Optimalisasi Aplikasi Flutter Untuk Sistem Point Of Sales (POS) Pada Perangkat Mobile Low End

Yosi Rumandha[1], Muhammad Asep Subandri.[2]

[1,2] *Program Studi Rekayasa Perangkat Lunak - Politeknik Negeri Bengkalis*

[3] *Jalan Bathin Alam Sei Alam – Bengakalis - Indonesia*

yosirmndhaa04@gmail.com, subandri@polbeng.ac.id

Abstrak— Penelitian ini bertujuan mengembangkan aplikasi *Point of Sales* (POS) berbasis Flutter yang ringan dan dapat berjalan stabil pada perangkat *mobile low-end*. Aplikasi dirancang untuk membantu pelaku UMKM dalam mengelola transaksi penjualan, data barang, serta laporan keuangan secara offline. Metode penelitian menggunakan SDLC dengan pendekatan *Waterfall*. Teknik optimalisasi yang diterapkan meliputi *lazy loading, GetX state management*, kompresi asset, dan *build optimization*. Pengujian dilakukan menggunakan *Black Box Testing*, *Flutter DevTools, App Size Analyzer*, dan ADB. Hasil penelitian menunjukkan ukuran APK berhasil dikurangi dari ±30 MB menjadi 17,3 MB, penggunaan memori menurun dari ±92 MB menjadi 11,6 MB, serta performa aplikasi meningkat hingga 60 FPS dengan *loading time* kurang dari 1 detik. Aplikasi yang dikembangkan dinilai ringan, responsif, dan sesuai digunakan pada perangkat berspesifikasi rendah.

Kata Kunci— Flutter, *Point of Sales* (POS), Optimalisasi Aplikasi, Perangkat *Low-End*, UMKM.

Abstract— *This study aims to develop a lightweight Flutter-based Point of Sales (POS) application that can run stably on low-end mobile devices. The application is designed to help MSMEs manage sales transactions, product data, and financial reports offline. The research method used is SDLC with the Waterfall approach. The optimization techniques applied include lazy loading, GetX state management, asset compression, and build optimization. Testing was conducted using Black Box Testing, Flutter DevTools, App Size Analyzer, and ADB. The results show that the APK size was reduced from approximately 30 MB to 17.3 MB, memory usage decreased from approximately 92 MB to 11.6 MB, and application performance improved to 60 FPS with loading times under 1 second. The developed application is considered lightweight, responsive, and suitable for low-specification devices.*

Keyword— *Flutter, Cashier Application, Low-End Devices, Financial, Recording, Offline*.

I. PENDAHULUAN

Perkembangan teknologi informasi memberikan dampak yang signifikan terhadap cara pelaku usaha menjalankan kegiatan operasionalnya. Salah satu bentuk penerapan teknologi yang umum digunakan adalah sistem *Point Of Sales* (POS) atau aplikasi kasir digital yang membantu proses pencatatan keuangan, pengelolaan stok, pembukuan, dan pengelolaan data usaha. Namun, berdasarkan hasil observasi dan penyebaran kuesioner kepada pelaku UMKM serta toko kecil, ditemukan bahwa banyak pengguna mengalami kesulitan dalam mengoperasikan aplikasi kasir yang tersedia saat ini karena dianggap terlalu kompleks, berat ketika dijalankan, dan membutuhkan koneksi internet yang stabil.

Namun, kenyataan di lapangan menunjukkan bahwa tidak semua pelaku UMKM mampu menggunakan aplikasi modern yang ada. Banyak di antara mereka masih menggunakan smartphone dengan spesifikasi yang rendah (*low end*) yang memiliki keterbatasan dalam memori dan kapasitas penyimpanan. Aplikasi *Point Of Sales* (POS) yang tersedia umunya memerlukan spesifikasi perangkat tinggi dan koneksi internet yang stabil, sehingga sulit dijalankan di perangkat sederhana.

