

Analisis Performa Teknologi Progressive Web App pada Aplikasi Online Notebook

Rizka Salim Arjanu¹, I Made Suartana²

^{1,2} Teknik Informatika/S-1 Teknik Informatika, Universitas Negeri Surabaya

¹rizka.21044@mhs.unesa.ac.id

²madesuartana@unesa.ac.id

Abstrak— Perkembangan aplikasi web menuntut performa tinggi dan dukungan akses offline. Progressive Web App (PWA) menawarkan solusi dengan menggabungkan keunggulan aplikasi web dan native. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi notebook online berbasis PWA yang dapat berjalan secara offline, serta membandingkan performanya dengan versi non-PWA. Metode yang digunakan adalah pengembangan dua versi aplikasi menggunakan Next.js dan Firebase. Pengujian dilakukan dengan dua tools yaitu, Lighthouse dan GTMetrix untuk mengukur beberapa metrik seperti waktu muat awal, efisiensi cache, dan interaktivitas, pada perangkat PC dan mobile. Hasil menunjukkan bahwa versi PWA lebih unggul, dengan waktu muat rata-rata 0,39 detik dibandingkan 0,98 detik pada non-PWA. Di samping pengujian terhadap performa, penelitian ini juga mencakup pengujian fungsionalitas dan kompatibilitas dalam berbagai kondisi konektivitas. Dengan demikian, penerapan PWA menghasilkan peningkatan performa yang terukur, seperti penurunan waktu muat awal dari 0,98 detik pada non-PWA menjadi 0,39 detik pada versi PWA, serta waktu muat penuh yang lebih cepat sebesar 440 ms dibandingkan 1,1 detik. Selain itu, versi PWA menunjukkan stabilitas akses offline dan kompatibilitas optimal di berbagai perangkat. Saran pengembangan selanjutnya adalah optimalisasi strategi cache dan perluasan pengujian pada lingkungan pengguna yang lebih beragam.

Kata Kunci— *Progressive Web App, Notebook Online, Lighthouse, PWA Offline, Online Access.*

I. PENDAHULUAN

Perkembangan teknologi telah memberikan kemudahan dalam berbagai aspek kehidupan, termasuk dalam aktivitas mencatat. Aplikasi catatan kini mudah diakses melalui perangkat mobile yang telah menjadi bagian penting dari kehidupan sehari-hari. Namun, aplikasi catatan konvensional seringkali memiliki keterbatasan, seperti data yang hanya tersimpan secara lokal, menyulitkan akses dari perangkat lain. Teknologi web mulai menjawab keterbatasan ini dengan menyediakan layanan menyerupai aplikasi native, seperti penyimpanan data di server dan akses melalui browser [1]. Hal ini menjadikan web sebagai solusi praktis untuk kebutuhan aksesibilitas dan fleksibilitas dalam pengelolaan catatan digital.

Web service tidak mungkin ada tanpa didahului oleh kemunculan teknologi web itu sendiri. Penemu situs web yaitu Sir Timothy John "Tim" Berners-Lee, sedangkan situs web yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Tujuan dari Tim ketika merancang situs web adalah untuk memudahkan tukar menukar dan memperbarui informasi pada sesama peneliti di tempat ia bekerja [2].

WWW memungkinkan pengguna mengakses dokumen yang saling terhubung melalui protokol HTTP. Pada tahap awalnya, web bersifat statis, namun menjadi fondasi bagi perkembangan teknologi web yang lebih kompleks dan dinamis.

Memasuki awal tahun 2000-an, web mengalami transisi besar ke era Web 2.0, yang menekankan interaktivitas dan kolaborasi. Istilah Web 2.0 mulai muncul ke permukaan ketika Tim O'Reilly dari O'Reilly Media mengadakan konferensi Web 2.0 pertama kalinya di tahun 2004 [3]. Web tidak lagi hanya menjadi sarana konsumsi informasi, tetapi juga memungkinkan pengguna menghasilkan konten. Contoh kriteria web 2.0 adalah Share content yang cepat dan efisien, seperti YouTube, dimana member dapat membuat video dan mengunggahnya, lalu orang lain dapat melihat video tersebut dalam waktu yang relatif cepat setelah pengunggahan video (dapat kurang dari 1 jam) [3]. Dengan teknologi seperti AJAX dan HTML5, web menjadi lebih responsif dan interaktif, serta semakin populer di kalangan pengguna [4].

