

# PEMROGRAMAN APLIKASI MOBILE E-WALLET “FULUS” BERBASIS SERVICE EXPRESS JS

Cadrasva Ardhevan Novitza<sup>1</sup>, Arif Hidayat<sup>2</sup>, Ahmad Surya Ramadhan<sup>3</sup>, Daril Pratomo Sriramdanu<sup>4</sup>  
Informatika

Universitas Pembangunan Nasional Veteran Jakarta  
Jalan RS. Fatmawati Raya, Pondok Labu, Cilandak, Jakarta Selatan, Jakarta 12450  
2110511127@mahasiswa.upnvj.ac.id<sup>1</sup>, 2110511157@mahasiswa.upnvj.ac.id<sup>2</sup>,  
2110511160@mahasiswa.upnvj.ac.id<sup>3</sup>, 2110511129@mahasiswa.upnvj.ac.id<sup>4</sup>

**Abstrak.** Seiring dengan pesatnya perkembangan teknologi, kebutuhan akan digitalisasi di berbagai aspek kehidupan semakin meningkat, termasuk dalam bidang finansial dan pembayaran. Aplikasi "Fulus" hadir sebagai solusi untuk mempermudah berbagai transaksi finansial sehari-hari pengguna. Aplikasi ini dirancang untuk memungkinkan pengguna melakukan berbagai jenis transaksi seperti transfer uang, pembayaran tagihan, dan *top up* saldo secara digital. Dalam pengembangan aplikasi ini, metode yang digunakan meliputi analisis produk digital yang ada di pasaran, penerapan sistem pembagian *frontend* dan *backend*, serta integrasi menggunakan *API*. Bagian *frontend* dikembangkan menggunakan *XML* dan *Java* untuk desain antarmuka pengguna dan logika bisnis, sementara bagian *backend* menggunakan *Express.js* untuk pengelolaan data dan komunikasi dengan *database*. Integrasi antara *frontend* dan *backend* melalui *API* memastikan semua fitur aplikasi berfungsi dengan baik dan memberikan pengalaman pengguna yang optimal. Melalui pendekatan ini, aplikasi "Fulus" diharapkan dapat memenuhi kebutuhan finansial pengguna dengan cara yang lebih efisien dan terorganisir.

**Kata Kunci:** Digitalisasi, Aplikasi Finansial, *Frontend*, *Backend*, *API*, Transaksi Digital.

## 1 Pendahuluan

Seiring dengan pesatnya perkembangan teknologi, kebutuhan akan digitalisasi di berbagai aspek kehidupan semakin meningkat, termasuk dalam bidang finansial dan pembayaran. *Financial Technology* (Fintech) telah mengubah cara masyarakat melakukan transaksi keuangan, terutama dengan munculnya uang elektronik dan dompet digital[1]. Aplikasi "Fulus" hadir sebagai solusi untuk mempermudah transaksi finansial sehari-hari dengan menonjolkan desain *UI (User Interface)* yang menarik dan kemampuan untuk melakukan transaksi internasional. Masalah umum seperti keamanan data pribadi, desain *UI* yang kurang menarik, serta keterbatasan penggunaan internasional diatasi melalui pengembangan sistem menggunakan *XML*, *Java*, dan *Express.js*, serta integrasi *API* yang optimal[2].

Selain itu, *backend* aplikasi "Fulus" memanfaatkan *Firestore Realtime Database* yang memungkinkan sinkronisasi data real-time dan layanan otentikasi yang aman. *Firestore* menyediakan berbagai layanan seperti analitik, pesan cloud, dan monitoring kinerja, memastikan data pengguna dapat diakses dengan cepat dan aman, bahkan saat offline[3]. Dengan pendekatan ini, "Fulus" diharapkan menjadi aplikasi dompet digital terdepan yang mengutamakan keamanan, kenyamanan, dan kemudahan akses di berbagai negara, memenuhi kebutuhan finansial pengguna secara efisien dan terorganisir[4].

