

# PERANCANGAN DAN IMPLEMENTASI AWAL APLIKASI FINEME BERBASIS FLUTTER UNTUK PENCATATAN KEUANGAN PRIBADI DIGITAL

Muhammad Midayatul Arifin\*<sup>1)</sup>, Salsa Nabila Harahap<sup>2)</sup>, Ruth Amelia Vega S. Meliala<sup>3)</sup>, M Yazid Noor<sup>4)</sup>, Adidtya Perdana<sup>5)</sup>

1. Ilmu Komputer, Universitas Negeri Medan  
email: arif.4233250005@mhs.unimed.ac.id
2. Ilmu Komputer, Universitas Negeri Medan  
email:salsanhrp.4231250027@mhs.unimed.ac.id
3. Ilmu Komputer, Universitas Negeri Medan  
email : ruthamelia.4233250035@mhs.unimed.ac.id
4. Ilmu Komputer, Universitas Negeri Medan  
email : yazidnoor.4233550013@mhs.unimed.ac.id
5. Ilmu Komputer, Universitas Negeri Medan  
email: adidtya@unimed.ac.id

## **Abstract**

*Personal financial record keeping is often delayed or messy, making it difficult to summarize data, unconnected to budget limits, and easily overlooked. This research designed and built FineMe, a Flutter-based application for recording transactions, managing categories, setting total and per-category budgets, displaying daily charts, exporting data to CSV, and managing recurring transactions. The development followed a layered architecture software engineering approach with SQLite in the data layer, a repository for logic and aggregation, Riverpod for state synchronization, and a Material interface. Requirements were derived from daily usage scenarios and implemented iteratively, while functional testing assessed the accuracy of calculations and interface responsiveness. Results showed that the income, expense, and balance summaries were updated instantly, two separate daily charts were easy to read, budget progress was calculated accurately, valid CSV files were opened in a spreadsheet, and recurring transaction rules reduced repetitive input. These findings confirm the effective combination of Flutter, SQLite, and reactive state management for building a precise, responsive, and scalable financial recorder.*

**Kata Kunci:** Flutter, Personal Finance Application, Budget Management, SQLite, Riverpod

## **A. PENDAHULUAN**

Kesejahteraan finansial (*financial well-being*) merupakan salah satu tujuan utama yang ingin dicapai oleh setiap individu, yang merefleksikan kondisi di mana seseorang mampu memenuhi kebutuhan hidup saat ini dan di masa

depan serta merasa aman secara finansial [1]. Namun, perilaku masyarakat Indonesia yang cenderung konsumtif menjadi tantangan dalam mencapai tujuan tersebut [2]. Masalah ini diperburuk oleh tingkat literasi keuangan yang masih tergolong rendah, di mana data Otoritas

Jasa Keuangan (OJK) pada tahun 2019 menunjukkan indeks literasi keuangan masyarakat Indonesia baru mencapai 38,03 persen [2]. Salah satu akar permasalahan dalam manajemen keuangan pribadi adalah metode pencatatan yang tidak efisien. Banyak individu yang masih enggan atau malas melakukan pencatatan pengeluaran dan pemasukan secara manual di atas kertas [3]. Metode manual ini tidak hanya memakan waktu tetapi juga sangat rentan terhadap berbagai masalah, seperti kesalahan pencatatan, ketidaksesuaian data, hingga kehilangan data fisik yang mengakibatkan pengelolaan keuangan menjadi tidak efisien [4], [5].

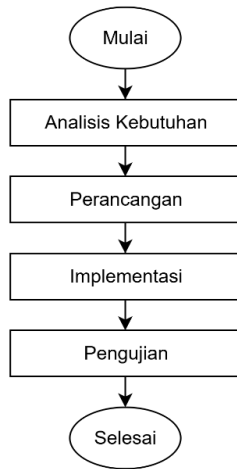
Seiring pesatnya perkembangan teknologi, aplikasi mobile menawarkan solusi praktis untuk mempermudah berbagai aspek kehidupan, termasuk manajemen keuangan [4]. Penggunaan aplikasi keuangan pada perangkat *smartphone* memungkinkan pengguna untuk mencatat, memantau, dan menganalisis kondisi keuangan secara *real-time* dengan lebih akurat dan efisien [5]. Mengingat dominasi platform Android dan iOS di pasar Indonesia [6], pengembangan aplikasi yang mampu berjalan di kedua sistem operasi menjadi sebuah keharusan untuk menjangkau pengguna yang lebih luas. Akan tetapi, pengembangan aplikasi *native* secara terpisah untuk setiap platform memerlukan sumber daya yang signifikan, basis kode yang berbeda, serta biaya dan waktu yang berlipat ganda, yang sering kali menghasilkan inkonsistensi fitur dan tampilan antarpengguna [6]. Sementara itu, aplikasi berbasis web yang diakses melalui peramban *mobile* sering kali tidak memberikan pengalaman pengguna yang optimal karena keterbatasan responsivitas antarmuka [6].