Sadar akan web 2.0 yang memiliki banyak pengguna, pakar internet meluncurkan web 3.0. Konsep Web 3.0 pertama kali diperkenalkan pada tahun 2001, yaitu saat Tim Berners-Lee, penemu World Wide Web, menulis sebuah artikel ilmiah yang menggambarkan Web 3.0 sebagai sebuah sarana bagi mesin untuk membaca halaman Web. Web 3.0 hadir dengan dukungan API. API merupakan sebuah interface yang diimplementasikan dengan menggunakan software sehingga software tersebut dapat berinteraksi dengan software lain, seperti halnya user interface yang memungkinkan user untuk berinteraksi dengan computer [3].

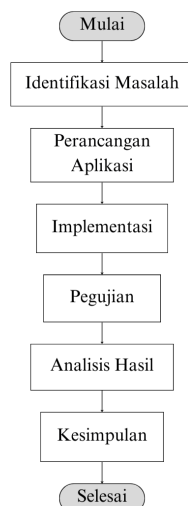
Meskipun telah berkembang, aplikasi web modern masih mengalami kendala dalam menyamai kualitas pengalaman pengguna yang ditawarkan oleh aplikasi native. Ketergantungan pada koneksi internet, performa yang inkonsisten, serta fitur yang terbatas menjadi hambatan. Pengguna memerlukan aplikasi yang cepat, responsif, dan akses dengan atau tanpa koneksi internet. Oleh karena itu, pengembang mulai mencari pendekatan yang menggabungkan fleksibilitas web dengan keunggulan aplikasi native.

Sebagai jawaban atas tantangan tersebut, Google memperkenalkan Progressive Web Apps (PWA) pada tahun 2015 dan mendorong pemilik situs untuk mengadopsinya. Fitur seperti instalasi instan, pengalaman offline, dan notifikasi pengguna menarik minat banyak pemilik situs. Contohnya, AliExpress dan Flipkart melaporkan bahwa penerapan PWA meningkatkan tingkat konversi dan durasi belanja pelanggan mereka secara signifikan [5]. PWA memungkinkan aplikasi web memiliki fitur seperti offline

mode, push notification, dan performa yang cepat melalui teknologi seperti caching dan service workers [6]. Pembuatan aplikasi menggunakan mobile web app dengan PWA juga cenderung lebih ringan serta dapat diakses lintas perangkat. Untuk memberikan pengalaman menyerupai aplikasi native mobile, aplikasi web harus menggunakan Service Worker, yang memungkinkan akses offline, pekerjaan latar belakang, notifikasi, dan instalasi [7]. Dengan PWA, pengguna dapat merasakan pengalaman seperti menggunakan aplikasi native tanpa harus mengunduh dari app store.

Penelitian ini bertujuan untuk menjawab permasalahan keterbatasan aplikasi catatan konvensional dengan mengembangkan prototipe aplikasi catatan online berbasis PWA. Penulis akan mengimplementasikan dan menguji performa PWA pada aplikasi pencatat online, termasuk skenario penggunaan offline. Diharapkan hasil dari penelitian ini mampu menjadi pijakan dalam pengembangan aplikasi yang memiliki tingkat kompatibilitas tinggi, efisiensi optimal, dan kinerja ringan, sekaligus membuka peluang sebagai pendekatan alternatif dalam evolusi pengembangan aplikasi mobile di masa mendatang.

II. METODE PENELITIAN



Gbr. 1 Struktur Langkah-langkah Penelitian

Pada Gbr. 1 Struktur Langkah-langkah Penelitian, diagram menggambarkan tahapan penelitian yang dimulai dari identifikasi masalah, perancangan aplikasi, implementasi, pengujian, analisis hasil, dan kesimpulan. Setiap langkah bertujuan akhir untuk mendapatkan data perbandingan kedua aplikasi.