## 2 Metode Penelitian

Dalam pengembangan aplikasi "Fulus", kami menerapkan beberapa langkah dan metode untuk memastikan aplikasi ini berfungsi dengan baik dan memenuhi kebutuhan pengguna. Berikut adalah metode yang kami gunakan:

### 2.1 Menganalisa Produk Digital:

Langkah pertama dalam pengembangan aplikasi "Fulus" adalah melakukan analisis terhadap produk digital yang ada di pasaran. Analisis ini meliputi penelitian tentang aplikasi-aplikasi finansial yang sudah ada, fitur-fitur yang ditawarkan, serta kekurangan dan kelebihan dari masing-masing aplikasi. Hasil dari analisis ini digunakan sebagai dasar untuk merancang fitur-fitur yang akan diimplementasikan dalam aplikasi "Fulus" agar dapat bersaing di pasar.

## 2.2 Menerapkan Sistem Pembagian Frontend dan Backend:

Untuk memastikan aplikasi "Fulus" dapat berjalan dengan efisien dan terstruktur, kami menerapkan sistem pembagian antara frontend dan backend. Pembagian ini dilakukan agar pengembangan aplikasi dapat dilakukan secara paralel dan lebih terorganisir.

- a. **Frontend:** Bagian frontend aplikasi "Fulus" dikembangkan menggunakan *XML* dan *Java*. *XML* digunakan untuk mendesain tampilan antarmuka pengguna (UI), sedangkan *Java* digunakan untuk mengimplementasikan logika bisnis dan interaksi pengguna dengan aplikasi.
- b. **Backend:** Bagian backend aplikasi "Fulus" dibangun menggunakan *Express.js*, sebuah *framework* untuk *Node.js* yang memudahkan dalam membangun aplikasi web dan *API*. *Backend* ini bertanggung jawab untuk mengelola data, memproses permintaan dari *frontend*, dan berkomunikasi dengan *database*.

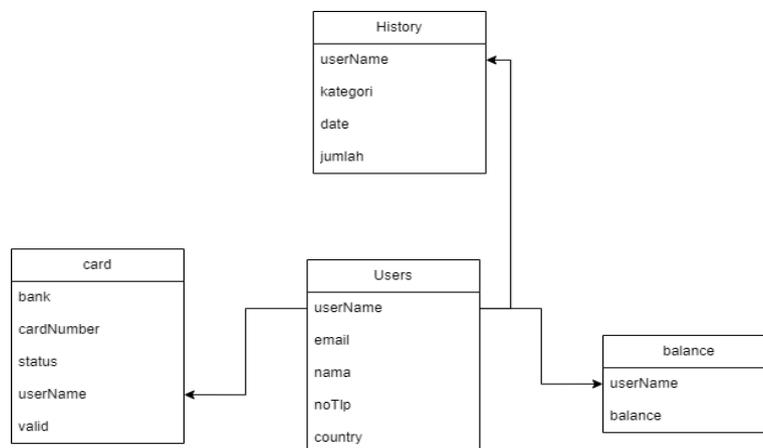
## 2.3 Mengintegrasikan Menggunakan API:

Salah satu komponen kunci dalam pengembangan aplikasi "Fulus" adalah integrasi antara *frontend* dan *backend* menggunakan *API* (*Application Programming Interface*). *API* memungkinkan *frontend* dan *backend* untuk saling berkomunikasi dan bertukar data. Dengan menggunakan *API*, *frontend* dapat mengirim permintaan (*request*) ke *backend* untuk mendapatkan data yang diperlukan, seperti informasi saldo, riwayat transaksi, dan lain-lain. Sebaliknya, *backend* dapat mengirim respon kembali ke *frontend* dengan data yang diminta.

Proses integrasi ini memastikan bahwa semua fitur aplikasi dapat berjalan dengan baik dan memberikan pengalaman pengguna yang optimal. Dengan metode-metode di atas, kami berharap dapat mengembangkan aplikasi "Fulus" yang tidak hanya memenuhi kebutuhan pengguna dalam hal transaksi finansial, tetapi juga memberikan kemudahan dan kenyamanan dalam penggunaannya.

