

Implementasi Progressive Web App (PWA) untuk Aplikasi Manajemen Keuangan Pribadi dengan Kemampuan Offline-First

Gheri Maulana Tri Istidar¹, Agus Prihanto²

^{1,2} Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹gheri.21036@mhs.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak— Ketergantungan aplikasi manajemen keuangan pada koneksi internet dapat menunda pencatatan transaksi dan memengaruhi ketelitian data ketika jaringan tidak stabil. Penelitian ini bertujuan mengimplementasikan BudgetBonsai, aplikasi manajemen keuangan pribadi berbasis Progressive Web App (PWA) dengan arsitektur offline-first. Pengembangan menerapkan Rapid Application Development (RAD) yang mencakup pemodelan bisnis dan data, pemodelan proses, pembangkitan aplikasi, serta pengujian. Antarmuka dibangun menggunakan React dan Vite, Service Worker mengelola cache aset, dan IndexedDB melalui Dexie.js menjadi penyimpanan utama untuk operasi lokal. Firebase Authentication dan Cloud Firestore digunakan untuk autentikasi serta sinkronisasi data manual saat koneksi tersedia. Pengujian Black Box mencakup 17 skenario pada modul autentikasi, transaksi, kategori, anggaran, tujuan keuangan, ekspor laporan, dan sinkronisasi; seluruh skenario dinyatakan berhasil. Pengujian teknis memperlihatkan aplikasi memenuhi kriteria instalasi PWA pada perangkat seluler dan desktop, Service Worker aktif, serta pemuatan ulang pada kondisi luring mengembalikan HTTP 200. Data yang dibuat saat luring berhasil dikirim ke Firestore tanpa kehilangan data. Hasil menunjukkan kombinasi PWA, cache, dan IndexedDB memungkinkan fungsi utama aplikasi tetap tersedia tanpa ketergantungan jaringan.

Kata Kunci— Progressive Web App, offline-first, manajemen keuangan, IndexedDB, Firebase, Rapid Application Development.

I. PENDAHULUAN

Pencatatan pemasukan dan pengeluaran secara konsisten diperlukan agar pengguna dapat memantau arus kas, menetapkan anggaran, dan mengelola tujuan keuangan. Namun, aplikasi yang mengutamakan layanan peladen (*server*) dapat menghambat pencatatan ketika jaringan lemah atau tidak tersedia. Penelitian sebelumnya telah menerapkan PWA untuk pengelolaan keuangan [4], sedangkan aplikasi Android lokal menunjukkan pentingnya ketersediaan pencatatan tanpa koneksi [5]. Kebutuhan tersebut mendorong rancangan aplikasi yang tetap dapat menerima dan menyimpan transaksi pada perangkat pengguna.

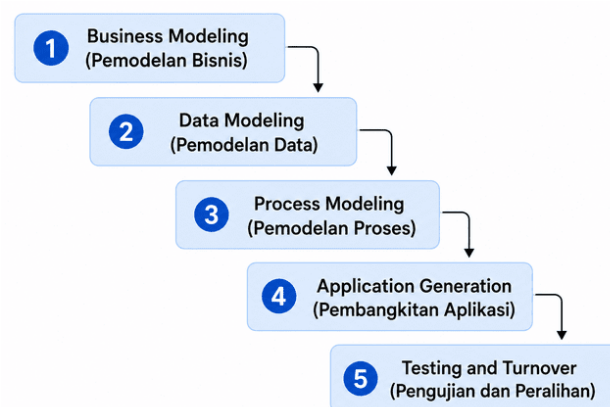
PWA memadukan aksesibilitas aplikasi web dengan kemampuan instalasi serta penggunaan cache melalui Service Worker. Kajian praktik PWA menunjukkan bahwa teknologi ini dapat meningkatkan keterandalan aplikasi web, meskipun strategi cache dan sinkronisasi data harus dirancang sesuai kebutuhan aplikasi [1], [2]. Pendekatan offline-first menempatkan basis data lokal sebagai titik utama operasi

sehingga koneksi diperlakukan sebagai pelengkap untuk pertukaran data [3].