A. Identifikasi Masalah

Langkah awal dalam penelitian ini adalah mengidentifikasi permasalahan dari penggunaan aplikasi catatan digital yang umum digunakan saat ini. Masalah utama yang ditemukan adalah keterbatasan akses di berbagai perangkat, ketergantungan pada koneksi internet, serta kurangnya performa dan fitur yang menyerupai aplikasi native. Sebagai respon terhadap kebutuhan akan akses yang fleksibel, efisiensi

sistem, dan kualitas interaksi pengguna yang optimal, teknologi Progressive Web App (PWA) diadopsi sebagai pendekatan strategis yang mampu menggabungkan keunggulan aplikasi web dan native dalam satu platform.

B. Perancangan Aplikasi

Aplikasi yang dikembangkan adalah notebook online berbasis web, dengan fitur utama berupa pembuatan, pengeditan, dan penyimpanan catatan teks. Arsitektur aplikasi mencakup:

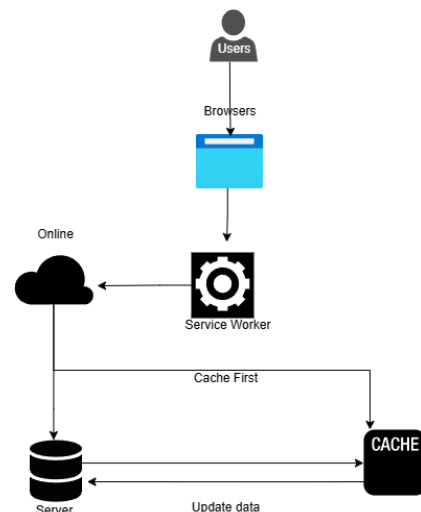
- **Frontend** dibangun menggunakan **React.js** dengan pendekatan komponen.
- **Backend** menggunakan **Firebase Firestore** untuk penyimpanan data dan **Firebase Authentication** untuk autentikasi pengguna via akun Google.
- **UI/UX** dirancang berbasis wireframe sederhana yang mendukung penggunaan di perangkat mobile dan desktop.

C. Integrasi Progressive Web App (PWA)

Setelah pembangunan struktur aplikasi selesai, tahap selanjutnya adalah integrasi konfigurasi PWA, di antaranya:

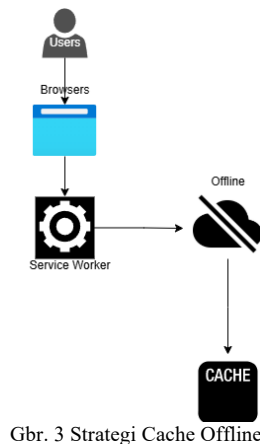
- **Manifest.json**: mendefinisikan metadata aplikasi agar bisa diinstal pada PC.
- **Service Worker**: menangani cache dan permintaan jaringan untuk mendukung fitur offline.
- **Strategi Caching**: menerapkan dua strategi yaitu, cache first dan state while revalidate.

Strategi cache sendiri menerapkan dua strategi yang pertama yaitu cache first untuk kecepatan akses ketika sudah di-cache aplikasi akan mengakses data cache terlebih dahulu, lalu yang kedua yaitu state while revalidate dimana untuk menkalibrasi data ketika kembali online dari keadaan offline. Untuk cara kerja aplikasi ketika online dan juga alur datanya akan dijelaskan oleh Gbr.2 Strategi Cache Online.



Gbr. 2 Strategi Cache Online

Strategi cache dalam kondisi offline menggunakan sepenuhnya kemampuan service worker dalam menghubungkan cache yang dijelaskan pada Gbr. 3 Strategi Cache Offline.