## 3 Hasil dan Pembahasan

Berikut merupakan *ERD* dari aplikasi "Fulus":

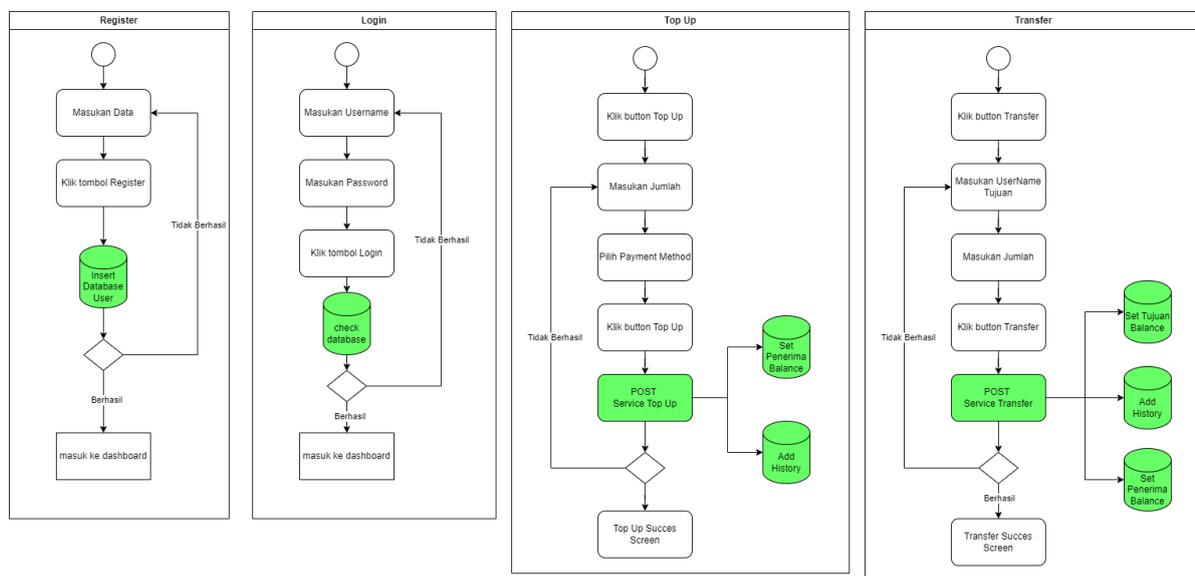


Gambar. 1. ERD Dari Aplikasi "Fulus"

Diagram ERD ini mengilustrasikan hubungan antara beberapa entitas dalam sistem, yaitu *Users*, *Card*, *History*, dan *Balance*. Entitas *Users* menyimpan informasi esensial mengenai pengguna, seperti nama pengguna, email, nama lengkap, nomor telepon, dan negara asal. Setiap pengguna dapat memiliki satu atau lebih kartu, yang berisi informasi seperti nama bank, nomor kartu, status kartu, dan masa berlaku, dengan keterkaitan melalui *userName*.

Selain itu, entitas *History* mencatat transaksi pengguna yang mencakup kategori transaksi, tanggal, dan jumlah, serta dihubungkan dengan pengguna melalui *userName*. Entitas *Balance* mencatat jumlah saldo setiap pengguna dan terhubung melalui *userName*, menunjukkan hubungan satu-ke-satu dengan entitas *Users*.

Secara keseluruhan, *ERD* ini menggambarkan relasi *one-to-many* antara entitas *Users* dengan *Card* dan *History*, serta relasi *one-to-one* antara *Users* dan *Balance*. Diagram ini memberikan gambaran menyeluruh tentang bagaimana data terkait pengguna, kartu, riwayat transaksi, dan saldo terstruktur dan berinteraksi dalam sistem yang dirancang.



Gambar. 2. Flowchart Register, Login, Top Up, Transfer

Berikut adalah penjelasan singkat dari flowchart diatas:

**a. Register**

Pengguna memasukkan data yang diperlukan untuk pendaftaran dan mengklik tombol *register* untuk memulai proses pendaftaran. Sistem kemudian mencoba memasukkan data pengguna ke dalam *database*. Jika data berhasil dimasukkan ke dalam *database*, pengguna akan diarahkan ke *dashboard*. Namun, jika data tidak berhasil dimasukkan, pengguna akan kembali ke langkah awal untuk memasukkan data lagi.