Penelitian ini mengimplementasikan BudgetBonsai, aplikasi manajemen keuangan pribadi yang menyediakan autentikasi, pencatatan transaksi, manajemen kategori, anggaran, tujuan keuangan, ekspor laporan, dan sinkronisasi data manual. Kontribusi utama penelitian adalah penerapan PWA offline-first menggunakan Service Worker dan IndexedDB, serta validasi fungsi utama, kemampuan instalasi, pemuatan luring, dan sinkronisasi ke Cloud Firestore.

II. METODE PENELITIAN

Pengembangan sistem menggunakan Rapid Application Development (RAD). Model ini dipilih karena mendukung siklus iteratif dan pembentukan prototipe pada ruang lingkup kebutuhan yang terdefinisi [6]. Tahapan yang diterapkan meliputi Business Modeling, Data Modeling, Process Modeling, Application Generation, serta Testing and Turnover.



Gbr. 1 Tahapan Pengembangan Aplikasi dengan Metode RAD

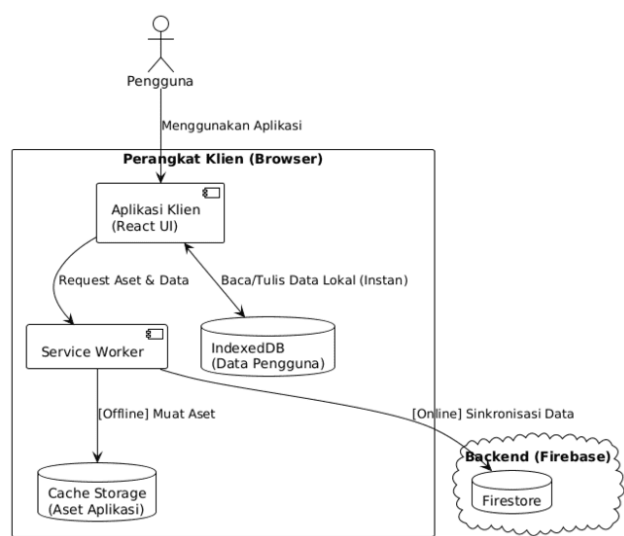
A. Rancangan Arsitektur

Pada kondisi normal maupun luring, React dan Vite menyediakan antarmuka aplikasi, sedangkan Service Worker menyimpan app shell di Cache Storage dan menangani navigasi lokal. Seluruh operasi CRUD diarahkan ke Dexie.js/IndexedDB agar data diproses langsung pada perangkat. Firebase Authentication digunakan untuk autentikasi; Cloud Firestore menjadi tujuan sinkronisasi manual saat jaringan tersedia. Setiap data memuat metadata

waktu pembaruan untuk menjaga urutan perubahan saat sinkronisasi.

TABEL I
KOMPONEN ARSITEKTUR SISTEM

Komponen	Peran
React dan Vite	Membangun antarmuka pengguna serta proses build aplikasi
Service Worker (WorkBox)	Menyimpan aset aplikasi dan menangani navigasi saat luring
Dexie.js dan IndexedDB	Menyimpan data lokal dan menjalankan operasi CRUD tanpa jaringan
Firebase Authentication dan Firestore	Mengelola autentikasi serta sinkronisasi data manual ke server