Berdasarkan hasil kuesioner kepada responden, sebagian besar pelaku usaha menginginkan aplikasi kasir yang mudah digunakan, tidak berbayar, ringan, dan dapat berjalan secara *offline*. Mereka berharap aplikasi dapat membantu dalam mencatat pemasukan dan pengeluaran usaha, penyusunan laporan keuangan, serta pemantauan pembukuan usaha tanpa memerlukan alat tambahan seperti *printer thermal, barcode scanner*, ataupun laci kasir otomatis. Temuan ini menunjukkan bahwa pelaku UMKM lebih membutuhkan aplikasi dengan fungsi dasar yang efisien dibandingkan fitur-fitur kompleks yang memerlukan perangkat eksternal. Oleh karena itu, diperlukan pengembangan aplikasi kasir sederhana berbasis *mobile* yang tetap fungsional, ringan, dan sesuai dengan kebutuhan nyata dilapangan.

Optimalisasi dalam konteks aplikasi *mobile* adalah proses meningkatkan kinerja aplikasi agar berjalan dengan lebih ringan, cepat, dan efisien, khususnya pada perangkat dengan spesifikasi rendah. Optimalisasi mencakup pengurangan ukuran file aplikasi, penghematan penggunaan memori dan prosesor, serta pengaturan menejemen kode dan *resource* seperti *lazy loading,* kompresi asset, *state management,* dan *build optimization.* Dengan optimalisasi, aplikasi dapat dijalankan secara stabil meskipun pada perangkat low end.

Permasalahan ini mendorong dilakukannya pengembangan sistem *Point of Sales* yang dioptimalkan secara khusus untuk perangkat *mobile low-end*, dengan tetap mempertahankan fungsionalitas penting seperti pencatatan keuangan, pelaporan keuangan, dan pembukuan sederhana. Solusi yang diterapkan dalam penelitian ini adalah penggunaan Flutter *framework User Interface* (UI) dari *Google* yang mendukung pengembangan aplikasi lintas platform dengan satu basis kode, serta dikenal efisien dalam konsumsi sumber daya perangkat penelitian ini. Farhan Mualif dan Ikrimach mengembangan sistem kasir berbasis Android dengan Flutter dan Laravel yang

berhasil meningkatkan efisiensi operasioanl serta mengurangi kesalahan dalam proses transaksi [1]. Hal serupa juga ditemukan dalam pengembangan sistem *Point of Sales* di Kooi Coffee, dimana Flutter digunakan untuk membangun antarmuka *mobile* yang terintegrasi dengan *REST API* dan thermal printer. Sistem ini terbukti mampu menyederhanakan proses transaksi dan mempercepat percetakan struk penjualan secara *realtime* [2].

Selain efisiensi, aspek kenyamanan pengguna juga menjadi perhatian penting. Aplikasi *Point of Sales* yang baik harus mudah di gunakan oleh kasir tanpa harus memerlukan pelatihan teknis yang rumit. Dalam penelitian oleh Ahmad Alif Fauzan dkk, pendekatan *Goal Directed Design* diterapkan untuk memastikan desain antarmuka aplikasi POS berbasis Flutter yang benar-benar sesuai dengan kebutuhan pengguna[3].

Dengan adanya pengembangan aplikasi kasir sederhana berbasis Flutter ini, diharapkan dapat membantu pelaku UMKM untuk meningkatkan produktivitas dan efisiensi operasional usaha mereka. Aplikasi ini tidak hanya mendukung proses digitalisasi usaha kecil, tetapi juga memperkuat kemampuan mereka dalam melakukan pencatatan dan pelaporan keuangan usaha dengan cara yang mudah, cepat, dan akurat. Oleh karena itu, penelitian ini menghasilkan aplikasi *Point Of Sales* (POS) yang dapat digunakan secara layak, ringan, dan sesuai dengan kebutuhan nyata pelaku UMKM dilapangan.