**b. Login**

Pengguna memasukkan *username* dan *password*, kemudian mengklik tombol login untuk memulai proses *login*. Sistem akan memeriksa data pengguna di *database*. Jika data pengguna ditemukan dan *valid*, pengguna akan diarahkan ke *dashboard*. Jika data pengguna tidak *valid* atau tidak ditemukan, pengguna akan kembali ke langkah awal untuk memasukkan *username* dan *password* lagi.

**c. Top Up**

Pengguna mengklik tombol *top up* untuk memulai proses pengisian saldo, kemudian memasukkan jumlah uang yang ingin diisi dan memilih metode pembayaran. Setelah itu, pengguna mengklik tombol *top up* untuk mengonfirmasi pengisian saldo. Sistem kemudian mengirimkan permintaan *top up* ke layanan

terkait. Jika proses *top up* berhasil, sistem akan memperbarui saldo penerima dan menambahkan riwayat transaksi, kemudian pengguna akan melihat layar sukses *top up*. Jika proses *top up* gagal, pengguna akan kembali ke langkah awal untuk memasukkan jumlah dan memilih metode pembayaran lagi.

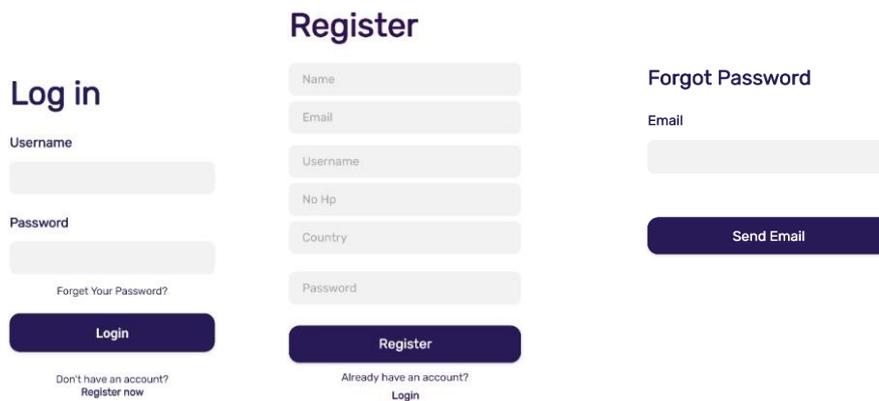
**d. Transfer**

Pengguna mengklik tombol *transfer* untuk memulai proses *transfer* uang, kemudian memasukkan username penerima dan jumlah uang yang ingin *ditransfer*. Setelah itu, pengguna mengklik tombol *transfer* untuk mengkonfirmasi *transfer* uang. Sistem kemudian mengirimkan permintaan *transfer* ke layanan terkait. Jika proses *transfer* berhasil, sistem akan memperbarui saldo tujuan dan saldo penerima, serta menambahkan riwayat transaksi, kemudian pengguna akan melihat layar sukses transfer. Jika proses *transfer* gagal, pengguna akan kembali ke langkah awal untuk memasukkan *username* tujuan dan *jumlah* uang lagi.

