge OVERDUE merah jika melewati batas waktu yang telah ditentukan). Thread percakapan divisualisasikan berdasarkan type entri: pesan user (type M) tampil di sisi kiri dengan latar putih, dan balasan staff (type R) di sisi kanan dengan latar biru muda. Setiap entri percakapan dilengkapi dengan timestamp dan informasi pengirim untuk memudahkan penelusuran riwayat komunikasi antar pihak. Di bagian bawah tersedia reply box yang berbeda tampilannya untuk role user (POST /message) dan staff (POST /reply dengan opsi ubah status dan tambah lampiran).



Gbr. 8 TicketDetailScreen dengan thread percakapan chat bubble.

6) *CreateTicketScreen*: Layar pembuatan tiket yang hanya dapat diakses oleh role user/pelapor. Menyediakan form dengan field: subjek (wajib), dropdown help topic yang secara otomatis mengisi departemen, SLA, dan prioritas saat dipilih (auto-fill dari GET /lookup/ticket-form), field lokasi, priority chips, deskripsi multi-baris, dan file picker yang mendukung unggahan hingga 5 file (foto/video, maksimum 5 MB per file). Validasi form dilakukan secara real-time sehingga pengguna mendapat umpan balik langsung apabila terdapat field wajib yang belum terisi. Setelah berhasil dikirim, dialog konfirmasi menampilkan nomor tiket unik yang dihasilkan backend dan pengguna secara otomatis diarahkan ke TicketDetailScreen dari tiket yang baru saja dibuat tersebut.



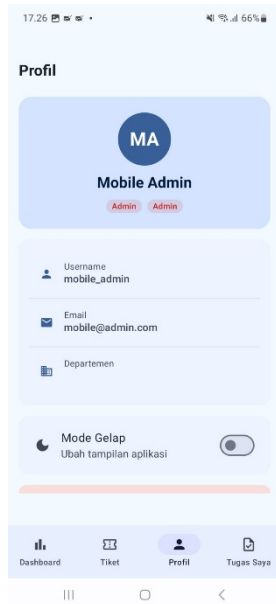
Gbr. 9 CreateTicketScreen (eksklusif role User/Pelapor).

7) *MyTasksScreen*: Layar yang hanya tersedia untuk role staff dan admin, menampilkan daftar tiket yang ditugaskan secara personal atau yang sedang ditangani oleh staf yang bersangkutan. Tiket difilter berdasarkan staff\_id yang sedang login. Layar ini membantu staf teknisi dan housekeeping memantau pekerjaan aktif tanpa perlu menelusuri seluruh daftar tiket. Chip filter Overdue dan Perlu Dibalas tersedia untuk memprioritaskan tiket yang membutuhkan perhatian segera.



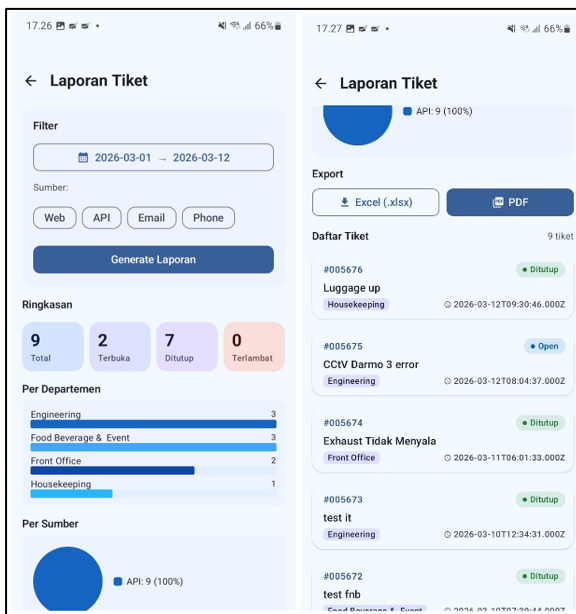
Gbr. 10 MyTasksScreen (role Staff/Admin).

8) *ProfileScreen*: Layar profil pengguna yang menampilkan avatar inisial, nama lengkap, username, email, departemen, dan role badge. Tersedia tombol toggle dark mode untuk mengatur preferensi tampilan yang disimpan secara persisten di DataStore. Tombol Keluar (logout) dilindungi dialog konfirmasi sebelum membersihkan sesi dan mengarahkan pengguna kembali ke LoginScreen dengan pembersihan seluruh back stack navigasi.



Gbr. 11 ProfileScreen dengan toggle dark mode.