Untuk mengatasi tantangan tersebut, penelitian ini mengadopsi *framework* Flutter. Flutter adalah sebuah *Software*

*Development Kit* (SDK) yang dikembangkan oleh Google untuk membangun aplikasi *multiplatform* yang menarik, cepat, dan produktif [7], [8]. Keunggulan utamanya adalah kemampuan untuk menghasilkan aplikasi bagi Android dan iOS dari satu basis kode (*single codebase*) yang sama [6], [9], sehingga dapat mengurangi waktu dan biaya pengembangan secara signifikan [7]. Dengan menggunakan bahasa pemrograman Dart, Flutter mampu menghasilkan aplikasi dengan performa tinggi yang mendekati performa aplikasi *native* [7], [6]. Karena kemampuannya yang fleksibel dan efisien, Flutter telah banyak diadopsi dalam pengembangan berbagai jenis aplikasi, termasuk di sektor pendidikan dan bisnis [10].

Berdasarkan latar belakang tersebut, penelitian ini berfokus pada perancangan dan pembangunan sebuah aplikasi keuangan pribadi bernama "FineMe". Aplikasi ini dikembangkan untuk menjawab kebutuhan akan alat bantu manajemen keuangan yang fungsional dan mudah digunakan. Penelitian ini memiliki beberapa tujuan utama. Pertama, menghasilkan purwarupa aplikasi "FineMe" yang fungsional menggunakan *framework* Flutter, dengan fitur utama yang mencakup pencatatan transaksi, penyusunan anggaran, dan pembuatan laporan keuangan. Kedua, mengembangkan aplikasi dengan desain antarmuka pengguna (UI/UX) yang intuitif dan ramah pengguna (*user friendly*) untuk memberikan pengalaman terbaik dalam mengelola keuangan. Ketiga, melakukan pengujian fungsionalitas dan konsistensi performa aplikasi "FineMe" agar dapat berjalan secara optimal pada sistem operasi Android.

## B. METODE PENELITIAN



Gambar 1. Flowchart Penelitian

### Pendekatan Penelitian

Penelitian ini merupakan penelitian dan pengembangan perangkat lunak dengan pola *Waterfall* yang disederhanakan. Rangkaian kegiatan dilakukan berurutan mulai dari analisis kebutuhan, dilanjutkan perancangan, kemudian implementasi, dan diakhiri pengujian. Alur tersebut sesuai flowchart pada Gambar 1 sehingga keterkaitan antartahap mudah dipahami. Fokus penelitian adalah membangun FineMe, yaitu aplikasi Android yang kami namai untuk pengelolaan keuangan pribadi yang mampu mencatat transaksi, menetapkan anggaran bulanan, dan menyajikan laporan berupa ringkasan serta dua grafik harian yang memisahkan pemasukan dan pengeluaran.

### Ruang Lingkup dan Lingkungan

Ruang lingkup terbatas pada pengembangan aplikasi Android dengan penyimpanan lokal, sinkronisasi lintas perangkat dan integrasi layanan eksternal tidak menjadi bagian penelitian ini. Proses pengembangan dilakukan pada Flutter versi stabil dengan Android SDK. Pengujian dilakukan pada *emulator* dan perangkat fisik agar perilaku aplikasi dapat diamati pada kondisi yang beragam.

Adapun perangkat dan pustaka utama yang kami gunakan :

- Flutter/Dart untuk antarmuka dan logika aplikasi; prinsip Material 3 digunakan agar tampilan konsisten.
- Riverpod untuk pengelolaan state dan injeksi dependensi.
- SQLite melalui sqllite sebagai basis data tertanam.
- Pustaka pendukung intl (format tanggal/angka), fl\_chart (grafik), csv dan share\_plus (ekspor).