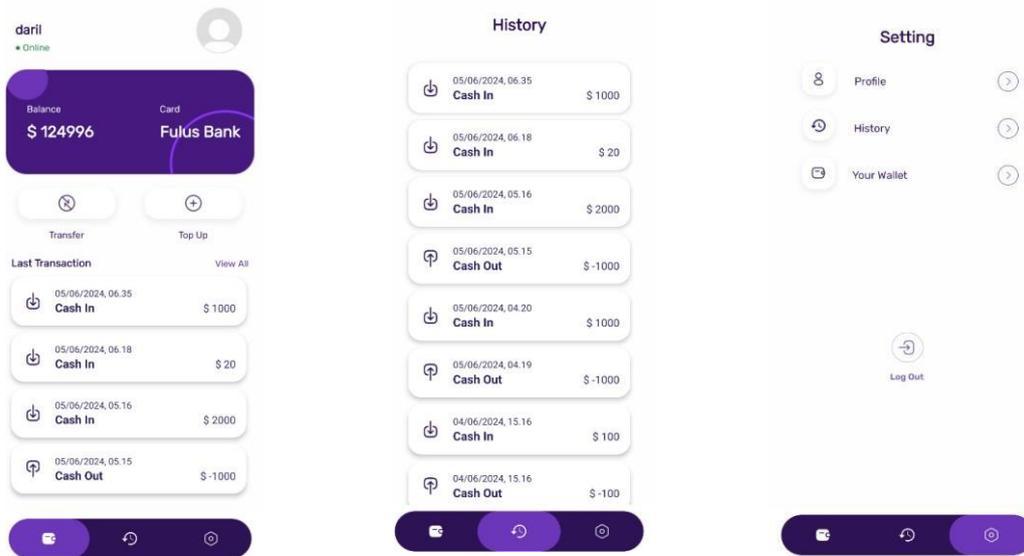
**3.1 Tampilan Aplikasi**



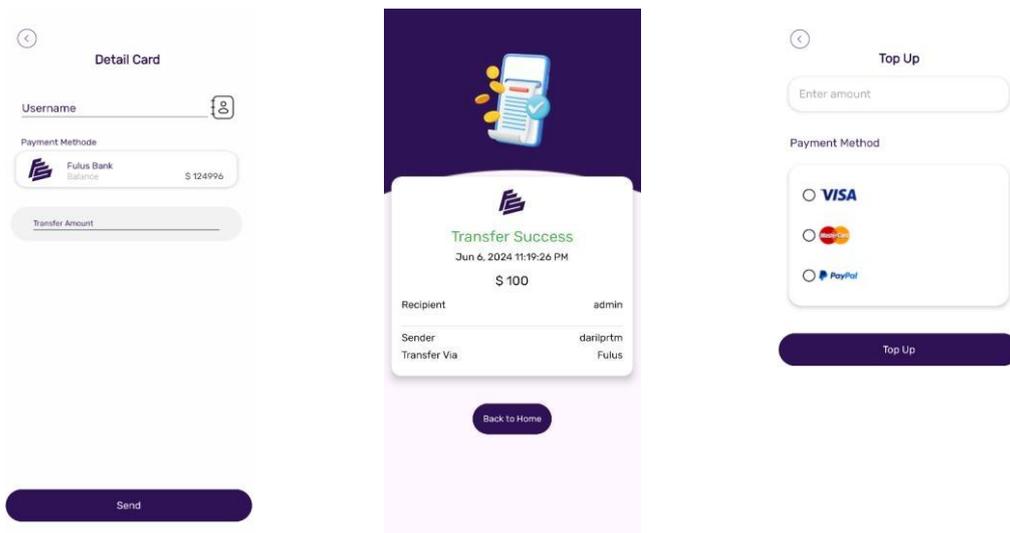
**Gambar. 3.** *Splash Screens*



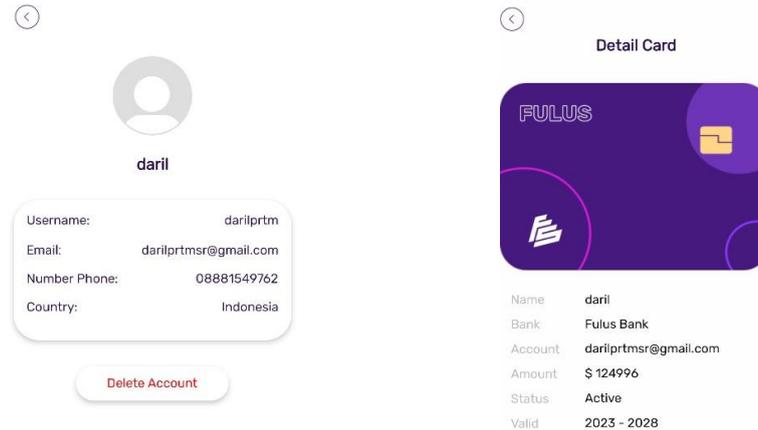
**Gambar. 4.** *Log in, Register, Forgot Password*