9) *ReportScreen*: Layar laporan yang hanya dapat diakses role staff dan admin. Menyediakan filter multi-kriteria: *DateRangePicker* Material 3 untuk rentang tanggal, dropdown departemen dan status, serta multi-select chip untuk sumber tiket (API, Web, Email, Phone). Hasil laporan divisualisasikan melalui dua jenis grafik yang dirender menggunakan *Compose Canvas*: bar chart horizontal (distribusi tiket per departemen) dan pie chart (proporsi tiket berdasarkan sumber). Tombol ekspor memungkinkan unduhan laporan dalam format XLSX (menggunakan *exceljs*) atau PDF (menggunakan *pdfkit*) langsung ke folder Downloads perangkat.



Gbr. 12 ReportScreen dengan visualisasi chart dan ekspor laporan.

B. Hasil Pengujian Sistem

Pengujian sistem dilaksanakan secara bertahap sesuai sisi kanan V-Model. Setiap jenis pengujian memverifikasi artefak yang dihasilkan oleh fase perancangan yang bersesuaian di sisi kiri.

1) *Unit Testing (Verifikasi Module Design)*: Dilaksanakan menggunakan *JUnit 4* dan *Mockk* untuk menguji logika bisnis unit terkecil secara terisolasi dari komponen eksternal. Tabel II menyajikan seluruh 6 skenario yang dieksekusi.

TABEL II HASIL UNIT TESTING

ID	Komponen (Unit)	Skenario Pengujian	Hasil Diharapkan	Status
U T-01	LoginViewMode	Input username kosong, tekan login	UiState.Error("Username tidak boleh kosong"); isSubmitting=false	Pass
U T-02	LoginViewMode	Input valid; API mock return 200+token	UiState: isSuccess=true; token tersimpan ke DataStore mock	Pass
U T-03	TicketRepository	Network offline (mock IOException); panggil getTicketDetail()	Return NetworkResult.Error("Tidak ada koneksi internet")	Pass
U T-04	CreateTicketViewModel	Submit form dengan subject kosong	UiState.error="Subjek tidak boleh kosong"; isSubmitting=false	Pass
U T-05	AuthInterceptor	Response 401 dari server mock	clearSession() dipanggil; AuthEventBus.emitUnauthorized() di-emit	Pass
U T-06	SessionDataStore	saveSession() dipanggil lalu getToken()	Token tersimpan sama dengan token yang dimasukkan	Pass

2) *Integration Testing (Verifikasi Architectural Design)*: Memverifikasi komunikasi dan pertukaran data antar-modul secara end-to-end, mencakup alur dari lapisan Android hingga database. Tabel III menyajikan 6 skenario yang dieksekusi.

TABEL III HASIL INTEGRATION TESTING

ID	Modul Terkait	Skenario Alur Data	Kriteria Sukses	Status
IT-01	Auth ↔ API ↔ DataStore	Login staff berhasil → token diterima → tersimpan ke DataStore → navigasi ke Dashboard	Token tersimpan; Dashboard terbuka tanpa login ulang setelah restart	Pass
IT-02	TicketList ↔ TicketDetail	Tap TicketCard → ticket ID diteruskan → TicketDetailScreen memuat data sesuai	TicketDetail menampilkan subject, status, thread yang sesuai ID	Pass
IT-03	CreateTicket ↔ Upload ↔ API	Isi form → pilih 2 foto → submit → multipart ke POST /tickets	Response {id, number}; lampiran terlihat di TicketDetail dan panel web	Pass

IT-04	Thread ↔ Reply ↔ API	Staff kirim reply (type R) via POST /:id/reply → fetchDetail() dipanggil ulang	Balasan baru muncul di chat bubble tanpa pull-to-refresh manual	Pass
IT-05	Lookup ↔ CreateTicket	CreateTicketScreen dibuka → GET /lookup/ticket-form → data mengisi dropdown	Dropdown tidak kosong; saat topic dipilih, dept dan prioritas terisi otomatis	Pass
IT-06	Dashboard ↔ API (role-based)	Login sebagai admin → GET /dashboard → respons memuat byDept & staffPerformance	Section staffPerformance hanya tampil untuk admin; tidak tampil untuk user biasa	Pass

3) System Testing / Pengujian Fungsional (Verifikasi Requirement Analysis): Pengujian kotak hitam (black-box) yang memvalidasi seluruh sistem dari perspektif pengguna akhir terhadap seluruh kebutuhan fungsional dan non-fungsional. Tabel IV menyajikan 10 skenario yang dieksekusi.

