

IMPLEMENTASI *PROGRESSIVE WEB APPLICATION* PADA WEBSITE JAHITKU MENGGUNAKAN *FRAMEWORK LARAVEL*

Indah Kurnia Illahi¹, Ahyar Muawwal^{2*}, Yeni Saharaeni³
^{1,2,3}Sistem Informasi, STMIK KHARISMA Makassar
e-mail: ¹indahkurnia_21@kharisma.ac.id, ²ahyar@kharisma.ac.id,
³yenisaharaeni@kharisma.ac.id

Abstrak

Penelitian ini mengeksplorasi penerapan teknologi *Progressive Web Application (PWA)* pada website Jahitku dengan memanfaatkan *framework Laravel*. Fokus utama dari penelitian ini adalah untuk menciptakan pengalaman pengguna yang lebih responsif dan cepat, mirip dengan aplikasi *mobile*, serta memastikan aksesibilitas dalam kondisi *offline*. Melalui survei yang melibatkan 17 pengguna, ditemukan bahwa kebutuhan akan akses yang stabil dan cepat menjadi prioritas utama. Metodologi yang digunakan mencakup pengumpulan data melalui studi literatur dan pengujian performa menggunakan alat seperti *Lighthouse* dan *GTMetrix*. Hasil penelitian menunjukkan bahwa implementasi PWA pada website Jahitku tidak hanya memenuhi semua kriteria PWA, tetapi juga menghasilkan peningkatan signifikan dalam performa, dengan skor *Lighthouse* meningkat dari 44% menjadi 99% dan *GTMetrix* dari 61% menjadi 97%. Penelitian ini menegaskan pentingnya adopsi teknologi PWA dalam meningkatkan kinerja website dan memberikan nilai tambah bagi pengguna, serta membuka peluang untuk penelitian lebih lanjut dalam pengembangan aplikasi berbasis web.

Kata kunci: *Progressive Web App, Laravel, Framework, Jahitku,*

Abstract

This research explores the implementation of Progressive Web Application (PWA) technology on the Jahitku website, leveraging the Laravel framework. The main focus of this study is to create a more responsive and faster user experience, similar to a mobile application, and ensure offline accessibility. Through a survey involving 17 users, it was found that the need for stable and fast access is the top priority. The methodology used includes data collection through literature studies and performance testing using tools like Lighthouse and GTMetrix. The research findings show that the implementation of PWA on the Jahitku website not only meets all PWA criteria but also results in a significant improvement in performance, with Lighthouse scores increasing from 44% to 99% and GTMetrix from 61% to 97%. This study emphasizes the importance of adopting PWA technology in enhancing website performance and providing added value to users, as well as opening opportunities for further research in web-based application development.

Keywords: *Progressive Web App, Laravel, Framework, Jahitku,*

1. PENDAHULUAN

Jahitku merupakan sebuah platform *online* yang berupa website yang menyediakan jasa jahit *online* yang saat ini dapat diakses melalui Alamat <https://jahitku.my.id/>. Platform Jahitku mengklasifikasikan fitur-fiturnya ke dalam kategori "*must have*" dan "*should have*" untuk memenuhi kebutuhan dasar pengguna. Fitur "*must have*" yang telah dibangun mencakup fitur

* Corresponding author : Ahyar Muawwal (ahyar@kharisma.ac.id)

pencarian produk, halaman produk, detail produk, dan *login/signup*, sementara fitur "*should have*" meliputi fitur seperti *Progressive Web Application (PWA)*, *Progressive Web Application (PWA)* itu sendiri merupakan sebuah konsep dalam pengembangan website yang mengadopsi teknologi *service worker*, *web manifest*, dan *cache API* [1]. *PWA (Progressive Web Application)* memanfaatkan kemampuan web modern yang menggabungkan berbagai teknologi, konsep desain, dan *API Web* untuk memberikan pengalaman pengguna yang mirip dengan aplikasi native [2]. Maka dari itu akan mengimplementasikan fitur *PWA* pada website Jahitku untuk melanjutkan penambahan fitur "*should have*" setelah fitur "*must have*" telah dibangun.

Survei terhadap 17 pengguna website Jahitku menemukan bahwa sebagian besar pengguna (70,5%) ingin dapat mengakses website dengan mudah, bahkan saat koneksi internet buruk atau dalam keadaan *offline*. Terlihat juga besarnya peningkatan pemakaian aplikasi *mobile* telah meningkatkan seiring perkembangan teknologi. Terlebih lagi sejumlah *competitor*, yang juga menyediakan layanan jasa jahit *online* telah mendukung *platform mobile* untuk mengakses layanan-layanannya. Dilansir dari *investor.id* pengguna *smartphone* di Indonesia mencapai 194,26 juta pada tahun 2024 ini bertambah 4,23 juta (2,23%) dibandingkan tahun 2023 masih 190,03 juta [3], serta bisa terlihat pada data dari Badan Pusat Statistik (BPS), penetrasi *smartphone* di Indonesia pada tahun 2023 mencapai 92,5%. Artinya, lebih dari 9 dari 10 orang Indonesia telah memiliki *smartphone* [4]. Maka dari itu penelitian ini bertujuan untuk mengimplementasikan teknologi *Progressive Web App (PWA)* pada website Jahitku dengan menggunakan *framework Laravel*. Tujuannya adalah agar website Jahitku dapat diakses seperti aplikasi native melalui layar beranda *smartphone* dan memiliki halaman statis khusus untuk *offline mode*.