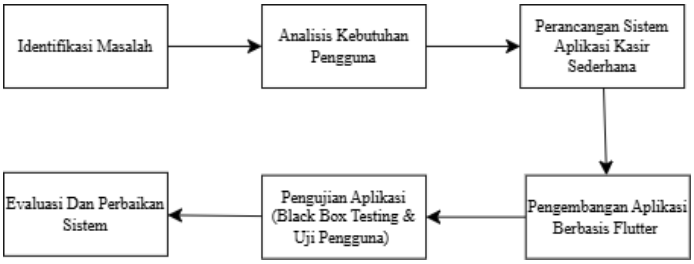
II. METODE PENELITIAN

A. *Software Development Life Cycle (SDLC)*

Metode penelitian yang digunakan dalam pengembangan sistem ini adalah *Software Development Life Cycle(SDLC*) dengan pendekatan model *Waterfall*. Tahapan pengembangan meliputi: (1) identifikasi masalah dan analisis kebutuhan melalui wawancara serta kuesioner terhadap pelaku UMKM, (2) perancangan sistem menggunakan diagram UML (*Use Case dan Activity Diagram),* (3) implementasi aplikasi berbasis *framework Flutter* dengan fokus pada optimalisasi performa melalui penerapan *lazy loading, state management GetX,* kompresi aset, dan *build optimization*, serta (4) pengujian fungsionalitas menggunakan metode *Black Box Testing*. Selain itu, dilakukan pengujian performa dan efisiensi sumber daya menggunakan *Flutter DevTools, App Size Analyzer,* dan *Android Debug Bridge* (ADB) untuk memastikan aplikasi berjalan stabil pada perangkat mobile *low-end* [4].

B. *Prosedur Penelitian*

Penelitian ini dilakukan beberapa tahapan yang disusun secara sistematis untuk mengembangkan aplikasi kasir sederhana berbasis Flutter. Tahap awal dimulai dengan identifikasi masalah pada sistem transaksi manual dan analisis kebutuhan pengguna melalui kuesioner untuk mengetahui fitur apa yang dibutuhkan. Selanjutnya dilakukan perancangan sistem dan antarmuka aplikasi, kemudian pengembangan aplikasi menggunakan Flutter dengan fokus pada optimasi agar ringan dan efisien diperangkat *low end.* Setelah aplikasi selesai dibuat, dilakukan pegujian menggunakan metode *black box testing* untuk memastikan seluruh fitur berjalan dengan baik dan diakhiri dengan evaluasi dan perbaikan sistem berdasarkan hasil pengujian. Bisa dilihat pada gambar 1.



Gbr 1 Prosedur Penelitian

C. *Pengumpulan Data*

Pengumpulan data dilakukan melalui wawancara dan penyebaran kuesioner kepada pelaku UMKM. Wawancara digunakan untuk mengetahui proses transaksi dan kendala yang dihadapi pengguna, sedangkan kuesioner digunakan untuk mengetahui kebutuhan fitur, kemudahan penggunaan, dan performa aplikasi pada perangkat *mobile low-end*. Data yang diperoleh digunakan sebagai dasar dalam analisis kebutuhan dan pengembangan aplikasi POS berbasis Flutter .

D. *Teknik Optimasi*

Langkah-langkah dalam tahapan optimasi dalam peneitian ini yakni sebagai berikut:

a. Lazy Loading

Teknik *lazy loading* digunakan untuk memuat data atau halaman hanya saat dibutuhkan oleh pengguna. Dengan teknik ini, aplikasi tidak langsung memuat seluruh data secara bersamaan sehingga penggunaan RAM menjadi lebih ringan dan proses loading aplikasi menjadi lebih cepat, terutama pada perangkat *mobile low-end* [5].

b. GetX State Management.

GetX digunakan sebagai *state management* untuk mengelola data, navigasi, dan perubahan tampilan aplikasi secara lebih efisien. Penggunaan *GetX* membantu mengurangi penggunaan *resource* yang berlebihan sehingga aplikasi dapat berjalan lebih ringan, cepat, dan *responsive* [6].

c. Kompresi Asset

Kompresi asset dilakukan dengan mengurangi ukuran gambar, ikon, dan file pendukung lainnya tanpa mengurangi kualitas secara signifikan. Teknik ini bertujuan untuk memperkecil ukuran aplikasi (APK) serta menghemat penggunaan penyimpanan pada perangkat pengguna [7].

d. Build Optimization

Build optimization dilakukan dengan menghapus kode, library, dan resource yang tidak digunakan pada aplikasi. Teknik ini membantu mengurangi ukuran akhir aplikasi serta meningkatkan performa sistem agar lebih optimal saat dijalankan pada perangkat dengan spesifikasi rendah [8].