TABEL IV HASIL SYSTEM TESTING (BLACK-BOX)

ID	Fitur (Requirement)	Prosedur Singkat	Hasil Aktual	Status
ST-01	Autentikasi Multi-Role (F1)	Masukkan username & password salah → tekan Masuk	SnackBar merah: "Token tidak valid"; tetap di LoginScreen	Pass
ST-02	Pembuatan Tiket (F2)	Login user → Buat Tiket → isi form + 1 foto → Kirim	Dialog nomor tiket unik; tiket muncul di TicketList berstatus Open	Pass
ST-03	Filter Tiket (F7)	Di TicketList tekan chip filter "Overdue"	Hanya tiket dengan label OVERDUE merah yang ditampilkan	Pass
ST-04	Smart Search (F7)	Ketik angka "748291" di search bar	Sistem kirim ?number=748291 ; tiket bernomor tersebut muncul pertama	Pass
ST-05	Reply & Update Status (F3)	Login staff → buka tiket Open → pilih Response+Close d → Kirim	Balasan type R di thread; status tiket berubah jadi Closed	Pass
ST-06	Dashboard Role-based (F10)	Login sebagai user → cek tab Dashboard	Hanya tampil recent tickets; tanpa counter scrollable dan tren chart	Pass
ST-07	Riwayat Percakapan (F9)	Login user → buka tiket dengan balasan staff → cek thread	Reply staff (type R) tampil; internal note (type N) tidak terlihat user	Pass
ST-08	Ekspor Laporan (F8)	Login admin → Report → set filter tanggal → Export PDF	File .pdf terunduh ke folder Downloads perangkat Android	Pass

ST-09	Auto-Logout (NF2)	Paksa token expired → lakukan request	SnackBar "Sesi berakhir" → navigasi otomatis ke LoginScreen; back stack bersih	Pass
ST-10	Offline Banner (NF1)	Matikan koneksi internet → navigasi ke TicketList	Banner kuning "Tidak ada koneksi internet" tampil di atas layar	Pass

Seluruh 22 skenario pengujian (6 unit + 6 integrasi + 10 fungsional) menghasilkan status Pass tanpa pengecualian. Tidak terdapat satu pun skenario yang menghasilkan status Fail. Keberhasilan seluruh tahap pengujian teknis ini menjadi prasyarat objektif agar evaluasi usability pada tahap Acceptance Testing dapat dilakukan tanpa bias akibat cacat teknis sistem.

### C. Hasil Evaluasi Usability (SUS)

Sesi usability testing dilaksanakan secara langsung di Hotel Mercure Surabaya Grand Mirama dan melibatkan 21 responden staf internal dari beragam departemen: Front Office (8 orang), Engineering/Teknisi (3 orang), IT/Sistem (3 orang), Food & Beverage (2 orang), Housekeeping (1 orang), dan Lainnya (3 orang). Sebaran role: 9 responden mengerjakan Task Set A (User/Pelapor) dan 12 responden mengerjakan Task Set B (Staff/Admin).

Overall task success rate mencapai 100% (105/105 percobaan): 95 mandiri (90,48%), 8 dengan sedikit kesulitan, dan 2 dengan bantuan peneliti. Tidak terdapat tugas yang gagal. Pain point utama: (1) komponen file-picker kurang menonjol saat membuat tiket (3/9 responden User, 33%); (2) satu responden Staff memerlukan bantuan memilih mode login pada segmented control; dan (3) mayoritas responden memerlukan 1-2 menit eksplorasi awal sebelum familiar dengan pola navigasi bottom bar berbasis role.

TABEL V SKOR SUS PER RESPONDEN

ID	Departemen	Role	Skor SUS
R-01	Front Office	Staff	90,0
R-02	Front Office	User	97,5
R-03	Front Office	Staff	82,5
R-04	Front Office	User	95,0
R-05	IT / Sistem	Admin	100,0
R-06	IT / Sistem	Staff	47,5
R-07	Engineering	User	82,5
R-08	Front Office	User	95,0
R-09	Engineering	Staff	60,0
R-10	Lainnya	Staff	65,0
R-11	Front Office	User	90,0
R-12	IT / Sistem	User	80,0
R-13	Front Office	Staff	100,0
R-14	Housekeeping	Staff	100,0
R-15	Food & Beverage	Staff	95,0
R-16	Front Office	Staff	95,0
R-17	Engineering	User	92,5
R-18	Food & Beverage	Staff	87,5
R-19	Lainnya	User	90,0
R-20	Lainnya	User	87,5
R-21	Front Office	Staff	82,5

Rata-rata	—	—	86,43
-----------	---	---	-------

Tabel VI menampilkan interpretasi skor SUS berdasarkan pedoman Bangor et al. [7].