### Perancangan Sistem

Perancangan sistem pada aplikasi FineMe menggunakan pendekatan arsitektur berlapis (*layered architecture*) yang memungkinkan setiap komponen memiliki tanggung jawab yang terpisah dan mudah dikelola. Lapisan presentasi bertugas menampilkan antarmuka pengguna dengan lima *navigation bar* atau menu utama, yakni *Dashboard*, Transaksi, Anggaran, Kategori, dan Pengaturan. Lapisan ini menjadi titik interaksi utama antara pengguna dan sistem.

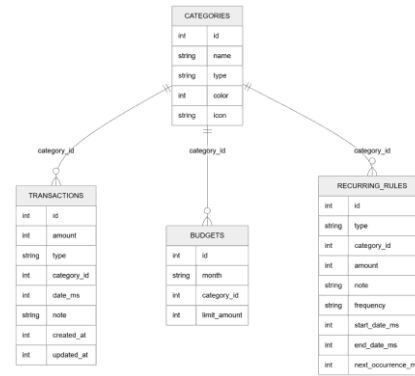
Selanjutnya, lapisan *state management* berperan menjaga konsistensi data dengan memperbarui tampilan secara otomatis setiap kali terjadi perubahan nilai di dalam sistem. Lapisan *repository* bertanggung jawab terhadap logika bisnis yang meliputi operasi penyimpanan, pengeditan, penghapusan, serta penghitungan agregat dari data yang disimpan. Sedangkan lapisan data access difungsikan sebagai penghubung antara aplikasi dan basis data lokal berbasis SQLite agar proses penyimpanan berlangsung efisien serta tetap dapat dioperasikan tanpa koneksi internet.

Struktur basis data dirancang dengan prinsip kesederhanaan namun tetap memperhatikan efisiensi dan

skalabilitas sistem. Terdapat tiga tabel utama yang menjadi inti penyimpanan data, yaitu:

- *Categories*, berisi informasi mengenai nama kategori, jenis transaksi (pemasukan atau pengeluaran), warna penanda, serta ikon yang digunakan dalam antarmuka.
- *Transactions*, mencatat aktivitas keuangan pengguna berupa jenis transaksi, referensi kategori, nominal dalam satuan rupiah yang disimpan sebagai bilangan bulat, waktu dalam format milidetik epoch, serta catatan tambahan dari pengguna.
- *Budgets*, menyimpan data terkait bulan pengamatan, acuan kategori yang bersifat opsional yang memungkinkan penyusunan anggaran total maupun per kategori dan nilai batas anggaran.

Selain ketiga tabel tersebut, terdapat tabel *Recurring Rules* yang digunakan untuk mengatur transaksi berulang secara otomatis berdasarkan periode tertentu seperti harian, mingguan, atau bulanan. Keempat tabel ini saling berhubungan melalui relasi kunci asing *category\_id*, di mana tabel *categories* menjadi entitas induk bagi *transactions*, *budgets*, dan *recurring\_rules*. Struktur hubungan antartabel divisualisasikan pada Gambar 2 yang menampilkan rancangan *Entity Relationship Diagram (ERD)* dari sistem FineMe.



**Gambar 2.** Entity Relationship Diagram (ERD) Aplikasi FineMe

Dalam perancangan tabel *transactions*, kolom waktu diberi indeks untuk mempercepat proses rekapitulasi bulanan serta pembentukan tren harian yang ditampilkan pada grafik *dashboard*. Sementara itu, pada tabel *budgets* diterapkan aturan keunikan pada pasangan *month* dan *category\_id* guna mencegah duplikasi data pada bulan yang sama.

Pendekatan ini menghasilkan struktur data yang konsisten, efisien, dan mudah dikembangkan. Selain itu, rancangan tersebut juga mendukung visualisasi keuangan yang akurat, di mana sumbu horizontal pada grafik merepresentasikan waktu (hari), sedangkan sumbu vertikal menunjukkan nilai transaksi dalam rupiah. Dengan desain semacam ini, aplikasi FineMe mampu memberikan pengalaman pengelolaan keuangan yang stabil, informatif, dan mudah diperluas untuk fitur-fitur berikutnya.