Gbr. 3 Strategi Cache Offline

D. Implementasi

Implementasi dilakukan secara bertahap:

- Backend: Pengelolaan catatan (CRUD) ditangani menggunakan Firebase Firestore. Autentikasi pengguna memanfaatkan Google Login melalui Firebase Authentication.
- Frontend: UI dikembangkan berdasarkan wireframe yang telah dirancang, kemudian dikonversi ke dalam komponen React.
- Versi Non-PWA dibuat dengan mengkloning proyek utama lalu menghapus konfigurasi terkait PWA (manifest, service worker, cache setup) agar dapat digunakan sebagai pembandingan dalam pengujian.

E. Pengujian

Pengujian dalam penelitian ini bertujuan untuk mengevaluasi aspek performa, fungsionalitas, serta efektivitas strategi cache yang diterapkan dalam aplikasi notebook online berbasis Progressive Web App (PWA). Alat utama yang digunakan dalam pengujian adalah Google Lighthouse, untuk mengukur performa dan kepatuhan terhadap standar PWA, serta GTMetrix, untuk mengamati kecepatan muat dan responsivitas halaman.

Pengujian dilakukan melalui berbagai skenario yang mencerminkan penggunaan aplikasi secara nyata. Untuk aspek fungsionalitas, pengujian dilakukan pada versi PWA baik dalam kondisi online maupun offline, di perangkat PC dan mobile. Versi non-PWA juga diuji dalam kondisi online sebagai pembandingan. Fokus dari pengujian ini adalah untuk memastikan bahwa aplikasi dapat dijalankan secara stabil, dapat diinstal ke perangkat pengguna (installable), dan mampu tetap berfungsi meskipun tidak terhubung ke internet, khususnya pada versi PWA.

Selain itu, dilakukan pengujian terhadap strategi cache yang diterapkan dalam versi PWA. Skenario pengujian mencakup penggunaan aplikasi secara online dalam kondisi cache aktif, akses ulang secara offline setelah pemuatan

pertama, serta pengujian saat aplikasi dijalankan tanpa cache. Versi non-PWA juga diuji untuk menunjukkan perbedaan performa akibat tidak adanya sistem cache.

Dari seluruh skenario tersebut, data performa dikumpulkan untuk kemudian dianalisis secara kuantitatif berdasarkan metrik seperti First Contentful Paint (FCP), Speed Index (SI), Time to Interactive (TTI), dan lainnya. Evaluasi ini penting untuk menilai sejauh mana teknologi PWA dan strategi cache memberikan kontribusi terhadap peningkatan kecepatan akses, efisiensi sumber daya, dan pengalaman pengguna secara keseluruhan.

F. Analisis Hasil

Data yang diperoleh dari masing-masing skenario diuji dan dibandingkan. Fokus utama adalah mengevaluasi:

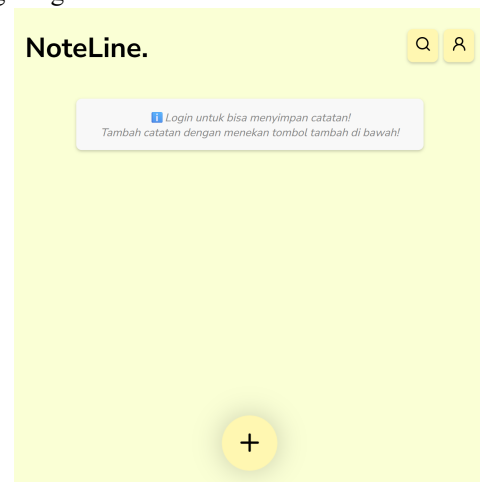
- Performa versi PWA vs non-PWA (berdasarkan FCP, LCP, SI, TTI, TBT, CLS, dll.).
- Keberhasilan akses offline dan instalasi.
- Efektivitas cache terhadap waktu muat ulang dan stabilitas aplikasi. Analisis dilakukan secara kuantitatif berdasarkan skor Lighthouse dan GTMetrix, serta observasi kualitatif terhadap pengalaman pengguna.

III. HASIL DAN PEMBAHASAN

Bab ini menampilkan hasil implementasi dari kedua aplikasi yang berhasil dirancang sedemikian rupa, dan menampilkan hasil dari berbagai pengujian.