Gbr. 2 Arsitektur Sistem PWA Offline-First

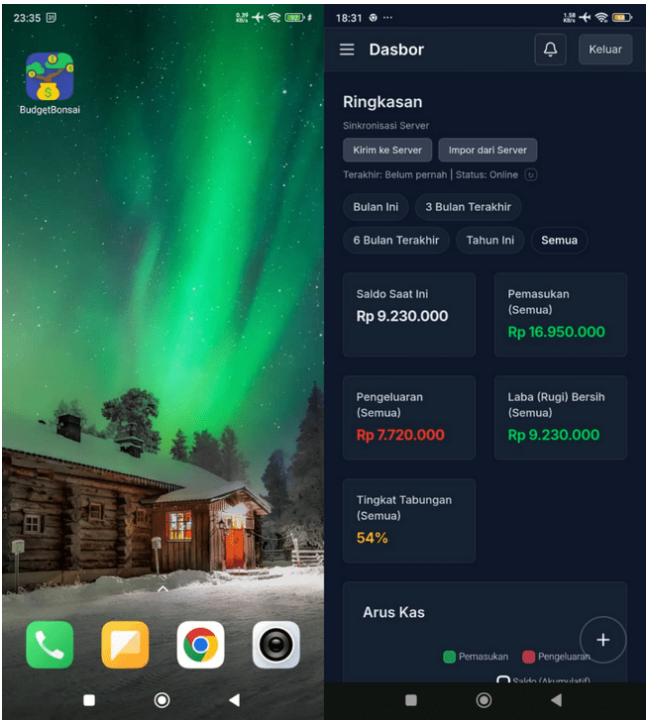
B. Pengujian Sistem

Validasi fungsional menggunakan Black Box Testing pada 17 skenario. Skenario mencakup registrasi dan login, pengelolaan transaksi serta kategori, anggaran, tujuan keuangan, ekspor PDF/XLSX, dan sinkronisasi data. Pengujian nonfungsional memeriksa kelayakan instalasi PWA pada peramban seluler dan desktop, status Service Worker, respons aplikasi ketika jaringan diputus, serta keberhasilan pengiriman data IndexedDB ke Firestore.

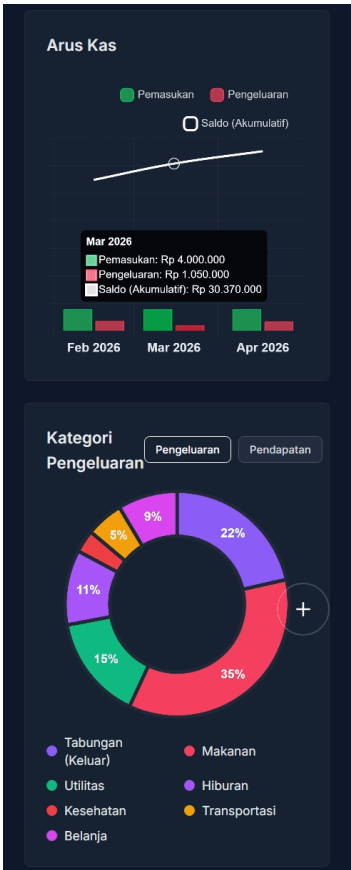
III. HASIL DAN PEMBAHASAN

A. Implementasi Aplikasi

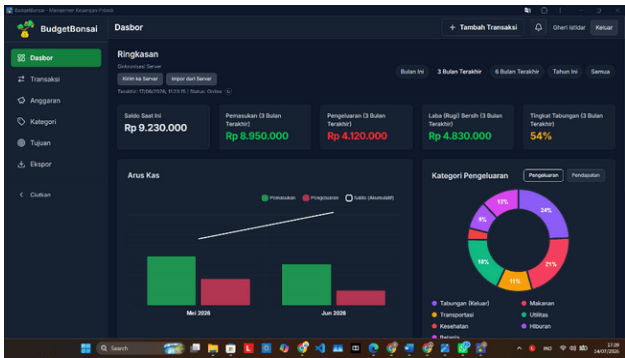
BudgetBonsai berhasil diimplementasikan sebagai aplikasi web yang dapat dipasang. Setelah pengguna masuk, dasbor menampilkan ringkasan saldo, pemasukan, pengeluaran, dan kontrol sinkronisasi. Modul transaksi mendukung penambahan, perubahan, dan penghapusan catatan pemasukan maupun pengeluaran. Pengguna juga dapat mengelola kategori, menetapkan anggaran, mencatat progres tujuan keuangan, dan mengekspor rekap data ke format PDF atau XLSX



Gbr. 3 Tampilan Ikon Aplikasi pada Layar Utama Mobile dan Tampilan Dasbor Aplikasi BudgetBonsai