### Alur Kerja dan Urutan Implementasi

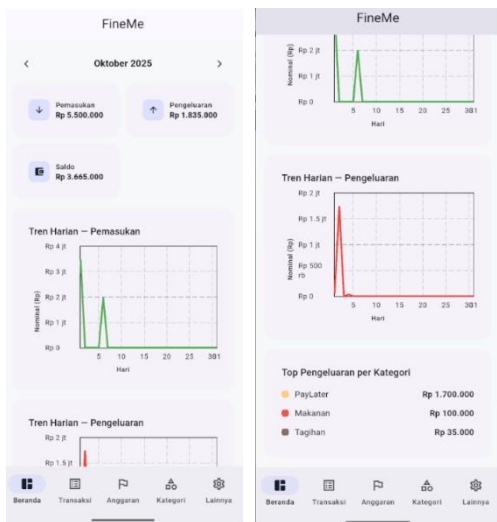
Alur penggunaan dirancang ringkas. Ketika pengguna menambahkan transaksi melalui formulir yang berisi jenis, kategori, jumlah, tanggal, dan catatan, aplikasi memvalidasi masukan lalu menyimpan data ke basis data. Sesudah penyimpanan berhasil, daftar transaksi diperbarui. Pada saat yang sama, *Dashboard* menghitung ulang total pemasukan, total pengeluaran, serta saldo.

Jika pada bulan yang sama terdapat anggaran, indikator progres otomatis menyesuaikan. Pengguna dapat memperbaiki atau menghapus transaksi melalui ikon yang tersedia, dan seluruh data transaksi dapat diekspor ke berkas CSV dari menu Pengaturan.

Untuk menjaga kestabilan pembangunan dan menghindari pekerjaan ulang, urutan implementasi mengikuti ketergantungan data. Tahap pertama adalah Kategori sebagai sumber referensi bagi transaksi. Tahap kedua adalah Transaksi sebagai data operasional yang menjadi dasar seluruh perhitungan. Tahap ketiga adalah Anggaran yang menghitung progres berdasarkan pengeluaran pada bulan aktif. Tahap terakhir adalah Laporan dan Ekspor, yakni penyusunan ringkasan serta dua grafik harian dan penyediaan berkas CSV untuk keperluan arsip atau analisis lanjutan.

### C. HASIL DAN PEMBAHASAN

#### Hasil Fungsional Aplikasi



**Gambar 3.** Halaman Dashboard

Pada Gambar 3 terlihat tampilan utama aplikasi FineMe, yaitu *Dashboard* yang berfungsi sebagai pusat informasi keuangan pengguna. Bagian atas layar menampilkan tiga ringkasan sederhana

namun informatif, yaitu total pemasukan, total pengeluaran, dan saldo terkini. Di bawahnya terdapat dua *line chart* yang masing-masing menggambarkan pergerakan pemasukan dan pengeluaran per hari selama satu bulan. Pemisahan kedua grafik ini bertujuan agar perubahan nilai harian mudah diamati tanpa saling tumpang-tindih. Garis pada grafik menunjukkan kenaikan setiap kali terjadi transaksi dan tetap datar pada hari tanpa aktivitas keuangan. Dengan visualisasi seperti ini, pengguna dapat memahami pola keuangan pribadinya secara cepat dan intuitif.

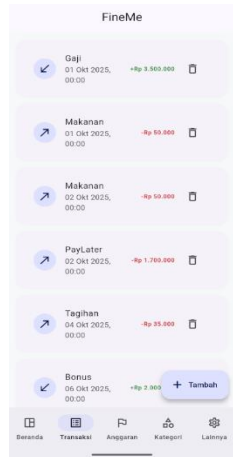
The 'Tambah Transaksi' form includes the following fields and options:

- Transaction Type:** Radio buttons for 'Pengeluaran' (checked) and 'Pemasukan'.
- Kategori (Category):** A dropdown menu.
- Jumlah (IDR, tanpa titik):** A text input field.
- Catatan (Notes):** A text input field.
- Tanggal (Date):** A date picker showing '09 Okt 2025, 03:10' with a 'Pilih Tanggal' button.
- Buttons:** 'Batal' (Cancel) and 'Simpan' (Save).