Pengimplementasi *PWA* pada website Jahitku menggunakan *framework Laravel* dengan mempertimbangkan beberapa hal seperti kemudahan dalam mengelola template statis website Jahitku karena *Laravel* terintegrasi dengan berbagai *template engine* populer seperti *Blade*, *Twig*, atau *Handlebars*, proses *rendering* website di sisi *server (server-side rendering)* yang disediakan oleh *Laravel* [5]. Dengan begitu halaman website dapat ditampilkan lebih cepat, fitur-fitur bawaan *Laravel* yang memastikan website menjadi *SEO-friendly* dan memiliki performa yang baik, seperti *URL routing*, *sitemaps*, dan *caching*, kemudahan dalam *setup* dan *deployment* proyek *Laravel* menggunakan *tools* seperti *Composer*, *Artisan*, dan *Laravel Forger* [6]. Kemampuan *Laravel* dalam pengembangan *API* dan dukungan terhadap berbagai *third-party middleware* dan sifat *Laravel* yang *unopinionated*, memberikan fleksibilitas dalam hal kustomisasi struktur folder aplikasi dan fungsi *middleware* [7]. Dengan pertimbangan-pertimbangan tersebut, diharapkan implementasi *PWA* pada website Jahitku menggunakan *framework Laravel* dapat dilakukan dengan mudah dan dapat meningkatkan kinerja website Jahitku secara keseluruhan.

Penelitian yang telah dilakukan sebelumnya terkait *pwa* memiliki beberapa perbedaan dan keunikan jika dibandingkan dengan penelitian-penelitian terdahulu. Pertama, berencana menggunakan kerangka kerja *Laravel* dalam mengimplementasikan teknologi *Progressive Web Application (PWA)* pada website yang akan diteliti, berbeda dengan penelitian-penelitian

sebelumnya yang cenderung menggunakan kerangka kerja lain seperti *Quasar*[8], *Next.js*[9], atau *CodeIgniter*[10]. Selanjutnya, selain fokus pada implementasi PWA, juga berencana untuk melakukan pengujian yang lebih komprehensif dibandingkan dengan penelitian-penelitian sebelumnya. Rencana adalah untuk menguji apakah website memenuhi kriteria PWA, menganalisis performa website, serta mengukur ukuran sumber daya yang ditransfer sebelum dan setelah implementasi PWA. Terakhir, sementara penelitian-penelitian terdahulu berfokus pada website atau aplikasi lain, akan memusatkan perhatian pada implementasi PWA pada website Jahitku. Dengan demikian, penelitian yang akan dilaksanakan memiliki keunikan dan kontribusi baru dibandingkan dengan penelitian-penelitian sebelumnya, terutama dalam hal penggunaan kerangka kerja *Laravel* dan rencana pengujian yang lebih komprehensif. Hal ini dapat menjadi nilai tambah bagi penelitian di masa mendatang.

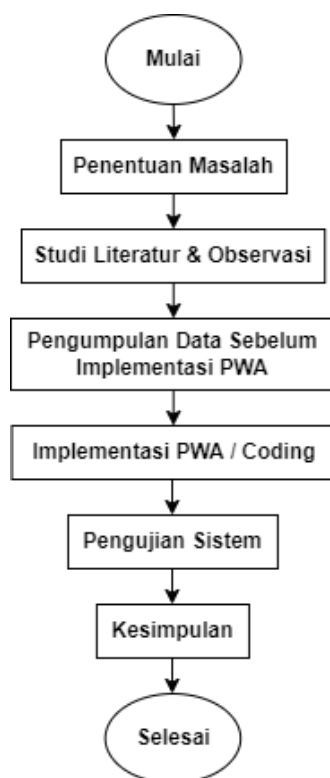
2. METODE PENELITIAN

2.1 Tahapan Penelitian

Proses pengujian yang dilakukan untuk memastikan aplikasi tersebut memenuhi kriteria *Progressive Web Application* (PWA) meliputi beberapa tahap. Pertama, pengujian *installable* yang mencakup pengujian kompatibilitas aplikasi dengan berbagai tipe perangkat *mobile* dan *Browser*. Hal ini dilakukan untuk memastikan aplikasi dapat diinstal dan diakses dengan baik pada berbagai perangkat. Selain itu, pengujian juga dilakukan menggunakan *Lighthouse* untuk memverifikasi terpenuhinya dua parameter utama dalam kriteria PWA, yaitu *Installable* dan *PWA Optimized*.

Tahap selanjutnya adalah pengujian manual serta penggunaan alat *PWABuilder*, dengan fokus pada dua parameter lainnya, yakni *Service worker* dan *Manifest.Json*. Selain pengujian *installable*, dilakukan pula pengujian performa aplikasi. Pengujian performa menggunakan *Lighthouse* berfokus pada enam variabel, sementara penggunaan *GTMetric* berfokus pada tujuh variabel pengujian. Tujuan dari pengujian performa ini adalah untuk memastikan aplikasi memiliki kinerja yang optimal dan dapat memberikan pengalaman pengguna yang baik.