E. *Parameter Pengujian*

a. Flutter DevTools

Flutter DevTools digunakan untuk menganalisis performa aplikasi selama proses pengujian. Tools ini membantu dalam

memantau penggunaan memori (RAM), CPU, serta performa rendering aplikasi untuk memastikan aplikasi dapat berjalan dengan ringan dan stabil pada perangkat mobile low-end.

b. App Size Analyzer

App Size Analyzer digunakan untuk menganalisis ukuran aplikasi dan mengetahui komponen yang paling banyak menggunakan ruang penyimpanan. Melalui tools ini, proses optimasi ukuran APK dapat dilakukan dengan mengurangi asset dan library yang tidak diperlukan agar aplikasi menjadi lebih ringan [9].

c. *Android Debug Bridge* (ADB)

ADB digunakan untuk membantu proses pengujian dan monitoring aplikasi pada perangkat Android. Tools ini digunakan untuk melihat performa aplikasi secara langsung, seperti penggunaan memori, proses debugging, serta memastikan aplikasi berjalan dengan baik pada perangkat dengan spesifikasi rendah [10].

III. HASIL DAN PEMBAHASAN

Tahap ini memaparkan hasil implementasi dari perancangan sistem yang telah dilakukan, serta pembahasan mengenai bagaimana sistem mampu membantu dalam melakukan pencatatan keuangan.

A. Hasil Implementasi Sistem

Aplikasi *Point of Sales* (POS) berbasis Flutter berhasil dikembangkan dengan fitur utama: (1) autentikasi pengguna berbasis peran (Pemilik Toko dan Kasir), (2) pengelolaan data barang dengan dukungan *scan barcode*, (3) transaksi penjualan offline, (4) pencatatan pemasukan/pengeluaran, dan (5) laporan keuangan sederhana. Antarmuka dirancang minimalis untuk memudahkan penggunaan pada perangkat *low-end*. Seluruh fitur diimplementasikan menggunakan *state management GetX* dan penyimpanan lokal SQLite untuk memastikan aplikasi berjalan tanpa koneksi internet.

B. Hasil Pengujian Fungsional

Pengujian fungsional menggunakan metode *Black Box Testing* terhadap 7 modul utama (Login, Data Barang, *Scan Barcode*, Transaksi, Riwayat, Laporan, Logout) menunjukkan hasil 100% berhasil. Seluruh skenario pengujian menghasilkan output sesuai, termasuk validasi input, manajemen hak akses, dan penyimpanan data lokal. Bisa dilihat pada tabel 1.

TABEL I
RINGKASAN PENGUJAIN FUNGSIONAL

Modul	Scenario Kritis	Hasil
Login	Autentikasi role berbeda	Berhasil
Data Barang	CRUD & Scan Barcode	Berhasil
Transaksi	Perhitungan total & kembalian	Berhasil
Laporan	Fiter periode & ekspor PDF	Berhasil
Logout	Reset sesi & keamanan data	Berhasil

C. Hasil Optimalisasi Performa

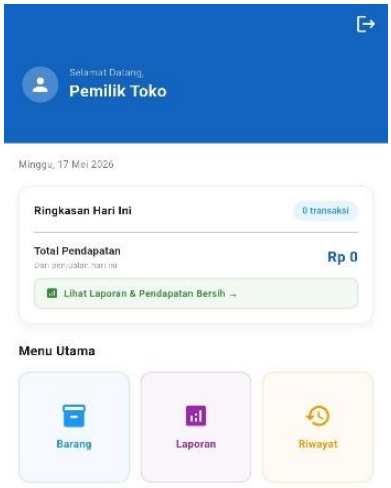
Penerapan teknik optimasi berhasil meningkatkan performa aplikasi secara signifikan. Ukuran APK berkurang dari ±30 MB menjadi 17,3 MB, penggunaan memori menurun dari ±92 MB menjadi 11,6 MB, dan frame rate meningkat hingga 60 FPS. Selain itu, waktu loading menjadi kurang dari 1 detik dan penggunaan CPU menurun menjadi kurang dari 5%. Hasil ini menunjukkan bahwa aplikasi lebih ringan dan sesuai digunakan pada perangkat mobile *low-end*. Bisa dilihat pada tabel 2.