**Gambar 4.** Formulir Transaksi

Cuplikan pada Gambar 4 memperlihatkan proses pengguna ketika menambahkan transaksi baru. Formulir dirancang sederhana agar proses pencatatan terasa cepat dan tidak membingungkan. Pengguna hanya perlu memilih jenis transaksi (pemasukan atau pengeluaran), menentukan kategori, menuliskan jumlah dalam rupiah, serta memilih tanggal dan catatan tambahan bila diperlukan. Setelah menekan tombol simpan, data otomatis masuk ke basis data

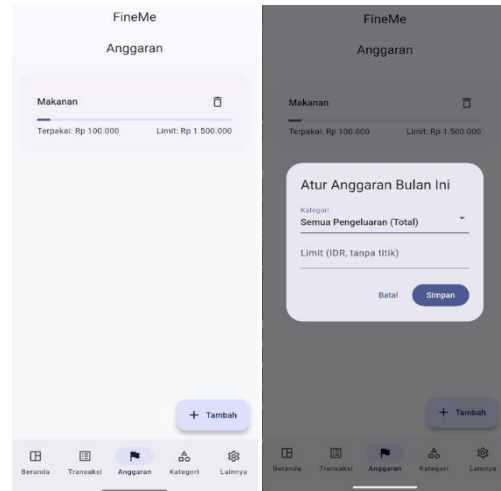
dan seluruh tampilan di *Dashboard* ikut diperbarui secara langsung tanpa perlu dimuat ulang seperti yang diperlihatkan pada Gambar 5 berikut ini.



**Gambar 5.** Daftar Transaksi

Daftar transaksi seperti yang terlihat pada Gambar 5 menampilkan seluruh riwayat aktivitas keuangan. Entri paling baru akan berada di bagian atas agar pengguna mudah melihat transaksi terakhir. Setiap entri dilengkapi ikon hapus yang disertai dialog konfirmasi sebelum penghapusan dilakukan. Cara ini meminimalkan kesalahan pengguna sekaligus menjaga integritas data. Perubahan apa pun yang dilakukan pada daftar ini otomatis akan mempengaruhi nilai total pemasukan, pengeluaran, dan saldo pada *Dashboard*.

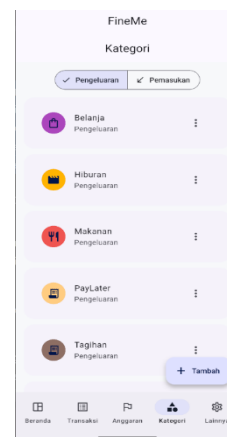
Gambar 6 memperlihatkan fitur pengaturan anggaran yang membantu pengguna mengelola keuangan bulanan. Pengguna dapat menentukan batas maksimal pengeluaran, baik secara keseluruhan maupun untuk setiap kategori. Aplikasi akan menampilkan indikator progres yang menunjukkan persentase pengeluaran terhadap batas yang telah ditetapkan.



**Gambar 6.** Halaman Anggaran

Saat nilai mendekati atau melebihi batas, indikator memberikan tanda warna yang mudah dikenali sehingga pengguna dapat menyesuaikan pola belanjanya.

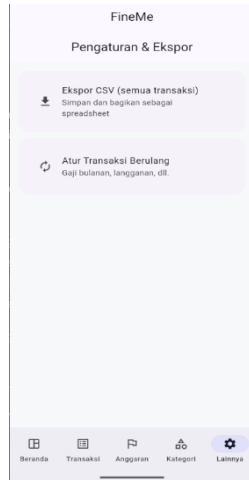
Daftar kategori berfungsi sebagai pengelompokan transaksi agar laporan keuangan lebih terstruktur. Setiap kategori memiliki nama, ikon, warna, serta jenis transaksi (pemasukan atau pengeluaran).



**Gambar 7.** Daftar Kategori dengan variasi ikon dan warna untuk membedakan jenis transaksi

Daftar kategori berfungsi sebagai pengelompokan transaksi agar laporan keuangan lebih terstruktur. Setiap kategori memiliki nama, ikon, warna, serta jenis transaksi (pemasukan atau pengeluaran).

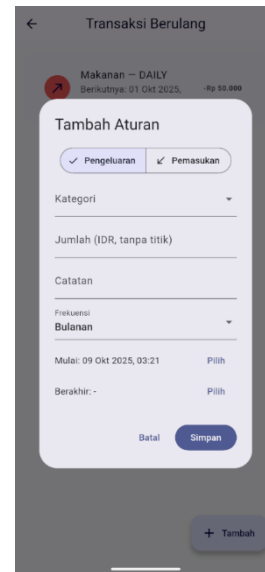
Perubahan pada kategori secara otomatis diperbarui di seluruh bagian aplikasi, termasuk formulir transaksi dan rekapitulasi keuangan bulanan.