Keseluruhan proses pengujian ini dirancang untuk memastikan aplikasi dapat diinstal dan diakses dengan mudah pada berbagai perangkat, serta memenuhi kriteria PWA dan memiliki performa yang optimal. Untuk pengujian *offline mode website* Jahitku, hal tersebut dilakukan menggunakan *Browser DevTools*. Proses penelitian yang dilakukan akan dimulai dari identifikasi masalah, melakukan implementasi coding PWA, melakukan pengujian sistem, dan terakhir akan membuat kesimpulan.



Gambar 1 Tahapan Penelitian

2.2 Jenis Data dan Sumber Data

Penelitian dilakukan dengan menggunakan data kuantitatif. Data kuantitatif diperoleh dari pengujian performa website Jahitku menggunakan alat seperti *Lighthouse* dan *GTMetrix*, serta pengujian fitur *Progressive Web Application* (PWA) menggunakan *PWABuilder*. Hasil pengujian berupa skor angka untuk aspek performa, PWA, struktur, dan metrik lainnya. Selain itu, juga dilakukan pengujian *offline* yang akan menghasilkan data kuantitatif terkait keberhasilan implementasi *service worker*.

3. Hasil dan Pembahasan

3.1. Implementasi PWA

Proses pengimplementasian PWA pada website Jahitku menggunakan *Laravel* melibatkan beberapa tahapan, yang meliputi :

1. Instalasi PWA pada *Laravel*.

```
composer require --dev ladumor/laravel-pwa
```

Gambar 2 Instalasi *ladumor Laravel pwa*

2. Mengatur *Manifest.json*

Manifest.json adalah file konfigurasi yang digunakan dalam PWA untuk memberikan informasi kepada *Browser* tentang bagaimana aplikasi harus ditampilkan dan beroperasi ketika diinstal sebagai aplikasi mandiri. Dalam konteks *Laravel* PWA, file *Manifest.json* digunakan untuk mengkonfigurasi pengaturan PWA aplikasi [5].

```
public > {} manifest.json > ...
1      {
2          "name": "Jahitku",
3          "short_name": "JHT",
4          "start_url": "/index.php",
5          "background_color": "#C1813B",
6          "description": "Jahitku",
7          "display": "fullscreen",
8          "theme_color": "#C1813B",
9          "icons": [
10         {
11             "src": "logo_jahit.PNG",
12             "sizes": "512x512",
13             "type": "image/png",
14             "purpose": "any maskable"
15         }
16     ]
17 }
18
```

Gambar 3 Menambah *Manifest.json*

Pada Gambar 3 terlihat file ini berisi berbagai pengaturan yang dapat disesuaikan sesuai kebutuhan *Manifest.json* memberikan informasi penting kepada *Browser* tentang tampilan dan perilaku PWA saat diinstal sebagai aplikasi mandiri. *browser* menggunakan informasi ini untuk menyesuaikan tata letak, ikon, dan interaksi dengan pengguna.

```
20     <!-- PWA -->
21     <meta name="theme-color" content="#6777ef"/>
22     <link rel="apple-touch-icon" href="{{ asset('image/navbar/logo.png') }}">
23     <link rel="manifest" href="{{ asset('/manifest.json') }}">
24     </head>
25
```

Gambar 4 Memanggil *Manifest.json*

Selanjutnya memanggil *manifest.json* pada *root blade file*. Tindakan ini dilakukan untuk memuat *file manifest.json* dan memberitahu *browser* bahwa situs web *Jahitku* telah mengadopsi teknologi PWA dan dapat diinstal oleh pengguna.

3. Menambahkan *Service worker*

Service worker adalah skrip JavaScript yang berjalan di latar belakang *browser*, berperan sebagai pintu masuk untuk fitur-fitur PWA seperti penanganan permintaan jaringan, penyimpanan *cache*, dan *offline mode*. Karena terpisah dari halaman *website*, *service worker* memerlukan proses registrasi terlebih dahulu [6].

```
<script src="{{ asset('/sw.js') }}"></script>
<script>
if ("serviceWorker" in navigator) {
  // Register a service worker hosted at the root of the
  // site using the default scope.
  navigator.serviceWorker.register("/sw.js").then(
    (registration) => {
      console.log("Service worker registration succeeded:", registration);
    },
    (error) => {
      console.error("Service worker registration failed: ${error}`);
    },
  );
} else {
  console.error("Service workers are not supported.");
}
</script>
```

Gambar 5 Melakukan Registrasi

```
const preLoad = function () {
  return caches.open("offline").then(function (cache) {
    // caching index and important routes
    return cache.addAll(filesToCache);
  });
};
```

Gambar 6 Mengatur Service worker

Gambar 6 menjalankan fungsi *preLoad* dalam *service worker*, yang membuka *cache* bernama "offline" menggunakan *caches.open()* dan menyimpan file-file penting untuk *offline*. Fungsi ini dapat dipanggil saat instalasi atau aktivasi *service worker*, sehingga file-file tersebut dapat diambil dari *cache* saat aplikasi *offline*.

```
const filesToCache = [
  '/',
  '/offline.html',
  '/no-wifi.png',
  '/awan_galau.png',
  '/Cute_Dino.ttf',
  '/style.css'
];
```