TABEL VI INTERPRETASI SKOR SUS (BANGOR ET AL., 2008)

Rentang Skor	Grade	Adjective Rating	Acceptability
> 80,3	A	Excellent	Acceptable
68 – 80,3	B	Good	Acceptable
51 – 67,9	C	OK / Fair	Marginal High
< 51	F	Poor	Not Acceptable
<b>MOOS Mobile: 86,43</b>	<b>A</b>	<b>Excellent</b>	<b>Acceptable</b>

TABEL VII STATISTIK DESKRIPTIF SKOR SUS

Parameter Statistik	Nilai
Jumlah Responden (n)	21
Skor Minimum	47,5
Skor Maksimum	100,0
Mean (Rata-rata)	86,43
Median	90,0
Standar Deviasi	13,80
Overall Task Success Rate	100% (105/105 percobaan)
Cronbach's Alpha ( $\alpha$ )	0,835
Interpretasi Grade	A – Excellent (Acceptable)

Nilai Cronbach's Alpha 0,835 melampaui batas minimum 0,70 [9], sehingga instrumen SUS dinyatakan reliabel. Median 90,0 yang lebih tinggi dari mean mengkonfirmasi distribusi skor yang condong ke nilai tinggi.

#### D. Pembahasan

Skor SUS rata-rata 86,43 melampaui ambang batas Acceptable ( $\geq 68$ ) dan mencapai kategori Excellent ( $> 80,3$ ) berdasarkan pedoman Bangor et al. [7]. Hasil ini membuktikan diterimanya Hipotesis  $H_2$  bahwa aplikasi MOOS Mobile memiliki usability tinggi (SUS  $\geq 68$ ). Jika outlier R-06 (skor 47,5) dikecualikan, rata-rata 20 responden lainnya mencapai 88,38, mengukuhkan penerimaan yang kuat dari mayoritas pengguna.

Dibandingkan penelitian terdahulu berbasis Jetpack Compose dan SUS pada domain sejenis, skor MOOS Mobile berada di atas: Putra & Tolle [10] (kategori Acceptable, skor  $< 80$ ), Ramadani & Khoirudin [11] (SUS 80), dan penelitian marketplace UMKM serupa (SUS 75). Keunggulan ini dikaitkan dengan tiga faktor: (1) konsistensi Material Design 3 pada seluruh 9 layar; (2) role-based navigation yang mengurangi cognitive load; dan (3) keberhasilan seluruh 22 skenario pengujian teknis sebelum evaluasi, yang meminimalkan gangguan teknis selama sesi [12].

Outlier R-06 berhasil menyelesaikan seluruh tugas meski memberikan skor subjektif 47,5, mengindikasikan bahwa persepsi subjektif terhadap kompleksitas tidak selalu mencerminkan kemampuan fungsional aktual pengguna [12]. Pain point utama (file-picker kurang menonjol dan eksplorasi awal navigasi) dapat ditindaklanjuti melalui peningkatan visual affordance pada komponen form di versi pengembangan berikutnya.

Keberhasilan keseluruhan proses pengembangan dengan V-Model berkorelasi positif dengan skor usability yang tinggi.

Setiap fase perancangan diverifikasi oleh pasangan pengujian yang bersesuaian, sehingga ketika pengguna mengevaluasi aplikasi, sistem telah bebas dari cacat teknis. Hal ini mendukung argumen bahwa kualitas proses pengembangan mempengaruhi persepsi usability pengguna akhir.

#### IV. KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan aplikasi MOOS Mobile berbasis Jetpack Compose menggunakan metode V-Model dengan arsitektur MVVM + Repository Pattern dan Node.js Express sebagai middleware API di atas database osTicket yang sudah ada. Seluruh 10 kebutuhan fungsional berhasil diimplementasikan dalam 9 layar aplikasi berbasis role. Semua skenario pengujian (6 unit testing, 6 integration testing, 10 system testing) menghasilkan status Pass. Evaluasi usability terhadap 21 responden menghasilkan skor SUS rata-rata 86,43 (Excellent, Acceptable) dengan overall task success rate 100% dan Cronbach's Alpha 0,835. Aplikasi MOOS Mobile dinyatakan layak diimplementasikan sebagai alat kerja operasional staf hotel.

Penelitian selanjutnya disarankan: (1) mengimplementasikan push notification berbasis FCM; (2) mengembangkan persistent offline caching dengan Room Database; (3) memperluas jumlah responden dari beragam hotel; serta (4) menggabungkan SUS dengan metode evaluasi komplementer seperti NASA-TLX atau Think Aloud Protocol.

#### UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Allah Subhanahu wa Ta'ala atas segala limpahan rahmat, kemudahan, dan kelancaran yang diberikan selama pelaksanaan penelitian hingga penulisan artikel ini. Terima kasih yang tak terhingga penulis sampaikan kepada keluarga tercinta atas doa yang tidak pernah putus, dukungan moral, dan semangat yang menjadi pondasi kekuatan selama proses ini berlangsung. Kepada sahabat-sahabat dan rekan-rekan Program Studi S1 Teknik Informatika yang senantiasa memberikan bantuan, masukan, dan motivasi, penulis mengucapkan terima kasih yang sebesar-besarnya.

Penulis juga mengucapkan terima kasih kepada Bapak Ronggo Alit, S.Kom., M.M., M.T. selaku dosen pembimbing atas arahan, bimbingan, dan kesabaran dalam membimbing penulis selama proses penelitian; kepada Ibu Paramitha Nerisafitra, S.ST., M.Kom. dan Bapak Aditya Prapanca, S.T., M.Kom. selaku dosen penguji atas masukan dan saran yang membangun; serta kepada seluruh dosen dan civitas akademika Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya atas dukungan akademis yang diberikan. Terima kasih kepada manajemen Hotel Mercure Surabaya Grand Mirama atas kesempatan pelaksanaan penelitian dan kepada seluruh staf internal yang telah berkenan meluangkan waktu sebagai responden penelitian.

#### REFERENSI

- [1] M. Nikopoulou, P. Kourouthanassis, G. Chasapi, A. Pateli, and N. Mylonas, "Determinants of Digital Transformation in the Hospitality Industry: Technological, Organizational, and

- Environmental Drivers,” *Sustainability (Switzerland)*, vol. 15, no. 3, p. 2736, 2023, doi: 10.3390/su15032736.
- [2] B. Chourasia, “The future of smart hotels and digital transformation in the hospitality industry,” in *Transforming the Service Sector With New Technology*, IGI Global, 2025, pp. 533–540. doi: 10.4018/979-8-3693-7447-4.ch032.
- [3] S. Ratna, S. Saide, A. M. Putri, R. E. Indrajit, and D. Muwardi, “Digital transformation in tourism and hospitality industry: a literature review of blockchain, financial technology, and knowledge management,” *EuroMed Journal of Business*, vol. 19, no. 1, pp. 84–112, 2024, doi: 10.1108/EMJB-04-2023-0118.
- [4] L. Wahlandt and A. Brännholm, “A Comparative Analysis of Jetpack Compose and XML Views,” in *Bachelor Degree Project in Information Technology*, 2024, p. 10.
- [5] S. Balaji and M. S. Murugaiyan, “WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC,” *International Journal of Information Technology and Business Management*, vol. 2, no. 1, pp. 26–30, 2012, [Online]. Available: <https://mediaweb.saintleo.edu/Courses/COM430/M2Readings/WATEERFALLVs V-MODEL Vs AGILE A COMPARATIVE STUDY ON SDLC.pdf>
- [6] J. Brooke, “SUS: A ‘quick and dirty’ usability scale,” in *Usability Evaluation in Industry*, CRC Press, 1996, pp. 189–194.
- [7] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008, doi: 10.1080/10447310802205776.
- [8] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *Int. J. Hum. Comput. Interact.*, vol. 34, no. 7, pp. 577–590, 2018, doi: 10.1080/10447318.2018.1455307.
- [9] M. Tavakol and R. Dennick, “Making sense of Cronbach’s alpha,” *Int. J. Med. Educ.*, vol. 2, pp. 53–55, 2011, doi: 10.5116/ijme.4dfb.8dfd.
- [10] I. P. R. P. Putra and H. Tolle, “Pengembangan Aplikasi Pembelajaran Bahasa Bali berbasis Android menggunakan MVVM Architecture dan Jetpack Compose,” vol. 7, no. 5, pp. 2205–2214, 2023, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12687>
- [11] Y. A. Ramadani and K. Khoirudin, “Implementasi Jetpack Compose Dan Firebase Authentication Dalam Pengembangan Aplikasi Museum Berbasis Android Di Yogyakarta,” *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 3, pp. 2118–2133, 2025, doi: 10.29100/jipi.v10i3.6214.
- [12] P. Weichbroth, “Usability Testing of Mobile Applications: A Methodological Framework,” *Applied Sciences (Switzerland)*, vol. 14, no. 5, 2024, doi: 10.3390/app14051792.