Perancangan Antarmuka Dinamis Menggunakan Jetpack Compose untuk Aplikasi Operasional Key's Laundry

Amelia Putri Aniyanto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ameliapa@student.telkomuniversity.ac.id

Roswan Latuconsina
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
roswan@telkomuniversity.ac.id

Astri Novianty
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
astrinov@telkomuniversity.ac.id

***Abstrak* — Aplikasi operasional Key's Laundry membutuhkan antarmuka yang efisien, adaptif, dan mudah digunakan untuk mendukung proses bisnis seperti pencatatan layanan transaksi, pengelolaan data pelanggan, dan manajemen karyawan seperti data rekap gaji dan absensi. Penelitian ini merancang antarmuka dinamis menggunakan Jetpack Compose pada aplikasi manajemen laundry berbasis Android. Jetpack Compose dipilih karena mendukung pembuatan UI deklaratif yang modular, efisien, dan responsif terhadap perubahan data. Aplikasi dikembangkan untuk tiga peran pengguna, yaitu admin, karyawan, dan pelanggan. Fitur utama yang disediakan meliputi pencatatan layanan, pengelolaan data pelanggan, rekap absensi dan gaji karyawan, laporan keuangan, serta dukungan pengiriman nota digital melalui WhatsApp API dan pencetakan nota menggunakan printer *bluetooth*. Proses pengembangan antarmuka mengikuti prinsip modularitas dan pemisahan logika tampilan, sehingga mempermudah pemeliharaan dan pengembangan lebih lanjut. Hasil pengujian menggunakan metode *user acceptance testing* (UAT) menunjukkan bahwa antarmuka yang dirancang mampu mempercepat proses transaksi, mengurangi kesalahan pencatatan, serta meningkatkan efisiensi operasional dalam kegiatan usaha Key's Laundry.**

***Kata kunci*— Jetpack Compose, UI Dinamis, Aplikasi Android, Manajemen Laundry, WhatsApp API, Efisiensi Operasional.**

I. PENDAHULUAN

Perkembangan teknologi digital sudah mendorong perusahaan dengan mengandalkan produk digital untuk membantu usaha mereka salah satu contohnya adalah penggunaan aplikasi *mobile* dalam usaha laundry. Penggunaan aplikasi membantu manajemen usaha mereka dalam beroperasi sehari-hari, berbagai macam jenis usaha laundry memiliki aplikasi sesuai jenis dengan kebutuhan mereka.

Perkembangan teknologi android telah memiliki pendekatan baru dalam membangun antarmuka yang memiliki *framework* deklaratif sehingga dapat memungkinkan membangun UI *(User Interface)* lebih cepat, adaptif, dan mudah di-maintain yaitu bernama Jetpack Compose. Jetpack Compose bekerja dengan cara mendukung pengembangan antarmuka pengguna yang responsif terhadap perubahan data dan cocok diterapkan pada sistem dengan alur transaksi yang dinamis seperti operasional laundry [1].

Hal ini diteliti untuk merancang antarmuka dinamis pada aplikasi operasional Key's Laundry menggunakan Jetpack Compose. Tujuannya ialah membangun tampilan antarmuka yang responsi, efesien, serta mampu mendukung peran admin, karyawan, dan pelanggan secara terintegrasi. Aplikasi ini dirancang dengan fitur-fitur utama seperti pembuatan sampai pencatatan pesanan, rekapitulasi laporan keuangan, pengiriman nota elektronik melalui integrasi *backend* ke WhatsApp serta dicetak menggunakan printer *bluetooth*, dan manajemen karyawan. Solusi ini diharapkan mampu meningkatkan efisiensi operasional dan kualitas layanan pada usaha Key's Laundry.