Gambar 7 menyimpan file ke cache

Pada Gambar 7 *service worker* akan memuat dan menyimpan file-file tersebut ke dalam *cache* sehingga dapat digunakan saat website diakses secara *offline* atau ketika koneksi internet tidak tersedia.

```
self.addEventListener("fetch", function (event) {
  event.respondWith(checkResponse(event.request).catch(function () {
    return returnFromCache(event.request);
  }));
  if(!event.request.url.startsWith('http')){
    event.waitUntil(addToCache(event.request));
  }
});
```

Gambar 8 Menangani *event fetch* pada *Service worker*

Pada Gambar 8 tersebut menunjukkan bahwa *service worker* akan menangani event "*fetch*". Ketika ada permintaan untuk mengambil suatu resource, *service worker* akan memeriksa respons dari permintaan tersebut. Jika respons berhasil diterima, respons tersebut akan dikembalikan sebagai respons utama. Namun, jika terjadi kesalahan atau respons memiliki status 404, *service worker* akan mengambil respons yang cocok dari *cache*. Jika permintaan adalah permintaan lokal atau internal, *service worker* akan juga menyimpan permintaan tersebut ke dalam *cache*

3.2. Pengujian Sistem

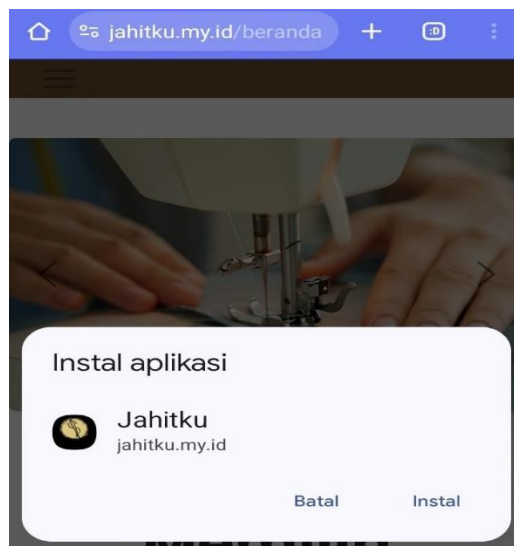
Terdapat beberapa macam pengujian sistem terhadap website Jahitku untuk memastikan keberhasilan dalam mengimplementasikan PWA pada website Jahitku, yaitu:

1. Pengujian *Installable*

Pengujian "*Installable*" pada situs web Jahitku dilakukan untuk memastikan apakah situs web tersebut dapat diinstal seperti aplikasi native atau desktop pada perangkat pengguna. Pengujian ini akan dilakukan untuk perangkat desktop maupun perangkat *mobile*.

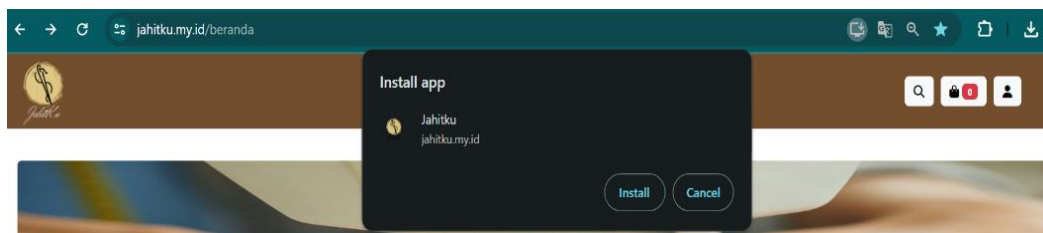


Gambar 9 Icon Aplikasi Jahitku



Gambar 10 Add to Home Screen & pop up install aplikasi Jahitku

Pengujian Pemasangan Desktop (Desktop *Installable*) dengan pengujian pada perangkat dengan *platform* yang berbeda yaitu *Chrome* dan *Edge*. Adapun sepuluh perangkat *mobile* yang akan dilakukan pengujian ialah iPhone XR, iPhone 15 Pro Max, iPhone SE, iPhone 11, iPhone 13 Samsung Galaxy A21s, OPPO Reno3, POCO M5, Realme GT Neo 2 dan, HUAWEI nova 7.



Gambar 11 Aplikasi Desktop Jahitku Google Chrome

Untuk mengevaluasi kemampuan website Jahitku yang telah mengadopsi teknologi PWA, dilakukan pengujian instalasi pada perangkat dengan platform yang beragam, baik *mobile* maupun desktop. Hasil pengujian tersebut menunjukkan bahwa website Jahitku berhasil diinstal dengan baik pada berbagai perangkat pengguna, baik perangkat *mobile* maupun desktop. Oleh karena itu, website Jahitku diberikan nilai 1 sebagai indikasi keberhasilan dalam pengujian *mobile installable* dan desktop *installable*.