**Gambar. 5.** *Dashboard, History, Settings*



**Gambar. 6.** *Transfer, Transfer Information, Top Up*



Gambar. 7. Profile and Card Detail Page

- a. **Splash Screens:** Menampilkan layar awal aplikasi dengan logo dan nama "Fulus," memberikan kesan pertama yang menarik saat aplikasi dibuka. Layar ini biasanya ditampilkan selama beberapa detik sebelum pengguna diarahkan ke halaman *login* atau beranda, memberikan waktu bagi aplikasi untuk memuat konten utama di latar belakang.
- b. **Login Page:** Halaman untuk pengguna yang sudah terdaftar memasukkan email dan kata sandi untuk mengakses akun mereka. Halaman ini juga biasanya dilengkapi dengan opsi untuk mengingat pengguna agar tidak perlu memasukkan kredensial setiap kali membuka aplikasi, serta tautan untuk mengakses halaman lupa kata sandi jika diperlukan.
- c. **Register Page:** Halaman pendaftaran yang memungkinkan pengguna baru untuk membuat akun dengan mengisi informasi seperti nama, email, nomor telepon, dan kata sandi. Halaman ini memastikan bahwa semua informasi yang diperlukan untuk membuat akun yang aman dan personal telah disediakan. Terdapat juga verifikasi email atau nomor telepon untuk keamanan tambahan.
- d. **Forgot Password Page:** Halaman yang membantu pengguna memulihkan akses ke akun mereka jika lupa kata sandi, dengan memasukkan email untuk menerima instruksi pemulihan. Proses ini sering melibatkan pengiriman tautan reset kata sandi ke email pengguna, memastikan keamanan dan kemudahan dalam memulihkan akses.
- e. **Home Page:** Halaman utama yang memberikan akses cepat ke fitur-fitur utama seperti saldo, transfer, dan pembayaran. Halaman ini menampilkan saldo pengguna secara jelas dan menyediakan menu navigasi untuk mengakses fungsi lain seperti histori transaksi, pengaturan akun, dan lainnya. Desain yang intuitif dan menarik memastikan pengguna dapat dengan mudah menemukan dan menggunakan fitur yang mereka butuhkan.
- f. **History Page:** Menampilkan riwayat transaksi pengguna, memungkinkan mereka untuk melihat detail transaksi sebelumnya secara kronologis. Setiap entri biasanya mencakup informasi seperti tanggal, jumlah transaksi, penerima atau pengirim, dan status transaksi, membantu pengguna dalam melacak aktivitas keuangan mereka.
- g. **Settings Page:** Halaman pengaturan dimana pengguna dapat mengubah preferensi akun mereka, termasuk pengaturan keamanan seperti penggantian kata sandi, pengaturan notifikasi untuk menerima pembaruan

dan alert, serta informasi pribadi yang dapat diperbarui sesuai kebutuhan. Pengguna juga dapat menyesuaikan preferensi lainnya untuk meningkatkan pengalaman penggunaan aplikasi.

- h. **Transaction Page:** Halaman yang menunjukkan detail transaksi tertentu, termasuk informasi penerima, jumlah uang yang ditransfer, dan status transaksi. Halaman ini menyediakan rincian lengkap dari setiap transaksi, memungkinkan pengguna untuk mengkonfirmasi dan mereview transaksi yang telah dilakukan atau yang sedang dalam proses.
- i. **Status Page:** Halaman yang memperlihatkan status dari berbagai transaksi pengguna, apakah berhasil, pending, atau gagal. Ini membantu pengguna untuk segera mengetahui kondisi terkini dari transaksi mereka dan mengambil tindakan jika diperlukan, seperti menghubungi dukungan pelanggan jika ada transaksi yang gagal.
- j. **Methods Page:** Halaman yang menampilkan berbagai metode pembayaran yang tersedia dalam aplikasi "Fulus." Pengguna dapat memilih dan mengatur metode pembayaran yang mereka inginkan, seperti kartu kredit, debit, atau e-wallet lainnya. Halaman ini memastikan fleksibilitas dan kemudahan dalam memilih metode pembayaran yang paling sesuai untuk pengguna.
- k. **Profile Page:** Halaman yang menampilkan informasi pribadi pengguna, memungkinkan mereka untuk mengedit detail seperti foto profil, nama, dan kontak. Halaman profil ini juga sering kali menyediakan informasi tentang akun seperti tingkat keanggotaan atau status verifikasi, memberi pengguna kendali penuh atas informasi pribadi mereka.
- l. **Card Detail Page:** Halaman yang menampilkan rincian informasi kartu yang telah ditambahkan ke aplikasi, termasuk status dan histori penggunaan. Pengguna dapat melihat informasi seperti nomor kartu, tanggal kadaluarsa, dan transaksi yang dilakukan dengan kartu tersebut, serta mengelola kartu mereka dengan menambah atau menghapus sesuai kebutuhan.