TABEL II
PERBANDINGAN PENGUJAIN SEBEUM DAN SESUDAH OPTIMALISASI

Asepek Kinerja	Sebelum	Sesudah	Target Low-End
Ukuran APK	± 30 MB	17,3 MB	<25 MB
Dart Heap Memory	± 92 MB	11,6 MB	<50 MB
Frame Rate (FPS)	± 10 FPS (jank)	60 FPS (stabil)	>30 FPS
Loading Time	>2 detik	<1 detik	<1,5 detik
CPU Usage	Tinggi (>15%)	Rendah (<5%)	<10%

D. Halaman Dahboard Pemilik Toko

Halaman dashboard admin merupakan halaman utama yang ditampilkan setelah pengguna berhasil melakukan login. Halaman ini berfungsi untuk menampilkan ringkasan informasi keuangan harian, meliputi pemasukan, pengeluaran, dan saldo. Selain itu, halaman beranda juga menyediakan menu utama sebagai akses cepat ke fitur-fitur sistem seperti transaksi, data barang, riwayat, dan laporan. Bisa dilihat pada gambar 2.

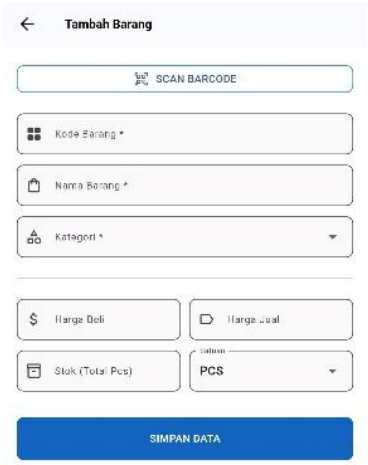


Gbr 2 Dashboard Admin

E. Halaman Tambah Data Barang (Pemilik Toko)

Modul Data Barang berfungsi sebagai pusat pengelolaan informasi inventaris yang memungkinkan Pemilik Toko dan Kasir untuk memantau stok serta detail produk secara terorganisir. Melaui halaman utama modul ini, pengguna dapat

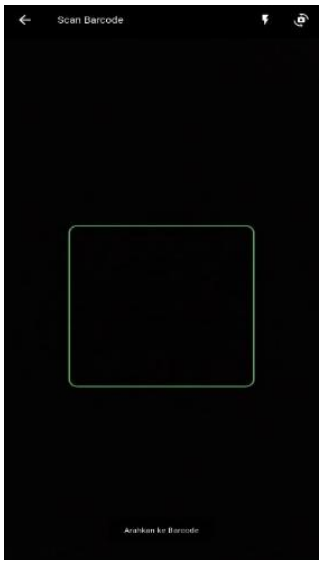
melihat daftar barang yang dilengkapi degan inisial visual, nama produk, kode barang, kategori, sissa stok, dan harga jual. Selain fitur pencarian yang memudahkan penemuan produk berdasarkan nama atau kode, modul ini juga menyediakan tombol "Tambah Barang" yang mengarahkan pengguna ke formular input lengkap. Pada bagian penginptan, sistem menyediakan opsi efisien berupa fitur *Scan Barcode* untuk mengisi kode barang secara otomatis menggunakan kamera, serta kolom manual untuk melengkapi nama, kategori, harga beli, harga jual, stok, dan satuan sebelum akhirnya disimpa ke dalam database. Bisa dilihat pada gambar 3.