II. KAJIAN TEORI

Pada pengembangan aplikasi mobile Key's Laundry ini menggunakan beberapa tools seperti Figma, Android Studio, Kotlin, dan Jetpack Compose tersendiri.

a) Figma

Figma adalah tools desain berbasis cloud yang digunakan untuk membuat prototype antarmuka aplikasi. Figma menjadi langkah pengambilan tools pertama yang dilakukan oleh developer untuk merancang tampilan kasar UI (*User Interface*).

b) Android Studio

Android Studio merupakan IDE resmi yang digunakan dalam membangun aplikasi *mobile* berbasis android, Android Studio menyediakan *code editor* untuk menuliskan program, *debugger* untuk mencari *error code*, dan *emulator* untuk menjalankan aplikasi.

c) Kotlin

Kotlin adalah bahasa pemograman modern yang dikembangkan oleh JetBrains dan didukung oleh Google untuk pengembangan Android. Pemilihan bahasa kotlin karena memiliki *sintaks* yang ringkas dan aman dalam membangun fitur yang kompleks.

d) Jetpack Compose

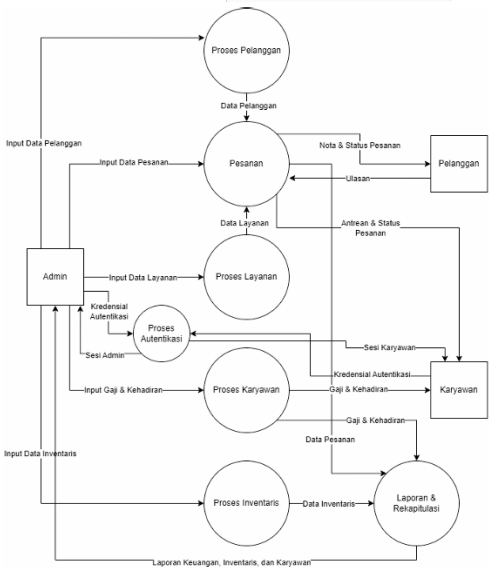
Jetpack compose ialah framework UI deklaratif terkini yang diciptakan oleh Google untuk menciptakan antarmuka Android dengan cara yang efisien, dan mendukung pembuatan tampilan yang dinamis serta responsif terhadap perubahan data.

III. METODE

Dalam pengembangan ini menerapkan beberapa tahapan yang dimulai dari mengidentifikasi kebutuhan sistem, merancang tampilan aplikasi, membangun aplikasi menggunakan teknologi Jetpack Compose, hingga melakukan validasi melalui pengujian penerimaan pengguna atau UAT. Selain itu, untuk memastikan pengembangan berjalan sistematis, kami juga memanfaatkan berbagai diagram dan model perancangan. DFD digunakan untuk memetakan bagaimana data mengalir dalam sistem, sementara ERD membantu untuk mendesain struktur database yang tepat. *Diagram Use Case* berperan dalam memahami bagaimana setiap pengguna berinteraksi dengan sistem, dan perancangan UI memastikan setiap jenis pengguna mendapat tampilan yang sesuai dengan kebutuhannya.

a. Data Flow Diagram

Data Flow Diagram (DFD) digunakan untuk membuat ilustrasi alur transaksi data pada aplikasi operasional Key's Laundry [2]. Berdasarkan gambar di bawah terdapat lima proses utama, yaitu proses pelanggan, proses layanan, proses karyawan, proses inventaris, dan pesanan. Semua terhubungn dengan Admin, Pegawai, dan Pelanggan.



GAMBAR 1
Data Flow Diagram

Proses diawali dari Admin yang memiliki akses penuh dalam menginput data pelanggan, layanan laundry, gaji, kehadiran pegawai, dan data rekap keuangan. Admin juga memiliki akses autentikasi untuk masuk ke sistem melalui proses autentikasi kode masuk Admin dan Kode OTP.

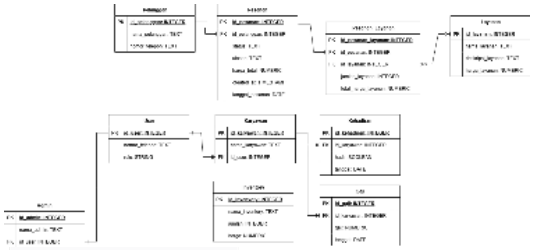
Pegawai juga melalui proses autentikasi Kode OTP sebelum dapat mengakses data pesana dan menginput kehadiran. Gaji, data rekap absensi, serta antrean data order laundry dikelola oleh admin dan menjadi bagian dari proses rekapitulasi laporan.

Pelanggan berinterkasi langsung dengan sistem melalui proses pemesanan layanan. Pelanggan dapat melihat nota, status pesanan, dan memberikan ulasan atas kualitas layanan laundry. Pesanan yang masuk diproses berdasarkan data layanan yang tersedia, kemudian dihubungkan dengan antrean layanan yang akan dikerjakan oleh karyawan.