A. Implementasi Aplikasi

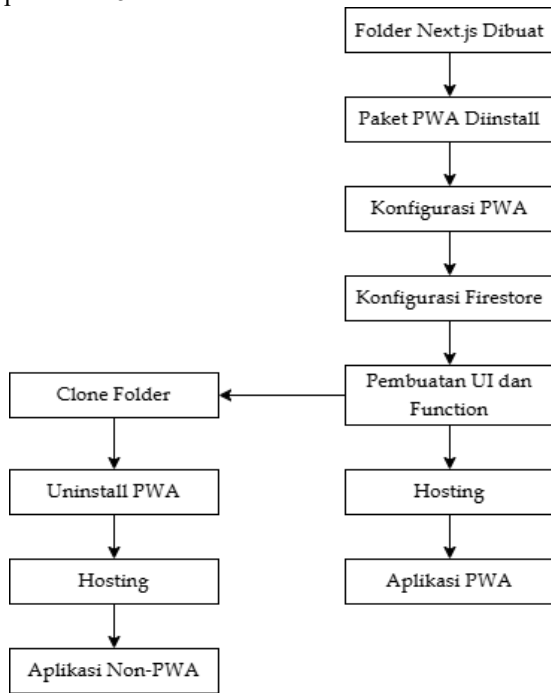
Aplikasi NoteLine dikembangkan dalam dua versi, yaitu versi Progressive Web App (PWA) dan versi non-PWA. Implementasi versi PWA dimulai dari pembuatan antarmuka dengan framework Next.js, integrasi Firebase untuk autentikasi dan database, serta penerapan fitur khas PWA seperti file manifest.json, service worker, dan strategi cache. Hasil pengembangan aplikasi versi PWA menghasilkan antarmuka yang responsif dan ringan, serta memiliki kemampuan untuk berjalan secara offline, mendukung pemasangan ke perangkat pengguna, dan dapat melakukan sinkronisasi data saat kembali terhubung dengan internet.



Gbr. 4 Tampilan Halaman Utama

Tampilan aplikasi versi PWA dapat dilihat pada Gbr. 4. Aplikasi ini dirancang untuk digunakan di desktop maupun perangkat mobile dengan tampilan yang konsisten.

Sementara itu, versi non-PWA dikembangkan dengan menduplikasi kode dari versi PWA dan menghapus semua konfigurasi yang berkaitan dengan PWA. Penghapusan mencakup file manifest, konfigurasi service worker, dan fitur cache. Versi ini tidak memiliki kemampuan offline, tidak dapat dipasang di perangkat, dan tidak memanfaatkan cache lokal. Diagram alur pengembangan versi non-PWA dapat dilihat pada Gbr. 5.



Gbr. 5 Alur Pembuatan Aplikasi

B. Uji Fungsionalitas

Pengujian dilakukan dalam berbagai skenario baik pada versi PWA maupun non-PWA. Fokus utama adalah memastikan fungsionalitas bekerja pada kondisi online maupun offline. Hasil pengujian fungsionalitas ditampilkan pada Tabel 1.

TABEL I
HASIL UJI FUNGSIONALITAS

No.	Fitur yang diuji	PWA	Non-PWA
1.	Membuka aplikasi saat offline	✓	✗
2.	Menyimpan catatan saat offline lalu sinkronisasi	✓	✗
3.	Menambahkan ke Home Screen	✓	✗
4.	Responsivitas di berbagai ukuran layar	✓	✓

5.	Notifikasi login saat menyimpan	✓	✓
----	---------------------------------	---	---

Tabel 1 menyajikan hasil dari pengujian fungsionalitas pada platform PC atau windows yang menampilkan apa yang PWA dapat berikan untuk aplikasi yang terintegrasi.