2. Pengujian Kriteria PWA

Proses pengujian kriteria PWA akan dilakukan dengan menggunakan alat-alat seperti *Lighthouse* dan *PWABuilder* untuk melakukan pengukuran langsung pada situs website jahitku

a. *Lighthouse*

Hasil pengujian terpenuhinya suatu kriteria PWA pada website Jahitku menggunakan *Lighthouse*, yaitu

Tabel 1. Hasil Pengujian Kriteria PWA Menggunakan *Lighthouse*
 [Sumber : Hasil Pengujian Kriteria PWA Jahitku Menggunakan *Lighthouse*, 2024]

No.	Kriteria	Sebelum Implementasi PWA		Setelah Implementasi PWA	
		Status	Nilai	Status	Nilai
<i>Installable</i>					
1.	<i>Web app manifest</i> atau <i>service worker</i> memenuhi persyaratan instalasi aplikasi	Tidak Terpenuhi	0	Terpenuhi	1
<i>PWA Optimized</i>					
2.	Mendaftarkan <i>service worker</i> yang mengontrol halaman dan <i>start_url</i>	Tidak Terpenuhi	0	Terpenuhi	1
3.	Konfigurasi untuk <i>splash screen</i> kustom	Tidak Terpenuhi	0	Terpenuhi	1
4.	Mengatur warna tema untuk <i>address bar</i>	Tidak Terpenuhi	0	Terpenuhi	1
5.	Ukuran konten tepat untuk <i>viewport</i>	Terpenuhi	1	Terpenuhi	1
6.	Mempunyai <i>tag</i> <code><meta name="viewport"></code> dengan atribut <i>width</i> atau <i>initial-scale</i>	Terpenuhi	1	Terpenuhi	1
7.	<i>Manifest</i> memiliki <i>maskable icon</i>	Tidak Terpenuhi	0	Terpenuhi	1
<i>Manually Check</i>					
8.	Situs dapat dijalankan di berbagai <i>Browser</i> (cross- <i>Browser</i>)	Tidak Terpenuhi	0	Terpenuhi	1
9.	Transisi halaman terasa cepat dan tidak terasa seperti diblokir di jaringan	Tidak Terpenuhi	0	Terpenuhi	1
10.	Setiap halaman memiliki <i>URL</i>	Tidak Terpenuhi	0	Terpenuhi	1
Total Nilai			2		10

Seperti yang ditunjukkan pada Tabel 3, setiap kriteria PWA yang terpenuhi akan diberikan skor 1, sedangkan yang tidak terpenuhi akan diberi skor 0. Sebelum implementasi PWA, website Jahitku hanya memenuhi 2 kriteria dengan skor 2. Namun, setelah implementasi, website ini memenuhi seluruh 10 kriteria dengan skor 10. Hal ini menunjukkan keberhasilan implementasi PWA pada website Jahitku, di mana seluruh kriteria yang direkomendasikan oleh *Lighthouse* telah terpenuhi.

b. *PWABuilder*

Berikut pengujian kesesuaian website Jahitku dengan kriteria PWA menggunakan *PWABuilder*, baik sebelum maupun setelah implementasi PWA.

Tabel 2. Hasil Pengujian Kriteria PWA Menggunakan *PWABuilder*

[Sumber : Hasil Pengujian Kriteria PWA Jahitku Menggunakan *PWABuilder*, 2024]

No.	Kriteria	Sebelum Implementasi PWA		Setelah Implementasi PWA	
		Status	Nilai	Status	Nilai
1.	<i>Manifest</i> memiliki properti <i>icon</i>	Tidak Terpenuhi	0	Terpenuhi	1
2.	<i>Manifest</i> memiliki properti <i>name</i>	Tidak Terpenuhi	0	Terpenuhi	1
3.	<i>Manifest</i> memiliki properti <i>short_name</i>	Tidak Terpenuhi	0	Terpenuhi	1
4.	<i>Manifest</i> memiliki properti <i>start_url</i>	Tidak Terpenuhi	0	Terpenuhi	1
5.	<i>Manifest</i> memiliki properti <i>background_color</i> dalam format <i>hex</i> .	Tidak Terpenuhi	0	Terpenuhi	1
6.	<i>Manifest</i> memiliki properti <i>description</i>	Tidak Terpenuhi	0	Terpenuhi	1
7.	<i>Manifest</i> memiliki properti <i>display</i>	Tidak Terpenuhi	0	Terpenuhi	1
8.	<i>Manifest</i> memiliki properti <i>display_override</i>	Tidak Terpenuhi	0	Terpenuhi	1
10.	<i>Icons</i> memiliki setidaknya satu <i>icon</i> dengan tujuan apapun	Tidak Terpenuhi	0	Terpenuhi	1
11.	<i>Manifest</i> memiliki properti <i>theme_color</i> dalam format <i>hex</i> .	Tidak Terpenuhi	0	Terpenuhi	1
12.	<i>Manifest</i> memiliki properti <i>shortcuts</i>	Tidak Terpenuhi	0	Terpenuhi	1
13.	Memiliki <i>service worker</i>	Tidak Terpenuhi	0	Terpenuhi	1
Total Nilai			0		13

Seperti yang terlihat pada Tabel 2, Sebelum implementasi PWA, website Jahitku memenuhi 2 kriteria dengan skor 2. Setelah implementasi, skor meningkat menjadi 13 dengan 13 kriteria terpenuhi. Pengujian menggunakan *PWABuilder* menunjukkan implementasi PWA