**Gambar 8.** Halaman Lainnya

Fitur ekspor pada Gambar 8 memungkinkan pengguna menyimpan seluruh data transaksi ke dalam format CSV yang dapat dibuka melalui aplikasi *spreadsheet* seperti Microsoft Excel atau Google Sheets. *File* hasil ekspor berisi kolom *id*, *type*, *category*, *amount*, *date*, dan *note*. Dengan fitur ini, pengguna dapat melakukan analisis tambahan atau mencetak laporan keuangan pribadi di luar aplikasi.

Fitur transaksi berulang pada Gambar 6,7 dan 9 dibuat untuk memudahkan pencatatan aktivitas keuangan yang bersifat rutin, seperti gaji bulanan, tagihan internet, atau langganan aplikasi. Setelah pengguna menyimpan aturan transaksi, sistem akan menambahkan data secara otomatis sesuai jadwal yang telah ditentukan. Dengan demikian, pengguna tidak perlu mencatat transaksi yang sama setiap periode dan data keuangan tetap konsisten.



**Gambar 9.** Halaman Atur Transaksi Berulang

### Implementasi Sistem

Aplikasi FineMe dikembangkan menggunakan pendekatan berlapis (*layered architecture*) agar setiap bagian sistem memiliki tanggung jawab yang jelas. Lapisan paling bawah adalah lapisan data yang menggunakan SQLite untuk penyimpanan lokal. Struktur data terdiri atas tiga tabel utama: *categories*, *transactions*, dan *budgets*. Penyimpanan nominal dilakukan dalam bentuk bilangan bulat rupiah untuk menjaga akurasi dan mencegah kesalahan pembulatan. Kolom *date\_ms* pada tabel transaksi diberikan indeks agar proses pencarian dan rekap data lebih cepat.

Lapisan berikutnya adalah *repository*, yang berfungsi mengatur proses pertukaran data antara basis data dan antarmuka pengguna. *Repository* menangani operasi penyimpanan, pembaruan, penghapusan, serta pengambilan data, termasuk perhitungan total pemasukan, total pengeluaran, progres anggaran, dan pembuatan grafik harian.

Lapisan manajemen *state* memanfaatkan *Riverpod* untuk menjaga

agar tampilan selalu sinkron dengan data terkini. Setiap kali pengguna menambah, menghapus, atau mengubah transaksi, seluruh bagian aplikasi yang terkait langsung diperbarui secara otomatis.

Lapisan teratas adalah presentasi (UI/UX), yang bertanggung jawab menampilkan antarmuka kepada pengguna. Antarmuka FineMe dirancang menggunakan konsep *Material Design* sehingga sederhana, konsisten, dan mudah digunakan. Lima halaman utama disediakan, yaitu *Dashboard*, *Transaksi*, *Anggaran*, *Kategori*, dan *Pengaturan*.

**Tabel 1.** Struktur Lapisan Implementasi FineMe

Lapisan	Fungsi Utama	Komponen Penting
<b>Data</b>	Menyimpan dan mengelola data lokal	SQLite (categories, transactions, budgets)
<b>Repository</b>	Menangani logika bisnis dan agregasi data	CRUD, perhitungan ringkasan, grafik, progres anggaran
<b>Manajemen State</b>	Menjaga sinkronisasi antara data dan tampilan	Provider Riverpod
<b>Presentasi (UI/UX)</b>	Menampilkan antarmuka pengguna	Halaman Dashboard, Transaksi, Anggaran, Kategori, Pengaturan

## Pembahasan

Hasil implementasi aplikasi FineMe menunjukkan bahwa sistem yang dikembangkan telah berjalan dengan stabil dan sesuai dengan rancangan awal. Setiap fitur utama bekerja sebagaimana mestinya dan memberikan pengalaman penggunaan yang mudah, efisien, serta membantu pengguna dalam memahami kondisi keuangannya secara lebih jelas.

Penerapan manajemen *state* menggunakan *Riverpod* terbukti efektif dalam menjaga konsistensi antara data dan tampilan antarmuka. Setiap perubahan data baik penambahan, penghapusan, maupun pembaruan transaksi akan langsung tercermin pada halaman *Dashboard* tanpa perlu melakukan pemuatan ulang. Hal ini menunjukkan bahwa sistem mampu memperbarui informasi secara waktu nyata dan memberikan respons yang cepat terhadap interaksi pengguna.