C. Pengujian Lighthouse

Pengujian Lighthouse dilakukan untuk mengukur performa aplikasi menggunakan empat metrik utama: First Contentful Paint (FCP), Total Blocking Time (TBT), Largest Contentful Paint (LCP), dan Speed Index (SI). Percobaan dilakukan sebanyak 5 kali dalam setiap skenario. Data pengujian disajikan dalam bentuk tabel dan grafik per metrik.

1. First Contentful Paint (FCP)

Pada pengujian FCP, versi PWA menunjukkan waktu muat yang lebih cepat dengan rata-rata sekitar 0,39 detik, sedangkan versi non-PWA mencatat rata-rata di atas 0,98 detik. Hal ini menunjukkan bahwa elemen pertama pada layar ditampilkan lebih cepat di versi PWA, memberikan pengalaman yang lebih responsif kepada pengguna. Data ditampilkan pada Tabel 2.

TABEL II
HASIL LIGHTHOUSE FCP

Jenis	Skenario	FCP (s)
PWA	cache offline	0.34
PWA	cache online	0.3
PWA	no cache offline	0.4
PWA	no cache online pwa	0.36
non PWA	cache online	1.0
non PWA	no cache online	1.0

Tabel 2 adalah hasil dari rata-rata dari 5 kali percobaan terhadap 6 skenario yang ditampilkan tabel.

2. Total Blocking Time (TBT)

Untuk metrik TBT, nilai pada versi PWA berkisar antara 20 hingga 28 milidetik, sedikit lebih tinggi dibandingkan non-PWA yang berada pada rentang 12 hingga 20 milidetik. Perbedaan ini disebabkan oleh aktivitas tambahan dari service worker dan caching, namun nilainya masih dalam batas optimal dan tidak mengganggu interaktivitas pengguna. Data ditampilkan pada Tabel 3.

TABEL III
HASIL LIGHTHOUSE TBT

Jenis	Skenario	TBT (ms)
PWA	cache offline	26.0
PWA	cache online	20.0
PWA	no cache offline	28.0
PWA	no cache online pwa	26.0
non PWA	cache online	20.0
Non PWA	No cache online	12.0

Tabel 3 adalah hasil dari rata-rata dari 5 kali percobaan terhadap 6 skenario yang ditampilkan tabel.

3. Largest Contentful Paint (LCP)

LCP sebagai indikator waktu render elemen terbesar juga menunjukkan keunggulan versi PWA dengan waktu sekitar 0,48 detik, sementara non-PWA memerlukan lebih dari 1,2 detik. Ini memperkuat

fakta bahwa PWA lebih cepat dalam menampilkan konten utama halaman. Data ditampilkan pada Tabel 4.

TABEL IV
HASIL LIGHTHOUSE LCP

Jenis	Skenario	LCP (s)
PWA	cache offline	0.48
PWA	cache online	0.44
PWA	no cache offline	0.64
PWA	no cache online pwa	0.52
non PWA	cache online	1.12
non PWA	no cache online	1.2

Tabel 4 adalah hasil dari rata-rata dari 5 kali percobaan terhadap 6 skenario yang ditampilkan tabel.

4. Speed Index (SI)

Speed Index pada versi PWA stabil di kisaran 1 detik, sedangkan non-PWA mencatat hingga 2,8 detik. Semakin kecil nilai Speed Index, semakin cepat dan halus tampilan visual halaman yang dirender. Data ditampilkan pada Tabel 5.