Gbr 3 Tambah Data Barang

F. *Halaman Implementasi Scan Barcode*

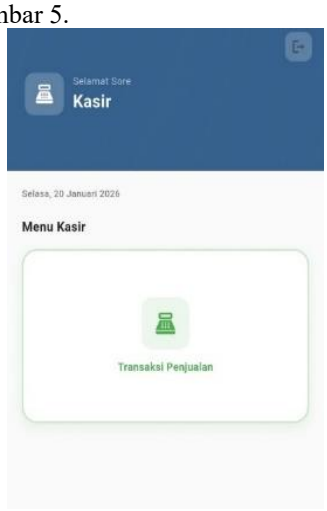
Modul *scan barcode* pada data barang merupakan fitur pendukung yang berfungsi sebagai opsi kedua untuk mempercepat penginputan kode barang baru ke dalam sistem. Melalui fitur ini, Pemilik Toko maupun Kasir dapat memindai barcode menggunakan kamera perangkat, yang secara otomatis akan mengisi kolom kode pada form tambah data barang. Secara teknis, implemenatsi ini menggunakan *Flutter barcode scanner* yang memanfaatkan kamera HP, dimana setiap kali barcode terdeteksi, sisem akan langsung menangkap nilainya menampilkan pada formular, dan memungkinkan pengguna untuk segera menyimpan data ke dalam database lokal [11]. Bisa dilihat pada gambar 4.



Gbr 4 *Scan Barcode*

G. *Halaman Dashboard Kasir*

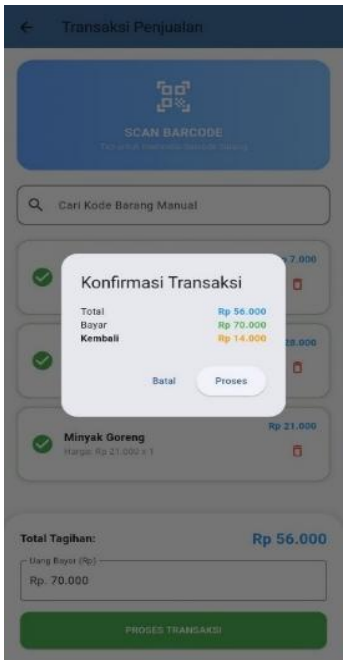
Halaman dashboard admin merupakan halaman utama yang ditampilkan setelah pengguna berhasil melakukan login. Bisa dilihat pada gambar 5.



Gbr 5 Dashboard Kasir

H. *Halaman Transaksi Penjualan*

Modul Transaksi Penjualan, kasir dapat memproses pesanan pelanggan dengan memindai kode barang secara praktis melalui fitur *Scan Barcode* atau mencarinya secara manual. Sistem akan menjumlahkan seluruh daftar barang belanjaan menjadi tota tagihan, dimana kasir kemudian cukup menginput nominal uang yang dibayarkan oleh pelanggan. Setelah menekan tombol Proses Transaksi, muncu jendela konfirmasi yang merincikan total belanja dan jumah kembalian secara otomatis. Bisa dilihat pada gambar 6.



Gbr 6 Transaksi Penjualan

I. Pembahasan

Hasil pengujian membuktikan bahwa teknik optimalisasi yang diterapkan efektif menurunkan penggunaan sumber daya aplikasi secara drastis. Pengurangan ukuran APK sebesar 42% dan penggunaan memori *Dart Heap* hingga 87% memungkinkan aplikasi berjalan stabil pada perangkat *low-end* tanpa mengorbankan fungsionalitas utama.

Pencapaian 60 FPS setelah optimalisasi menunjukkan bahwa pengelolaan state dengan *GetX* dan penerapan *lazy loading* berhasil meminimalkan jank pada antarmuka. Waktu loading <1 detik juga memenuhi ekspektasi pengguna UMKM yang membutuhkan responsivitas tinggi saat transaksi.

Keterbatasan yang teridentifikasi adalah peningkatan waktu render pada fitur ekspor PDF akibat kompleksitas *library* pdf, namun hal ini tidak mengganggu alur transaksi utama. Secara keseluruhan, aplikasi ini memenuhi kriteria "ringan, cepat, dan *offline-first*" yang dibutuhkan pelaku UMKM dengan perangkat terbatas.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa penerapan teknik optimalisasi pada aplikasi Point of Sales (POS) berbasis Flutter menunjukkan adanya peningkatan performa dibandingkan sebelum dilakukan optimalisasi. Peningkatan tersebut terlihat dari ukuran aplikasi yang lebih ringan, penggunaan memori yang lebih efisien, serta responsivitas sistem yang lebih baik saat dijalankan pada perangkat mobile berspesifikasi rendah (low-end). Meskipun demikian, hasil yang diperoleh belum sepenuhnya mencapai tingkat optimal yang maksimal dan masih terdapat ruang untuk pengembangan lebih lanjut guna meningkatkan efisiensi serta kinerja sistem pada penelitian selanjutnya.

B. Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan bahan pengembangan lebih lanjut. Aplikasi *Point of Sales* (POS) berbasis Flutter yang dikembangkan dalam penelitian ini telah berjalan dengan baik pada perangkat berspesifikasi rendah, namun masih memiliki potensi untuk ditingkatkan. Penelitian selanjutnya disarankan untuk menambahkan fitur lanjutan seperti manajemen pengguna yang lebih detail, laporan keuangan yang lebih kompleks, serta integrasi dengan layanan cloud agar data dapat tersinkronisasi antar perangkat. Selain itu, pengujian performa dapat diperluas dengan menggunakan lebih banyak variasi perangkat *low-end* dan versi sistem operasi Android yang berbeda untuk memperoleh hasil yang lebih komprehensif. Optimalisasi aplikasi juga dapat dikembangkan lebih lanjut dengan menerapkan teknik optimasi lanjutan guna meningkatkan efisiensi memori dan respons sistem. Diharapkan aplikasi ini dapat terus dikembangkan agar lebih adaptif terhadap kebutuhan UMKM dan perkembangan teknologi mobile di masa mendatang.

DAFTAR PUSTAKA

[1] View of Desain User Experience (UX) Pada Website Smart City untuk Meningkatkan Aksesibilitas Layanan Publik.

[2] A. Prawirdani and E. I. Sela, “Pengembangan Sistem Point of Sale Berbasis Web dan Mobile di Kooi Coffee,” *ILKOMNIKA*, vol. 6, no. 3, pp. 299–316, Dec. 2024, doi: 10.28926/ilkomnika.v6i3.689.

[3] A. Alif Fauzan and R. Taufiq Subagio, “Pembuatan Aplikasi Point Of Sales Berbasis Android Menggunakan Metode Goal Directed Design Untuk Meningkatkan Layanan Rumah Makan,” 2024.

[4] M. Sumbodo, “Analisis Metode System Development Life Cycle (Sdlc) Dalam Rancangan Sistem Informasi”

[5] V. Jain, “Optimizing Web Performance With Lazy Loading And Code Splitting,” *International Journal of Core Engineering & Management*, no. 7, 2022.

[6] U. Aswalekar and A. Vishwakarma, “State Management in Flutter: A Performance Comparison of GetX, Provider, Riverpod and BLoC”, [Online]. Available: www.ijsart.com

[7] U. Brawijaya, M. Z. Arfani, A. Pinandito, E. Muhammad, and A. Jonemaro, “Fakultas Ilmu Komputer Studi Perbandingan Pengembangan Aplikasi Progressive Web App pada Proses Kompresi Gambar Menggunakan React Image File Resizer dan Native,” 2017. [Online].

[8] T. Sotiropoulos, S. Chaliasos, D. Mitropoulos, and D. Spinellis, “Identifying Bugs in Make and JVM-Oriented Builds,” May 2020, doi: 10.1145/3428212.

[9] M. Markowski and J. Smołka, “A comparative analysis of resource use in the Flutter and React Native frameworks Analiza porównawcza wykorzystania zasobów w szkieletach programistycznych Flutter oraz React Native,” 2023.

[10] R. Prathiba and D. S. Ganesh, “International Journal of Innovative Research in Science Engineering and Technology (IJIRSET) ADB Debugging-Security Risks and Investigations”, doi: 10.15680/IJIRSET.2025.1404477.

[11] N. S. Mulyani and I. Najiyah, “Rancang Bangun Aplikasi Penjualan Menggunakan Barcode Scanner Berbasis Android,” *Jurnal Responsif*, vol. 4, no. 1, pp. 49–55, 2022,