Gbr. 4 Tampilan Dasbor Aplikasi BudgetBonsai Bagian Bawah



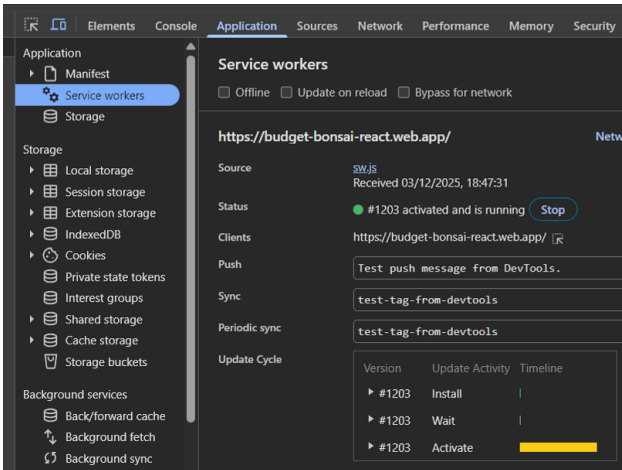
Gbr. 5 Tampilan Aplikasi di Desktop

Implementasi offline-first tidak mengirim mutasi data ke peladen (*server*) pada setiap interaksi pengguna. Data transaksi, kategori, anggaran, dan tujuan disimpan terlebih dahulu pada IndexedDB. Service Worker yang aktif menyediakan aset antarmuka dari cache, sehingga aplikasi tetap dapat dimuat ketika koneksi diputus. Sinkronisasi dilakukan secara sadar oleh pengguna melalui tombol pengiriman data setelah jaringan tersedia. Pola ini selaras dengan pendekatan PWA offline-first yang mengutamakan penanganan data dan sinkronisasi secara efisien [3].

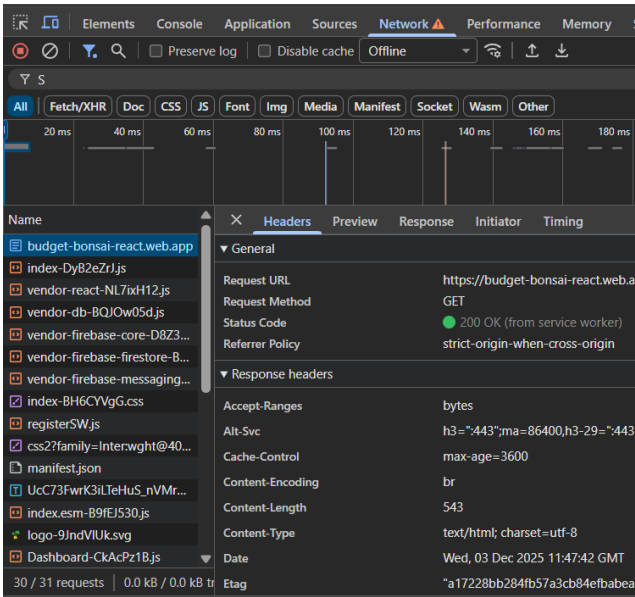
B. Hasil Pengujian

TABEL II
RINGKASAN HASIL PENGUJIAN SISTEM

Aspek	Skenario/Indikator	Hasil
Fungsional	17 skenario Black Box pada tujuh modul utama	Seluruh skenario berhasil
Instalasi PWA	Manifest, Service Worker, dan instalasi pada perangkat seluler serta desktop	Aplikasi dapat dipasang dan berjalan standalone
Kapabilitas Luring	Service Worker aktif; halaman dimuat ulang saat jaringan diputus	Respons HTTP 200; fungsi utama tetap berjalan
Sinkronisasi Data	Data lokal dikirim ke Firebase Firestore saat perangkat kembali daring	Data termuat dengan struktur sesuai tanpa kehilangan data