Semua proses menghasilkan keluaran yang mengalir ke laporan dan rekapitulasi, termasuk laporan keuangan, data inventaris serta rekap data karyawan.

b. Entity Relationship Diagram

ERD atau *Entity Relationship Diagram* ini dibuat untuk merancang *database* aplikasi Key's Laundry. Intinya, diagram ini menggambarkan semua data penting yang ada di sistem dan bagaimana data-data tersebut saling terhubung, sehingga informasi bisa tersimpan dengan rapi dan mudah diakses [3].



GAMBAR 2
Entity Relationship Diagram

Di sistem ini ada sebelas jenis data utama, yaitu Pelanggan, Pesanan, Pesanan_Layanan, Layanan, *User*, Pegawai, Kehadiran, Gaji, Inventory, Admin, dan Akun. Data Pelanggan berisi informasi dasar tentang *customer* dan langsung terkait dengan Pesanan, yang mencatat detail transaksi seperti tanggal order, total biaya, dan status cuciannya.

Yang menarik adalah Pesanan_Layanan yang berperan sebagai penghubung antara Pesanan dan Layanan. Ini diperlukan karena dalam satu pesanan bisa ada beberapa jenis layanan sekaligus, misalnya cuci kering plus setrika. Sementara Layanan sendiri menyimpan info nama dan harga setiap layanan.

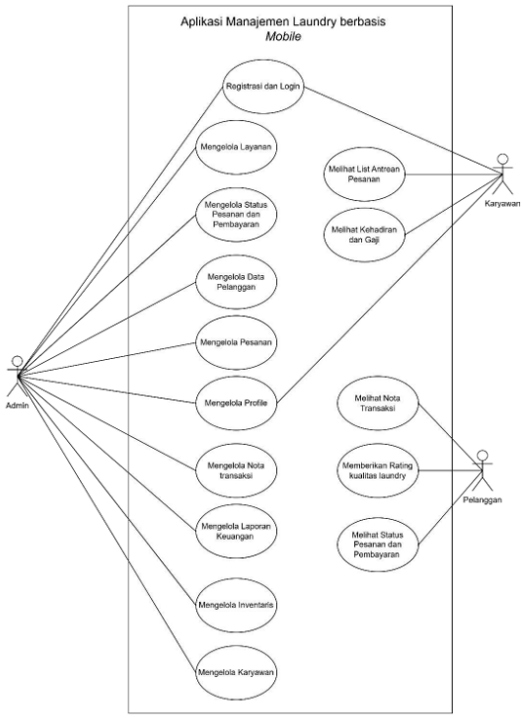
Untuk data karyawan internal, ada *user* yang mengatur sistem login dan terhubung dengan Karyawan dan Admin. Data Karyawan berisi informasi personal staff dan terkait dengan Kehadiran serta Gaji, yang mencatat absensi dan perhitungan gaji per periode.

Sistem juga mencatat *Inventory* untuk barang-barang operasional seperti deterjen, plastik

pembungkus, dan perlengkapan lainnya. Setiap jenis data punya kode unik (*primary key*) untuk menjaga data tetap akurat, plus kode penghubung (*foreign key*) agar semua data bisa saling terkait dan berguna untuk operasional sehari-hari maupun pembuatan laporan.

c. Use Case Diagram

Use Case Diagram ini menunjukkan bagaimana berbagai pengguna berinteraksi dengan aplikasi manajemen Key's Laundry [4]. Ada tiga jenis pengguna utama yang menggunakan sistem ini, yaitu Admin, Pegawai, dan Pelanggan. Masing-masing punya akses yang berbeda sesuai dengan peran mereka.