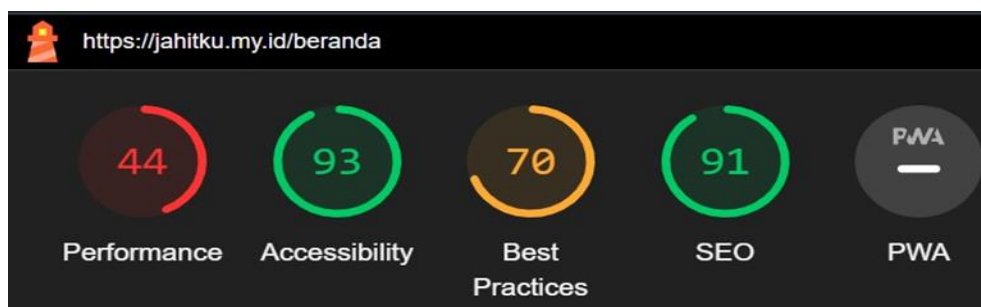
berhasil, bahkan melebihi beberapa kriteria yang direkomendasikan, seperti *Service worker* dan *Manifest*.

3. Pengujian *Performance*

Pengujian dilakukan melalui pengukuran langsung menggunakan tools seperti *Lighthouse* dan *GTMetrix*.

a. *Lighthouse*

Berikut hasil pengujian *performance* dari website Jahitku menggunakan *Lighthouse* sebelum dan sesudah pengimplementasian PWA



Gambar 12 Pengujian sebelum implementasi PWA

Seperti yang disajikan pada Tabel 3, hasil pengujian kinerja website Jahitku PWA pada website Jahitku meningkatkan kinerja website secara signifikan. Skor kinerja meningkat dari 44% menjadi 99%. *First Contentful Paint* membaik dari 1.7s menjadi 0.6s, *Total Blocking Time* berkurang dari 380ms menjadi 0ms, *Speed Index* membaik dari 9.3s menjadi 1.1s, *Largest Contentful Paint* menurun dari 27.2s menjadi 0.7s, dan *Cumulative Layout Shift* tetap stabil di 0.

Tabel 3. Hasil Pengujian Performance Jahitku Menggunakan *Lighthouse*
[Sumber : Hasil Pengujian Performance Jahitku Menggunakan *Lighthouse*, 2024]

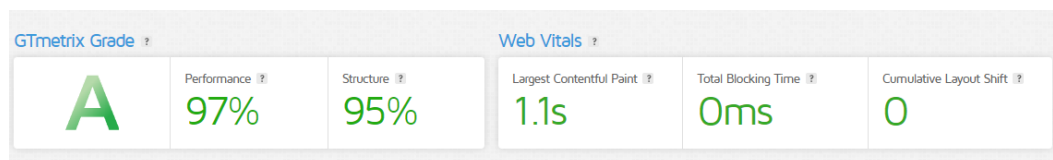
No.	Variabel	Sebelum Implementasi PWA	Setelah Implementasi PWA
1.	<i>Performance</i>	44%	99%
2.	<i>First Contentful Paint</i>	1.7s	0.6s
3.	<i>Total Blocking Time</i>	380ms	0ms
4.	<i>Speed Index</i>	9.3s	1.1s
5.	<i>Largest Contentful Paint</i>	27.2s	0.7s
6.	<i>Cumulative Layout Shift</i>	0	0

b. *GTMetrix*

Berikut hasil pengujian *performance* dari website Jahitku menggunakan *Lighthouse* sebelum dan sesudah pengimplementasian PWA



Gambar 14 Pengujian Performance Pada Jahitku Sebelum Implementasi PWA Menggunakan *GTmetrix*



Gambar 15 Pengujian Performance Pada Jahitku Sesudah Implementasi PWA Menggunakan *GTmetrix*

Berikut pengujian kinerja website Jahitku menggunakan *Lighthouse*, baik sebelum maupun setelah implementasi PWA, dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian *Performance* Jahitku Menggunakan *GTmetrix*

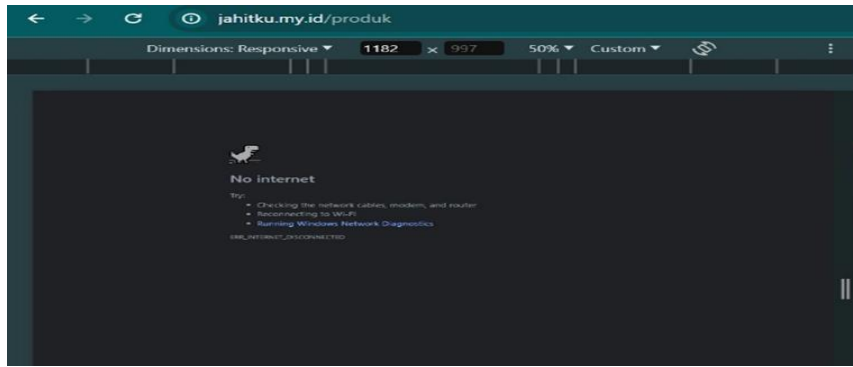
[Sumber : Hasil Pengujian *Performance* Jahitku Menggunakan *GTmetrix*, 2024]