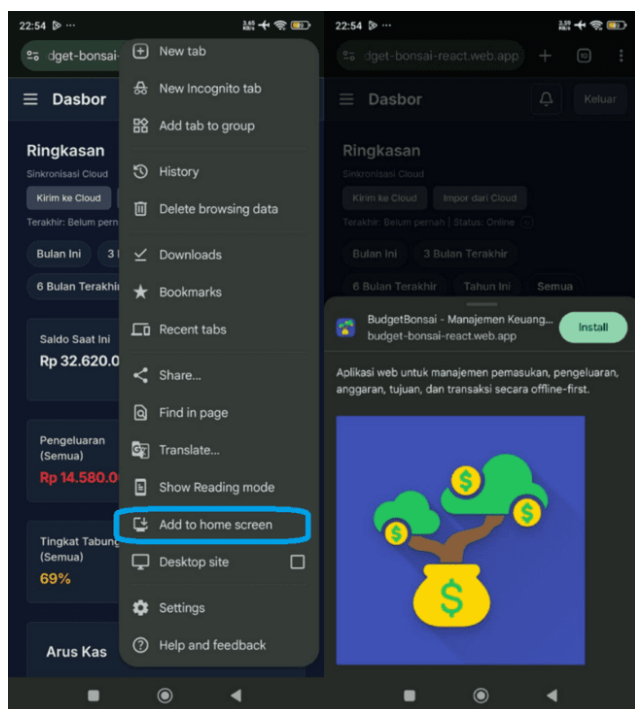


Gbr. 6 Status Service Worker Aktif pada Chrome DevTools



Gbr. 7 Respons Kode HTTP 200 saat Kondisi Luring

Hasil Black Box menunjukkan fungsi yang diuji dapat berjalan pada kondisi daring maupun luring sesuai skenario. Pada perangkat seluler, peramban menampilkan prompt instalasi ketika berkas manifest dan Service Worker tersedia. Pada perangkat desktop, ikon instalasi muncul pada bilah alamat dan aplikasi dapat dibuka melalui jendela standalone. Ketika koneksi disimulasikan luring, Service Worker berstatus aktif dan pemuatan ulang aplikasi menghasilkan HTTP 200 karena aset antarmuka diperoleh dari Cache Storage.

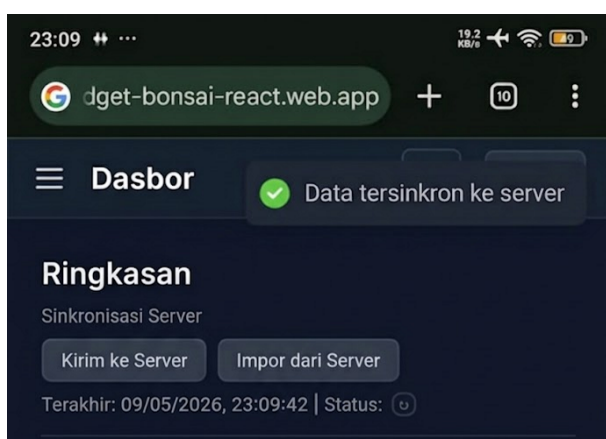


Gbr. 8 Tampilan Prompt Instalasi pada Perangkat Mobile

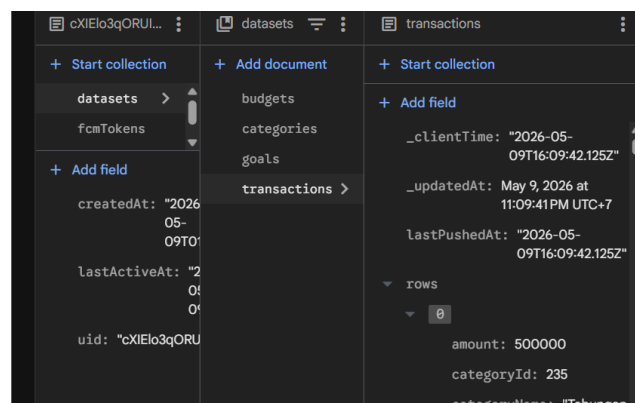


Gbr. 9 Tampilan Ikon Instalasi pada Bilah Alamat Desktop

Sinkronisasi manual berhasil membaca data dari IndexedDB, memetakan data ke koleksi Firestore pengguna, lalu memberikan notifikasi keberhasilan. Verifikasi pada Firestore memperlihatkan data masuk tanpa redudansi atau kehilangan data. Pemisahan operasi lokal dan proses sinkronisasi ini membuat pencatatan tetap responsif saat pengguna berada di area dengan koneksi terbatas.