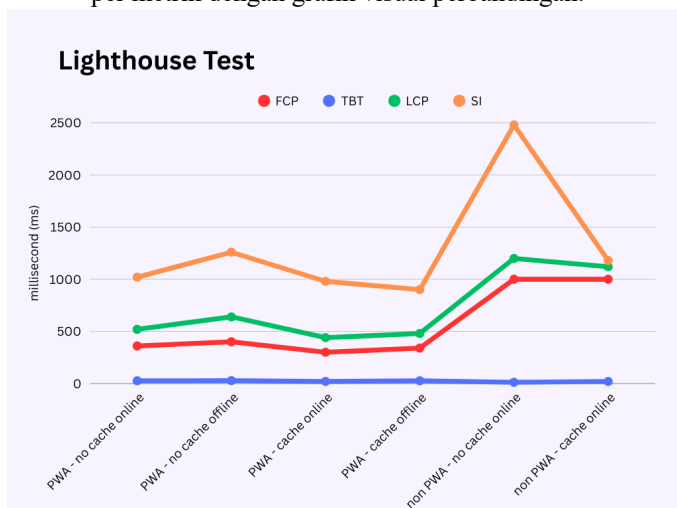
TABEL V
HASIL LIGHTHOUSE SI

Jenis	Skenario	SI (s)
PWA	cache offline	0.9
PWA	cache online	0.98
PWA	no cache offline	1.26
PWA	no cache online pwa	1.02
non PWA	cache online	1.18
non PWA	No cache online	2.48

Tabel 5 adalah hasil dari rata-rata dari 5 kali percobaan terhadap 6 skenario yang ditampilkan tabel.

5. Rata-rata Keseluruhan

Keseluruhan skor Lighthouse menunjukkan performa yang lebih baik secara menyeluruh pada versi PWA. Gbr. 6 menampilkan hasil rekap pengujian Lighthouse per metrik dengan grafik visual perbandingan.



Gbr. 6 Grafik Pengujian Lighthouse

D. Pengujian GTmetrix

GTmetrix digunakan untuk mengukur performa dari sisi kecepatan back-end dan pemuatan halaman. Pengujian menunjukkan bahwa versi PWA secara signifikan lebih unggul. Waktu First Contentful Paint, Time to Interactive, dan Largest Contentful Paint pada versi PWA berada di bawah 0,2 detik. Sebaliknya, versi non-PWA mencatat waktu lebih dari 1 detik pada semua metrik tersebut.

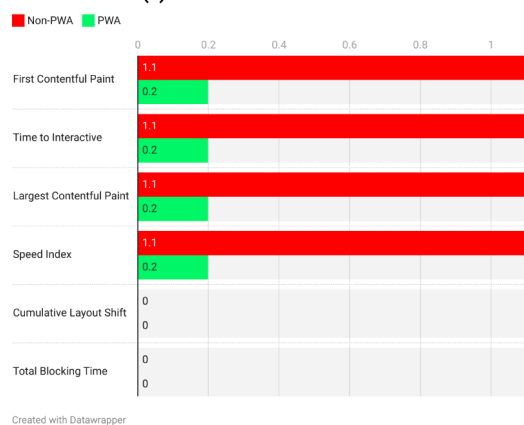
Pengukuran lainnya seperti Time to First Byte (TTFB) dan Backend Duration juga menunjukkan efisiensi yang lebih baik pada versi PWA. TTFB versi PWA hanya 96 milidetik, sementara non-PWA membutuhkan 264 milidetik. Backend Duration versi PWA sebesar 34 milidetik jauh lebih rendah dibandingkan 151 milidetik pada versi non-PWA. Tabel 6 merangkum hasil pengujian GTmetrix, dan Gbr. 7 menampilkan grafik visual perbandingan metrik antara kedua versi aplikasi.

TABEL VI
HASIL UJI GTMETRIX

Metrik	Non-PWA	PWA
Cumulative Layout Shift	0	0
First Contentful Paint	1.1 s	0.2 s
Time to Interactive	1.1 s	0.2 s
Largest Contentful Paint	1.1 s	0.2 s
Speed Index	1.1 s	0.2 s
Total Blocking Time	0 ms	0 ms
Redirect Duration	N/A	N/A
Connect Duration	113 ms	62 ms
Backend Duration	151 ms	34 ms
Time to First Byte	264 ms	96 ms
First Paint	1.1 s	223 ms
DOM Loaded	614 ms	128 ms
DOM Interactive	613 ms	147 ms
Onload Time	757 ms	314 ms
Fully Loaded Time	1.1 s	440 ms

Grafik perbandingan ditampilkan pada Gbr. 7 untuk mempermudah visualisasi data hasil uji.