GAMBAR 3
Use Case Diagram untuk Semua *Role*

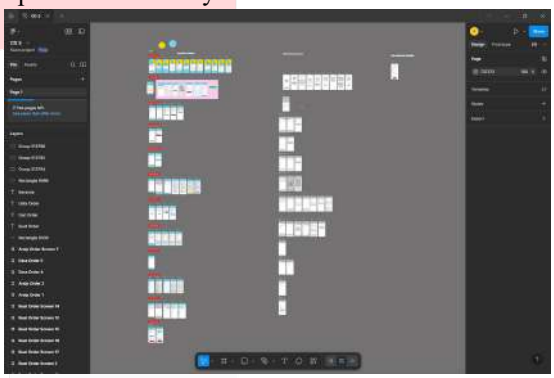
Admin adalah pengguna yang punya kontrol penuh terhadap sistem. Mereka bisa melakukan hampir semua hal, mulai dari daftar dan masuk ke sistem, mengatur jenis layanan yang tersedia, mengelola *database* customer, mengatur orderan, *update* profil bisnis, membuat nota pembayaran, menyusun laporan keuangan, mengecek stok barang, sampai mengurus data karyawan. Admin juga bisa memantau kondisi pembayaran dan histori transaksi dari semua pelanggan.

Karyawan aksesnya lebih terbatas, fokus pada hal-hal yang berkaitan dengan pekerjaan mereka sehari-hari. Mereka bisa melihat daftar cucian yang harus dikerjakan dan mengecek data absensi plus gaji mereka sendiri.

Pelanggan punya akses yang memungkinkan mereka tetap terhubung dengan layanan. Mereka bisa melihat struk transaksi, kasih *rating* untuk kualitas layanan, dan cek status cucian serta pembayaran. Dengan fitur ini, pelanggan bisa memantau proses laundry mereka secara langsung dan memberikan masukan tentang pelayanan yang mereka terima.

d. User Interface

Untuk membuat tampilan aplikasinya, kami menggunakan Figma sebagai alat utama dalam mendesain *prototype* awal. Tampilan ini dibuat dengan konsep yang simpel, jelas, dan *user-friendly* untuk tiga jenis pengguna utama: pemilik bisnis (admin), staff, dan customer. Intinya, kami ingin menciptakan aplikasi yang mudah dipahami dan digunakan, sehingga bisa membantu kelancaran operasional laundry sehari-hari.



GAMBAR 4
User Interface

Di Gambar 1.4, bisa dilihat keseluruhan rancangan UI yang dibuat di Figma. Ada berbagai halaman seperti login, beranda, menu layanan, form pemesanan, laporan keuangan, dan halaman untuk cek nota transaksi. Desainnya dibuat modular supaya lebih mudah saat mengaplikasikannya dengan Jetpack Compose.

GAMBAR 5
Tampilan Akhir Aplikasi Key's Laundry

Gambar 1.5 memperlihatkan contoh hasil akhir tampilan aplikasi Key's Laundry, yaitu halaman pembuka (*splash screen*) dan halaman login yang menampilkan pilihan peran pengguna.

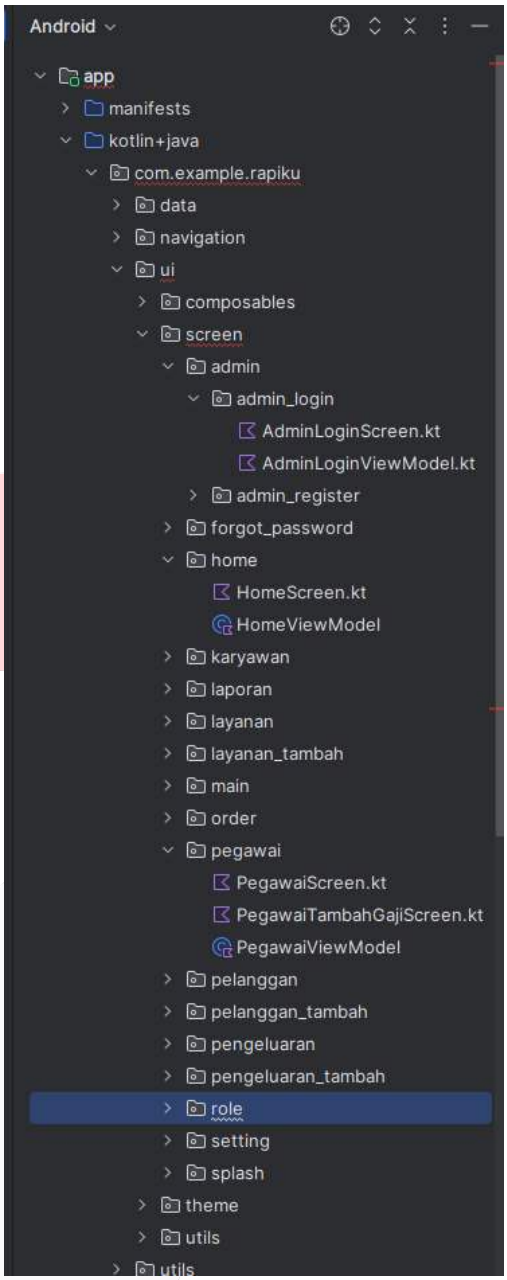
Semua elemen tampilan ini kemudian dibangun menggunakan Jetpack Compose, yang memungkinkan pembuatan UI secara deklaratif dan responsif. Kami memilih Jetpack Compose karena kemampuannya untuk membuat tampilan yang dinamis dan bisa menyesuaikan konten berdasarkan data serta peran pengguna secara *real-time*. Dengan cara ini, pengembangan dan *maintenance* aplikasi jadi lebih efisien, plus tampilan aplikasi bisa tetap konsisten di semua bagian.