Gbr. 10 Notifikasi Keberhasilan Sinkronisasi pada Antarmuka Aplikasi



Gbr. 11 Tampilan Data Masuk pada Basis Data Firebase Firestore

IV. KESIMPULAN

BudgetBonsai berhasil diimplementasikan sebagai aplikasi manajemen keuangan pribadi berbasis PWA dengan arsitektur offline-first. Aplikasi dapat dipasang pada peramban seluler maupun desktop dan menyediakan modul autentikasi, transaksi, kategori, anggaran, tujuan keuangan, ekspor laporan, serta sinkronisasi data.

Service Worker dan Cache Storage memungkinkan antarmuka tetap dimuat pada kondisi luring, sedangkan Dexie.js/IndexedDB memastikan operasi data utama tetap berjalan tanpa koneksi. Seluruh 17 skenario Black Box berhasil; pengujian teknis juga memvalidasi status Service Worker aktif, respons HTTP 200 pada kondisi luring, serta sinkronisasi manual ke Firestore tanpa kehilangan data.

Pengembangan berikutnya dapat menerapkan Background Sync untuk otomatisasi pertukaran data dan algoritma resolusi konflik berbasis timestamp bagi penggunaan multigawai. Evaluasi kegunaan oleh pengguna juga diperlukan agar kualitas pengalaman pengguna dapat diukur secara terpisah dari validasi fungsional sistem.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, hidayah, dan karunia-Nya, sehingga penelitian dan penyusunan artikel ilmiah ini dapat diselesaikan dengan baik. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada dosen pembimbing atas segala arahan, bimbingan, dan kesabaran yang diberikan selama proses penelitian. Ucapan terima kasih juga penulis tujuhan kepada orang tua dan keluarga tercinta atas dukungan moral, doa, dan materiil yang tiada henti. Tidak lupa, penulis berterima kasih kepada Jurusan Teknik Informatika Universitas Negeri Surabaya serta rekan-rekan sejawat yang telah memberikan bantuan dan motivasi dalam penyelesaian artikel ini.

REFERENSI

- [1] M. Al Hamid, I. Nuryasin, dan Z. Sari, "Penerapan Progressive Web Application Pada Website Online Public Access Catalog (OPAC) UMM," REPOSITOR, vol. 4, no. 2, pp. 125-136, 2022, doi: 10.22219/repositor.v4i2.31087.
- [2] R. Fauzan, I. Krisnahati, B. D. Nurwibowo, dan D. A. Wibowo, "A Systematic Literature Review on Progressive Web Application Practice

- and Challenges," IPTEK The Journal for Technology and Science, vol. 33, no. 1, p. 43, 2022, doi: 10.12962/j20882033.v33i1.13904.
- [3] K. Sirigiri, A. Narra, dan A. Sirumalla, "Offline-first PWA: Case Study on Efficient Data Handling and Synchronization," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 9, no. 3, pp. 747-754, 2023, doi: 10.32628/cseit25112782.
- [4] A. Prayitno dan E. Hariyanto, "Perancangan Aplikasi Pengelolaan Keuangan Menggunakan Metode Progressive Web Apps (Studi Kasus: SDIT Zahra Asy Syifa Patumbak Deli Serdang)," Bulletin of Information Technology (BIT), vol. 4, no. 1, pp. 9-10, 2023, doi: 10.47065/bit.v3i1.
- [5] U. Juhardi dan Khairullah, "Sistem Pencatatan dan Pengolahan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis Android," Jurnal JTIS (Jurnal Teknik Informatika dan Sistem), vol. 2, no. 1, pp. 24-29, 2019, doi: 10.36085/jtis.v2i1.215.
- [6] S. Aswati, M. S. Ramadhan, A. U. Firmansyah, dan K. Anwar, "Studi Analisis Model Rapid Application Development dalam Pengembangan Sistem Informasi," Matrik Jurnal Manajemen Teknik Informatika dan Rekayasa Komputer, vol. 16, no. 2, pp. 20-27, 2017, doi: 10.30812/matrik.v16i2.10.