## 4 Kesimpulan

Berdasarkan penelitian dan pengembangan yang telah dilakukan, aplikasi "Fulus" terbukti menjadi solusi efektif dalam mempermudah berbagai transaksi finansial sehari-hari pengguna. Melalui analisis produk digital yang ada di pasaran, aplikasi ini dirancang dengan mempertimbangkan kelebihan dan kekurangan aplikasi finansial yang sudah ada. Pendekatan pembagian frontend dan backend yang diterapkan, serta integrasi menggunakan *API*, memungkinkan aplikasi "Fulus" untuk berfungsi secara optimal dengan pengalaman pengguna yang intuitif dan efisien.

Penggunaan *XML* dan *Java* untuk pengembangan frontend serta *Express.js* untuk backend telah terbukti efektif dalam mengelola data dan komunikasi antar bagian sistem. Selain itu, implementasi *Firebase Realtime Database* memastikan sinkronisasi data real-time dan keamanan yang tinggi, memberikan kenyamanan dan keandalan bagi pengguna dalam melakukan transaksi, bahkan saat *offline*. *ERD* dan *flowchart* yang dikembangkan memberikan gambaran jelas tentang struktur data dan alur proses utama dalam aplikasi, seperti registrasi, *login*, *top up*, dan transfer uang.

Dengan demikian, aplikasi "Fulus" diharapkan dapat memenuhi kebutuhan finansial pengguna dengan lebih efisien dan terorganisir, serta mampu bersaing di pasar aplikasi dompet digital yang semakin kompetitif. Ke depan, pengembangan lebih lanjut dan peningkatan fitur dapat terus dilakukan untuk menjaga relevansi dan keunggulan aplikasi ini di tengah perkembangan teknologi yang pesat.

**Referensi**

- [1] I. Kumala and I. Mutia, "Pemanfaatan Aplikasi DOMPET Digital Terhadap Transaksi Retail Mahasiswa," *Semin. Nas. Ris. dan Teknol. (SEMNAS RISTEK)*, pp. 64–69, 2020.
- [2] B. K. Lacky and R. Malfiany, "Analisis Tingkat Penerimaan Pengguna Ovo Di Arawang Menggunakan Pendekatan Technology Acceptance Model (TAM)," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 15, no. 4, pp. 48–56, 2021, doi: 10.35969/interkom.v15i4.81.
- [3] A. T. Gaikwad, "FIREBASE -OVERVIEW AND USAGE FIREBASE - OVERVIEW AND USAGE," no. August, [4] 2022.
- [5] R. USMAN, "Karakteristik Uang Elektronik Dalam Sistem Pembayaran," *Yuridika*, vol. 32, no. 1, p. 134, 2017, doi: 10.20473/ydk.v32i1.4431.
- [6] R. Sulistyowati, L. Paais, and R. Rina, "Persepsi Konsumen Terhadap Penggunaan Dompot Digital," *ISOQUANT J. Ekon. Manaj. dan Akunt.*, vol. 4, no. 1, p. 17, 2020, doi: 10.24269/iso.v4i1.323.
- [7] R. I. Permana, "Analisis Faktor Penggunaan Dompot Digital di Kalangan Mahasiswa Perguruan Tinggi Surabaya,"
- [8] *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 1, pp. 312–322, 2021, doi: 10.35957/jatisi.v8i1.584.
- [9] M. S. Alif and A. R. Pratama, "Analisis Kesadaran Keamanan di Kalangan Pengguna E-Wallet di Indonesia," *J. Inf.*, vol. 2, no. 1, pp. 1–7, 2021, [Online]. Available: <https://journal.uii.ac.id/AUTOMATA/article/view/17279>.