IV. HASIL DAN PEMBAHASAN

Bagian ini menampilkan hasil akhir aplikasi Key's Laundry yang dibangun dengan Jetpack Compose, mulai dari tampilan final, fitur-fitur utama, hingga cara penggunaan untuk setiap peran pengguna. Aplikasi juga sudah diuji menggunakan *black box testing* untuk cek fungsionalitas, *white box testing* untuk verifikasi logika program, dan *user acceptance testing* (UAT) untuk memastikan aplikasi nyaman dan efektif digunakan oleh pengguna.

a. Implementasi

Berikut berisi daftar folder yang berisikan file code jetpack compose dari masing-masing *screen page* admin hingga pegawai. Pengorganisasian proyek dibuat berdasarkan fitur dan jenis pengguna, seperti yang terlihat di folder UI>Screen. Di dalamnya ada berbagai *subfolder* seperti admin, pegawai, layanan, laporan, order, dan lain-lain.



GAMBAR 6
Struktur File UI *Code* Jetpack Compose Aplikasi Key's Laundry

Setiap folder berisi file-file Jetpack Compose untuk halaman spesifik, seperti AdminLoginScreen.kt, HomeScreen.kt, PegawaiScreen.kt, dan sebagainya. Dalam setiap file, kami menggunakan komponen-komponen seperti Scaffold, Column, LazyColumn, Text, dan Button untuk menyusun elemen UI.

GAMBAR 7
Implementasi Jetpack Compose pada *Home Screen* Admin

Sebagai contoh, di file HomeScreen.kt, tata letak utama menggunakan Scaffold yang dilengkapi dengan TopAppBar di bagian atas dan konten utama di dalam Column. Untuk menampilkan daftar data seperti layanan atau pesanan, kami menggunakan LazyColumn yang lebih efisien dalam hal performa. Yang menarik dari Jetpack Compose adalah sifatnya yang reaktif terhadap perubahan data, jadi ketika data berubah, tampilan otomatis ikut berubah tanpa perlu *coding* tambahan seperti di sistem lama.

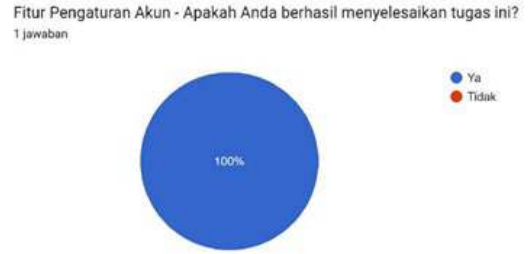
Setiap tampilan juga dihubungkan dengan ViewModel tersendiri (seperti HomeViewModel.kt) untuk memisahkan pengelolaan data dari tampilan UI, mendukung prinsip *clean architecture*. Dengan pendekatan ini, pengembangan aplikasi jadi lebih simpel, kode lebih ringkas, dan *maintenance* ke depannya juga lebih mudah.

b. *User Acceptance Testing* (UAT)