GTmetrix Test (s)



Gbr. 7 Hasil Uji GTmetrix

IV. KESIMPULAN

Penelitian ini telah menghasilkan sebuah aplikasi notebook online berbasis Progressive Web App (PWA) yang dapat

berjalan pada berbagai perangkat, baik desktop maupun mobile. Aplikasi tersebut mampu diakses dalam kondisi koneksi terbatas maupun tanpa koneksi internet sekalipun, berkat penerapan teknologi service worker dan strategi cache yang efektif. Implementasi dilakukan menggunakan framework Next.js dan layanan Firebase, yang kemudian dibandingkan dengan versi non-PWA yang memiliki konfigurasi minimal.

Berdasarkan hasil pengujian fungsionalitas, aplikasi versi PWA menunjukkan stabilitas yang baik dalam berbagai kondisi: online, offline, dan pada perangkat yang berbeda. Aplikasi dapat diinstal seperti aplikasi native dan tetap berjalan meskipun koneksi internet terputus.

Pengujian performa menggunakan Google Lighthouse menunjukkan bahwa aplikasi versi PWA memiliki metrik performa yang lebih baik dibandingkan dengan versi non-PWA. Peningkatan performa secara signifikan terlihat pada metrik FCP, LCP, TBT, dan SI di versi PWA. Sebagai contoh, waktu rata-rata First Contentful Paint tercatat sebesar 0,39 detik pada versi PWA, yang menunjukkan respons lebih cepat dibandingkan 0,98 detik pada versi non-PWA. Selain itu, skor performa keseluruhan juga lebih tinggi.

GTMetrix turut memperkuat hasil tersebut dengan data numerik yang menunjukkan bahwa versi PWA unggul dalam waktu rendering, backend duration, serta efisiensi struktur halaman.

Secara umum, hasil dari semua skenario pengujian menunjukkan bahwa implementasi teknologi PWA memberikan dampak positif terhadap performa, efisiensi, dan pengalaman pengguna pada aplikasi berbasis web. Oleh

karena itu, penggunaan teknologi PWA sangat tepat untuk aplikasi berbasis teks seperti notebook online.

Adapun saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut sebagai berikut:

1. Pengujian langsung dengan pengguna akhir untuk mengevaluasi kenyamanan dan desain UI.
2. Audit keamanan dan privasi, terutama karena aplikasi menyimpan data pengguna secara cloud.
3. Pertimbangan pengembangan versi native dengan Flutter atau React Native untuk skala lebih besar.

REFERENSI

- [1] A. Rachmawati, "Membangun Informasi Layanan Umum Rumah Kos Melalui Aplikasi Berbasis Web," *J. Ilm. FIFO*, vol. 9, no. 2, p. 155, 2017, doi: 10.22441/fifo.2017.v9i2.009.
- [2] E. Nurhayati and Y. S. Rahmi, "Sejarah Web Service Dalam Perkembangan Teknologi Informasi," *Sej. Web Serv. Dalam Perkemb. Teknol. Inf.*, p. 7, 2020.
- [3] A. E. Prasetyadi, "Web 3.0: Teknologi Web Masa Depan," vol. 1, no. 3, pp. 1–6, 2011.
- [4] A. A. Ramadhani and U. P. Raya, "Perancangan Dan Pembuatan Aplikasi Notes Berbasis Android," no. July, pp. 0–7, 2021.
- [5] J. Lee, H. Kim, J. Park, I. Shin, and S. Son, "Pride and prejudice in progressive web apps: Abusing native app-like features in Web applications," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1731–1746, 2018, doi: 10.1145/3243734.3243867.
- [6] R. A. Saputra, I. A. Kautsar, N. Ariyanti, and C. Aurusta, "Implementasi Progressive Web Apps Pada Platform Freelance Digital Dengan Metode Rapid Application Development," *Smatika J.*, vol. 14, no. 01, pp. 157–168, 2024, doi: 10.32664/smatika.v14i01.1228.
- [7] R. Kusnawan, "Implementasi Progressive Web App pada Aplikasi Manajemen Data Dropship," 2021.