Dari sisi kinerja dan efisiensi data, penggunaan indeks pada kolom waktu (*date\_ms*) terbukti meningkatkan kecepatan akses terhadap data transaksi. Dengan teknik ini, sistem tidak perlu membaca seluruh data setiap kali proses rekap atau visualisasi dilakukan. Penggunaan indeks tersebut mempercepat pemrosesan laporan bulanan dan pembuatan grafik harian, menjadikan tampilan tetap ringan serta responsif meskipun jumlah data terus bertambah.

Keakuratan perhitungan juga menjadi keunggulan tersendiri dari aplikasi ini. Penyimpanan nilai uang dalam bentuk bilangan bulat (*integer*) membantu menjaga presisi angka dan menghindari kesalahan pembulatan yang sering muncul pada tipe desimal. Pendekatan ini membuat hasil ringkasan, baik yang tampil di aplikasi maupun yang diekspor ke dalam format CSV, tetap akurat dan konsisten dengan perhitungan manual.

Dari aspek antarmuka dan kenyamanan pengguna, desain aplikasi dirancang sederhana namun tetap informatif. Pemisahan antara grafik pemasukan dan pengeluaran membantu pengguna memahami tren keuangannya tanpa kebingungan akibat tumpang tindih data. Selain itu, adanya indikator progres anggaran memberikan peringatan visual saat pengeluaran mendekati batas yang telah ditentukan, sehingga pengguna dapat mengatur pengeluarannya dengan lebih bijak. Fitur hapus transaksi yang dilengkapi konfirmasi juga berfungsi untuk meminimalkan kesalahan saat pengguna melakukan pengelolaan data.

Fitur transaksi berulang merupakan elemen tambahan yang memberikan nilai praktis bagi pengguna. Melalui fitur ini, transaksi yang bersifat rutin, seperti gaji, biaya langganan, atau pembayaran bulanan yang dapat dicatat secara otomatis sesuai jadwal yang ditentukan. Mekanisme tersebut menghemat waktu dan memastikan data keuangan pribadi selalu tercatat lengkap tanpa harus melakukan input berulang.

Meskipun sistem telah berfungsi dengan baik, masih terdapat ruang untuk penyempurnaan di masa mendatang. Saat ini, FineMe masih beroperasi secara lokal di satu perangkat, sehingga sinkronisasi lintas perangkat maupun pencadangan berbasis awan belum diimplementasikan. Integrasi dengan layanan penyimpanan *cloud* dapat menjadi langkah penting agar data keuangan lebih aman dan mudah diakses dari berbagai perangkat. Selain itu, pengembangan tambahan seperti pembuatan laporan lintas bulan, ekspor ke format PDF, serta penambahan variasi frekuensi dalam pengaturan transaksi berulang dapat meningkatkan fleksibilitas dan daya guna aplikasi.

Secara keseluruhan, pengujian dan hasil implementasi menunjukkan bahwa rancangan arsitektur berlapis yang digunakan dalam pengembangan FineMe

terbukti mendukung sistem yang efisien, akurat, dan mudah digunakan. Struktur data yang sederhana namun terorganisasi dengan baik, ditambah penerapan *Riverpod* sebagai pengendali alur data, berhasil menghasilkan sistem keuangan pribadi yang responsif dan stabil. Temuan ini memperlihatkan bahwa pengelolaan keuangan berbasis aplikasi lokal dapat berjalan efektif bila didukung oleh desain data yang tepat dan antarmuka yang ramah pengguna.

## D. KESIMPULAN DAN SARAN

### Kesimpulan

Penelitian ini berhasil mewujudkan FineMe sebagai pencatat keuangan pribadi yang sederhana, akurat, dan nyaman dipakai sehari-hari. Fitur inti berjalan selaras dimana pengguna dapat menambah dan menghapus transaksi, mengelola kategori, menetapkan anggaran baik total maupun per kategori, mengekspor data ke CSV, serta membuat aturan transaksi berulang. *Dashboard* merangkum kondisi bulan berjalan melalui tiga kartu ringkasan dan dua grafik harian yang dipisah antara pemasukan dan pengeluaran, sehingga perubahan nilai dari hari ke hari mudah dikenali. Penyimpanan nominal dalam bilangan bulat rupiah menjaga presisi, sedangkan pemakaian indeks waktu mempercepat perhitungan dan penyajian data.