Pengujian UAT pada admin dan pegawai dilakukan dengan cara pengujian langsung dengan aplikasi dengan beberapa *test case* terhadap fitur-fitur sesuai *role* pengguna [5]. Pengujian dilakukan menggunakan 3 perangkat android admin dan karyawan dengan OS yang berbeda yaitu android 11,12 dan 13. Untuk mendapatkan transparansi masukan dan saran oleh admin dan karyawan terhadap kemudahan aplikasi diajukan beberapa pertanyaan terbuka melalui kuisioner agar admin dan pegawai dapat menjelaskan apa saja kendala dan masukan untuk pengembangan selanjutnya. Berikut hasil pengujian langsung untuk pengujian UAT admin dan pegawai:

GAMBAR 8
Hasil UAT Fitur Login dan Registrasi Akun

Berdasarkan hasil pengujian UAT terhadap fitur login dan registrasi akun, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.

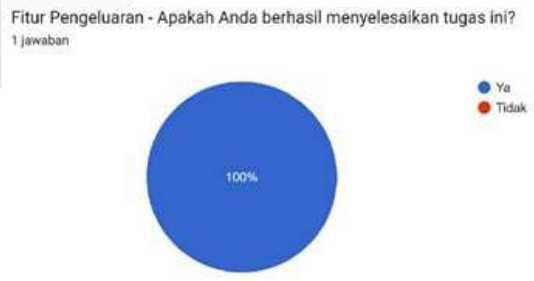


GAMBAR 9
Hasil UAT Fitur Pengaturan Akun

Berdasarkan hasil pengujian UAT terhadap fitur pengaturan akun, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.

GAMBAR 10
Hasil UAT Fitur Layanan dan Pelanggan

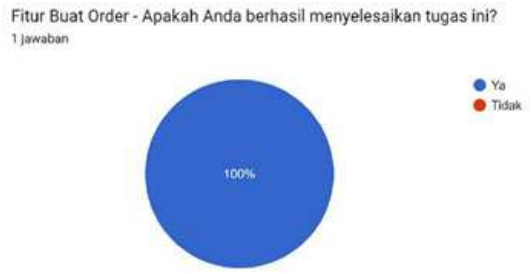
Berdasarkan hasil pengujian UAT terhadap fitur layanan dan pelanggan, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



GAMBAR 11
Hasil UAT Fitur Pengeluaran

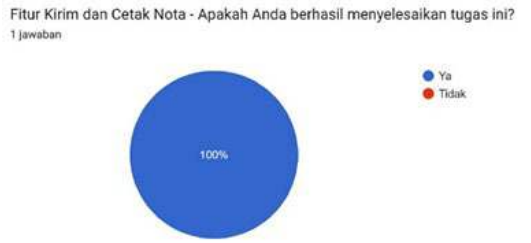
Berdasarkan hasil pengujian UAT terhadap fitur pengeluaran, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah

berfungsi dengan baik dan mudah digunakan oleh pengguna.



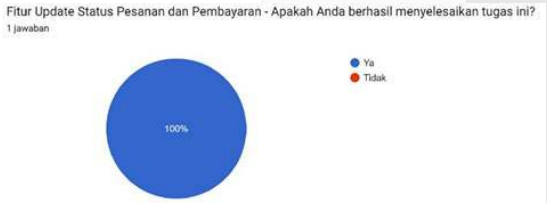
GAMBAR 12
Hasil UAT Fitur Buat Order

Berdasarkan hasil pengujian UAT terhadap fitur buat order, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



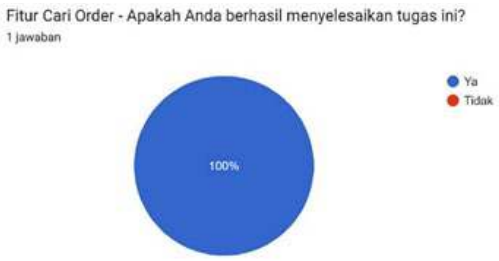
GAMBAR 13
Hasil UAT Fitur Kirim dan Cetak Nota

Berdasarkan hasil pengujian UAT terhadap fitur kirim dan cetak nota, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



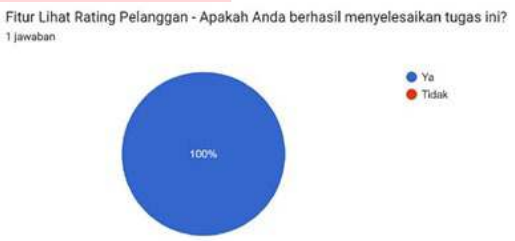
GAMBAR 14
Hasil UAT Fitur Update Status Pesanan dan Pembayaran