No.	Variabel	Sebelum Implementasi PWA	Setelah Implementasi PWA
1.	<i>Performance</i>	61% (<i>Grade C</i>)	97% (<i>Grade A</i>)
2.	<i>First Contentful Paint</i>	1.3s	876ms
3.	<i>Time to Interactive</i>	1.4s	1.4ms
4.	<i>Speed Index</i>	7.8s	1.1s
5.	<i>Total Blocking Time</i>	0ms	0ms
6.	<i>Largest Contentful Paint</i>	21.0s	1.1s
7.	<i>Cumulative Layout Shift</i>	0	0

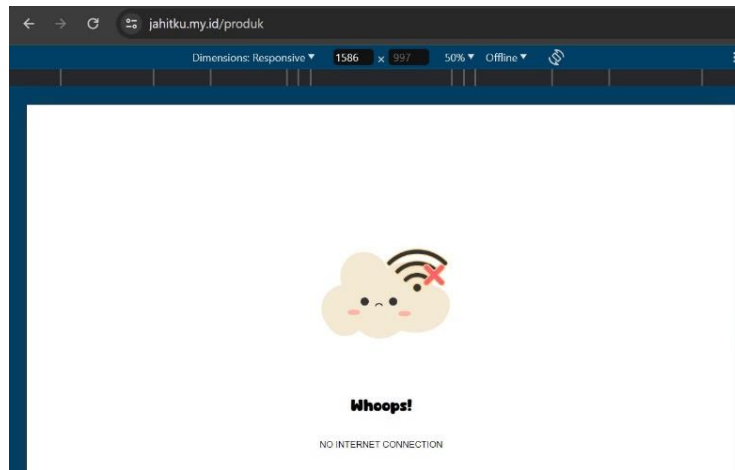
Seperti yang dapat dilihat pada Tabel 6, hasil pengujian kinerja website Jahitku menggunakan *GTmetrix* setelah implementasi PWA menunjukkan peningkatan yang signifikan dibandingkan sebelum implementasi. Skor kinerja website meningkat dari 61% menjadi 97%, durasi *First Contentful Paint* dari 1.4s menjadi 876ms, *Time to Interactive* dari 3.3s menjadi 1.4s, *Speed Index* dari 2.3s menjadi 432ms, *Total Blocking Time* tetap 0ms, *Largest Contentful Paint* dari 21.0s menjadi 1.1s, dan *Cumulative Layout Shift* tetap 0.

4. Pengujian *Offline Mode*

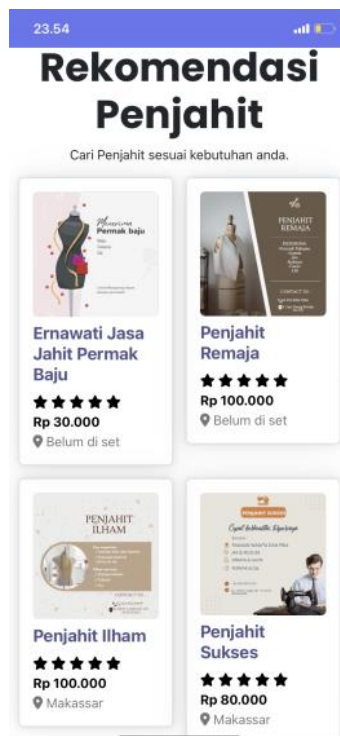
Pengujian *offline mode* dilakukan untuk menguji apakah situs web Jahitku memiliki fitur *offline mode*, yang merupakan salah satu fitur yang dimiliki oleh aplikasi web yang telah mengadopsi teknologi PWA. Pengujian ini dilakukan dengan menggunakan alat pengembang (*DevTools*) pada *Browser*, khususnya pada bagian pengaturan jaringan. Pengujian dilakukan dengan mengubah pengaturan dari "*no throttling*" menjadi "*offline*" atau dalam kondisi tidak terhubung ke internet.



Gambar 16 Pengujian Sebelum *Offline* Sebelum Implementasi PWA



Gambar 17. Pengujian *Offline Mode* Jahitku Setelah Implementasi PWA



Gambar 18. Pengujian *Offline Mode*

Berikut pengujian *offline mode* pada website Jahitku, sebelum dan sesudah implementasi PWA menggunakan *Browser DevTools*, dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian *Offline Mode* Menggunakan *Browser DevTools*

[Sumber : Hasil Pengujian *Offline Mode* Jahitku Menggunakan *Browser DevTools*, 2024]

No.	Pengujian		Sebelum Implementasi PWA		Setelah Implementasi PWA	
			Status	Nilai	Status	Nilai
1.	Akses <i>Offline</i>	Tidak dapat diakses	0		Dapat diakses	1
Total Nilai			0			1