Dari sisi rekayasa perangkat lunak, arsitektur berlapis yang memisahkan penyimpanan data berbasis SQLite, logika pada *repository*, manajemen *state* dengan *Riverpod*, dan antarmuka Material menghasilkan sistem yang konsisten dan responsif. Setiap perubahan data segera tercermin pada tampilan tanpa perlu memuat ulang. Uji fungsional menunjukkan stabilitas aplikasi yang mana ringkasan keuangan selalu mutakhir, indikator anggaran mengikuti

realisasi pengeluaran bulan berjalan, dan berkas CSV dapat dibuka di aplikasi lembar kerja tanpa penyesuaian. Dengan fondasi ini, FineMe layak dipakai dan dikembangkan lebih lanjut.

### Saran

Pengembangan aplikasi berikutnya sebaiknya menambahkan sinkronisasi dan pencadangan berbasis awan agar data aman, serta dapat diakses di lebih dari satu perangkat, dilengkapi autentikasi dan pengaturan privasi. Laporan dapat diperluas antar bulan hingga tahunan, tersedia ekspor PDF, serta analitik per kategori seperti tren dan peringatan dini saat pengeluaran mendekati batas. Modul transaksi berulang perlu dibuat lebih fleksibel dengan variasi frekuensi dan riwayat eksekusi. Dari sisi kegunaan, pertimbangkan peningkatan aksesibilitas, dukungan format lokal untuk tanggal dan mata uang, serta impor CSV agar migrasi dari aplikasi lain menjadi lebih mudah.

### E. REFERENSI

- [1] Iramani, R., & Lutfi, L. (2021). An integrated model of financial well-being: The role of financial behavior. *Accounting*, 7(3), 691–700. <https://doi.org/10.5267/j.ac.2020.12.007>
- [2] Khoirunnisa, I. R., & Rochmawati. (2021). Pengaruh literasi keuangan, sikap keuangan, dan pendidikan keuangan keluarga terhadap manajemen keuangan pribadi dengan locus of control sebagai variabel intervening. *Jurnal Pendidikan Akuntansi (JPAK)*, 9(2), 210–219.
- [3] Ratna, S. (2021). Aplikasi pengelola keuangan pribadi berbasis desktop. *Technologia*, 12(2), 68–71. <https://doi.org/10.31602/tji.v12i2.4572>
- [4] Alfonso, G., & Setiawati, P. (2024). Aplikasi manajemen uang kas kelas berbasis mobile dengan framework Flutter. *Journal of Computer Science and Information Technology (JCSIT)*, 2(1), 15–25. <https://doi.org/10.70248/jcsit.v2i1.1298>
- [5] Effendy, F. O., Al Azam, M. N., & Prastoeti, R. (2025). Catatan keuangan UMKM berbasis mobile menggunakan Flutter. *Repeater: Publikasi Teknik Informatika dan Jaringan*, 3(2), 75–88. <https://doi.org/10.62951/repeater.v3i2.416>
- [6] Rozi, M. F., Siregar, H., Hambali, Y. A., & Rasim, R. (2025). Rancang bangun sistem manajemen akademik mahasiswa berbasis mobile multiplatform menggunakan Flutter. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(2), 459–468. <https://doi.org/10.62712/juktisi.v4i2.436>
- [7] Fau, A. (2024). Pelatihan pengenalan dasar framework Flutter dalam pembangunan aplikasi mobil. *Jurnal Pengabdian kepada Masyarakat*, 1(1), 23–28.
- [8] Frendiana, V. (2024). Pemrograman mobile menggunakan Flutter. *Widina Media Utama*.
- [9] Setiawan, P. R., Ramadhan, R. A., & Labellapansa, A. (2022).

Pelatihan pemrograman Flutter.  
Jurnal Pengabdian Masyarakat dan  
Penerapan Ilmu Pengetahuan, 3,  
22–27.

- [10] Pesik, Y. H., Vidya, V., Agustian,  
I. J., & Trisno, I. B. (2022).  
Perancangan dan pembuatan  
aplikasi manajemen acara berbasis  
mobile menggunakan Flutter.  
Jurnal Nasional Komputasi dan  
Teknologi Informasi, 5(6), 989–  
997.  
[https://doi.org/10.32672/jnknti.v5i6.  
5436](https://doi.org/10.32672/jnknti.v5i6.5436)