Berdasarkan hasil pengujian UAT terhadap fitur update status pesanan dan pembayaran, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



GAMBAR 15
Hasil UAT Fitur Cari Order

Berdasarkan hasil pengujian UAT terhadap fitur cari order, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



GAMBAR 16
Hasil UAT Fitur Lihat Rating Pelanggan

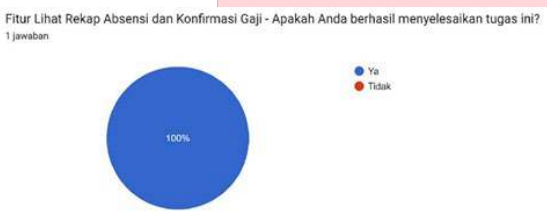
Berdasarkan hasil pengujian UAT terhadap fitur lihat rating pelanggan, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.

GAMBAR 17
Hasil UAT Fitur Lihat dan Export Laporan Keuangan

Berdasarkan hasil pengujian UAT terhadap fitur lihat dan export laporan keuangan, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.

GAMBAR 18
Hasil UAT Fitur Lihat Antrean Laundry

Berdasarkan hasil pengujian UAT terhadap fitur lihat antrean laundry, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.



GAMBAR 19
Hasil UAT Fitur Lihat Rekap Absensi dan Konfirmasi Gaji

Berdasarkan hasil pengujian UAT terhadap fitur lihat rekap absensi dan konfirmasi gaji, responden (100%) menyatakan berhasil menyelesaikan tugas tanpa mengalami kendala, yang menunjukkan bahwa fitur ini telah berfungsi dengan baik dan mudah digunakan oleh pengguna.

V. KESIMPULAN

Pengembangan tampilan dinamis aplikasi Key's Laundry menggunakan Jetpack Compose telah berhasil diselesaikan dengan pendekatan modular yang disesuaikan kebutuhan pengguna. Jetpack Compose memudahkan pembuatan tampilan secara deklaratif sehingga setiap halaman bisa dibangun lebih efisien, responsif terhadap data, dan mudah disesuaikan untuk peran admin maupun karyawan. Aplikasi ini dilengkapi fitur operasional lengkap seperti pengelolaan layanan, sistem pemesanan, pencatatan pelanggan, laporan keuangan, hingga nota digital dan pencetakan otomatis yang terintegrasi dalam satu sistem dengan tampilan konsisten. Hasil User Acceptance Testing (UAT) menunjukkan tingkat keberhasilan 100% dari pengguna dalam menggunakan semua fungsi utama tanpa kendala, membuktikan bahwa aplikasi tidak hanya berfungsi secara teknis tetapi juga mudah digunakan dan nyaman untuk berinteraksi, sehingga mampu mendukung digitalisasi operasional laundry secara efisien sesuai kebutuhan UMKM seperti Key's Laundry.

REFERENCES

[1] C. Diantoni, O. Komarudin and A. Rizal, "Arsitektur MVVM dan Framework Jetpack Compose pada Pengembangan Aplikasi Android," *Jurnal Mahasiswa Teknik Informatika,* pp. 3216-3217, 2024.

[2] M. Rahmadan and C. E. Gunawan, "Perancangan Data Flow Diagram Aplikasi Tabungan Sampah PT Pusri Palembang," *Prosiding Seminar Nasional Mini Riset Mahasisw,* vol. 3, no. 1, pp. 1-9, 2024.

[3] I. S. Akbar and T. Haryanti, "Pengembangan Entity Relationship Diagram Database," *Ilmiah Computing Insight,* vol. 3 No.2, pp. 29-31, 2021.

[4] L. Setiyanti, "Desain Sistem: Use Case Diagram," *Inovasi dan Adopsi Teknologi ,* vol. 1 No.1, pp. 246-258, 2021.

[5] A. P. Kusuma and A. Yufron, "Analisis User Acceptance Test pada Aplikasi Pengiriman Barang dalam Menentukan Kualitas Sistem," *Ilmiah Teknik Informatika,* vol. 18 No.2, pp. 234-243, 2024.