Pengujian *offline mode* pada website Jahitku menunjukkan hasil yang berbeda sebelum dan setelah implementasi PWA. Sebelum implementasi, skor pengujiannya adalah 0, yang berarti gagal. Namun, setelah implementasi PWA, skor pengujiannya meningkat menjadi 1, yang berarti berhasil.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan oleh penulis, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Untuk mengimplementasikan PWA pada website Jahitku berbasis *Laravel*, terdapat dua komponen penting yang dibutuhkan. Pertama adalah *Web App Manifest*, yang menyediakan informasi rinci kepada *browser* mengenai aplikasi web tersebut. Kedua adalah *service worker*, yang terhubung dengan halaman utama website Jahitku dan dapat didaftarkan sebagai proses latar belakang di dalam *browser* pengguna.
2. Implementasi PWA pada website Jahitku memungkinkan pengguna untuk menginstal dan mengakses website tersebut seperti aplikasi *mobile native*, baik melalui *home screen smartphone* maupun aplikasi desktop pada perangkat mereka. Dengan fitur ini, pengguna dapat mengakses website tanpa perlu membuka *Browser* atau melakukan instalasi melalui toko aplikasi seperti *Play Store* atau *App Store*.
3. Teknologi PWA telah berhasil diimplementasikan pada website Jahitku, memenuhi semua kriteria yang diperlukan agar dapat dianggap sebagai PWA. Hasil pengujian dengan *Lighthouse* menunjukkan skor 10/10, mengindikasikan bahwa seluruh kriteria PWA yang direkomendasikan *Lighthouse* telah terpenuhi. Sementara pengujian menggunakan *PWABuilder* menghasilkan skor total 13, dengan 13 kriteria pengujian yang terpenuhi. Bahkan, skor ini melebihi beberapa kriteria yang direkomendasikan *PWABuilder*, seperti 2 kriteria (*Highly Recommended*) untuk *service worker*, dan 4 kriteria (*Required*) untuk *Manifest*.
4. Implementasi PWA pada website Jahitku telah secara signifikan meningkatkan performa website. Pengujian menggunakan *Lighthouse* menunjukkan skor performa naik dari 44%

sebelum implementasi PWA menjadi 99% setelahnya. Sementara pengujian dengan *GTMetrix* menunjukkan skor performa website meningkat dari 61% (Peringkat C) sebelum implementasi, menjadi 97% (Peringkat A) setelah implementasi PWA.

5. Dengan mengimplementasikan teknologi PWA, website Jahitku kini dapat memanfaatkan *offline mode*. Fitur ini memungkinkan website Jahitku tetap dapat diakses bahkan saat pengguna sedang *offline* atau memiliki koneksi internet yang tidak stabil. Dalam kondisi *offline*, website Jahitku akan menampilkan halaman statis khusus yang memberitahu pengguna bahwa mereka sedang *offline*.

DAFTAR PUSTAKA

- [1] S. Aripin and S. Somantri, "Implementasi *Progressive Web Application* (PWA) pada Repository E-Portofolio Mahasiswa," *J. Eksplora Inform.*, vol. 10, no. 2, pp. 148–158, 2021, doi: 10.30864/eksplora.v10i2.486.
- [2] A. Prayitno and E. Hariyanto, "Perancangan Aplikasi Pengelolaan Keuangan Menggunakan Metode *Progressive Web Application* (Studi Kasus : SDIT Zahra Asy Syifa Patumbak Deli Serdang)," *Bull. Inf. Technol.*, vol. 4, no. 1, pp. 9–10, 2023, doi: 10.47065/bit.v3i1.
- [3] Abdul Muslim, "Pengguna Smartphone RI Diprediksi 194 Juta," <https://investor.id>.
- [4] Sampoerna University, "Mengulik Perkembangan Penggunaan Smartphone di Indonesia," <https://www.sampoernauniversity.ac.id/>.
- [5] Desma Aipina and Harry Witriyono, "Pemanfaatan Framework *Laravel* Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web," *J. Media Infotama*, vol. 18, no. 1, pp. 36–42, 2022.
- [6] J. Nugraha, M. Dimas, A. Sudarna, D. Moeis, S. Tinggi, and I. Komputer, "SISTEM INFORMASI PROFIL PERUSAHAAN BERBASIS WEBSITE MENGGUNAKAN *LARAVEL* 8," vol. 2, no. 1, pp. 554–567, 2024.
- [7] F. Sinlae, E. Irwanda, Z. Maulana, and V. E. Syahputra, "Penggunaan Framework *Laravel* dalam Membangun Aplikasi Website Berbasis PHP," vol. 2, no. 2, pp. 119–132, 2024.
- [8] G. L. Dewi, S. Tjandra, and Ricardo, "Pemanfaatan *Progressive Web Application* Pada Web Akuntansi," *Teknika*, vol. 9, no. 1, pp. 38–47, 2020, doi: 10.34148/teknika.v9i1.252.
- [9] R. Jubhari Phie Joarno, M. Fajar, A. Yunus, S. Informasi, and S. Kharisma Makassar, "IMPLEMENTASI *PROGRESSIVE WEB APPLICATION* PADA WEBSITE GETHHELP MENGGUNAKAN NEXT.JS", [Online]. Available: <https://jurnal.kharisma.ac.id/kharismatech/>
- [10] A. I. Putra, A. Putra Kharisma, and N. Yudistira, "Implementasi Progressive Web Application dan Framework CodeIgniter pada Sistem Informasi Lomba Karya Tulis Ilmiah Mahasiswa," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 9, pp. 3773–3783, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>