

Implementation Of Progressive Web Apps On Digital Freelance Platform With Rapid Application Development Method

[Implementasi Progressive Web Apps Pada Platform Freelance Digital Dengan Metode Rapid Application Development]

Reyhan Adi Saputra¹⁾, Irwan Alnarus Kautsar^{*2)}

¹⁾ Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾ Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: irwan@umsida.ac.id

Abstract. Dalam beberapa tahun terakhir, jumlah freelancer di Indonesia meningkat signifikan. Banyak orang memilih menjadi freelancer karena fleksibilitas waktu, potensi penghasilan lebih besar, dan kendali atas pekerjaan. Freelancer juga dapat memberikan pekerjaan kepada mitra di platform freelance digital saat orderan menumpuk. Platform freelance digital menghubungkan mitra dan freelancer melalui website, namun mengunjungi URL website berulang kali kurang efisien. Progressive Web Apps (PWA) adalah solusi untuk masalah ini. Penelitian ini bertujuan mengimplementasikan PWA pada platform freelance digital. Metode yang digunakan adalah Rapid Application Development (RAD) dengan tahapan Requirements Planning, Design Workshop, dan Implementation. Hasil penelitian menunjukkan PWA berhasil diimplementasikan dan diterima dengan baik, dengan rata-rata kepuasan pengguna sebesar 93,5% dalam kategori "Sangat Bagus".

Keywords - Freelancer, Progressive Web Apps, Rapid Application Development, Service Worker, Implementasi PWA.

Abstrak. In recent years, the number of freelancers in Indonesia has increased significantly. Many people choose to freelance because of the flexibility of time, greater earning potential, and control over work. Freelancers can also assign work to partners on digital freelance platforms when orders pile up. Digital freelance platforms connect partners and freelancers via websites, but visiting the website URL repeatedly is less efficient. Progressive Web Apps (PWA) are the solution to this problem. This research aims to implement PWA on a digital freelance platform. The method used is Rapid Application Development (RAD) with Requirements Planning, Design Workshop and Implementation stages. The research results show that PWA was successfully implemented and well received, with an average user satisfaction of 93.5% in the "Very Good" category.

Kata Kunci - Freelancer, Progressive Web Apps, Rapid Application Development, Service Worker, Implementation PWA.

I. PENDAHULUAN

Dalam beberapa tahun belakangan ini, jumlah pekerja lepas atau *freelancer* di Indonesia telah meningkat secara signifikan. Banyak orang memilih menjadi *freelancer* karena mereka bisa bekerja dengan waktu yang lebih *fleksibel*, memiliki potensi penghasilan yang lebih besar, dan lebih memiliki kendali atas pekerjaan yang mereka lakukan [1]. *Freelancer* juga dapat memberikan pekerjaan terhadap mitra di platform *freelance* digital ketika orderan menumpuk dan tidak dapat ditangani sendiri. Kemitraan dianggap sebagai strategi yang tepat karena kesesuaiannya dengan sifat fleksibilitas dan berkonsep bekerja sama dengan pengguna lain sehingga dapat disebut dengan mitra kerja yang menawarkan daya tawar dapat membuka peluang kerja bagi para mitra [2].

Platform *freelance* digital menyediakan ruang dimana mitra dan *freelancer* dapat terhubung secara fleksibel melalui *website*, akan tetapi untuk mengunjungi *Uniform Resource Locator (URL) website* dilakukan secara berulang saat ingin mengunjungi platform tersebut. *Progressive Web Apps (PWA)* menjadikan solusi sebuah aplikasi yang berbasis *website*.

PWA adalah aplikasi web yang menggunakan teknologi web modern untuk memberikan pengalaman mirip dengan aplikasi yang mendukung multi platform dan responsif kepada pengguna. PWA menawarkan fitur seperti instalasi ke layar utama, penyimpanan konten dalam *cache*, timbal balik ketika koneksi internet terputus, dan berbagai fungsi lainnya [3], [4]. Salah satu keunggulan lain yang dimiliki oleh PWA adalah dapat diakses melalui peramban web yang berbasis *chromium*, dengan itu memudahkan pengguna untuk melakukan instalasi aplikasi melalui *website* yang dituju tanpa batasan perangkat yang digunakan dan tidak perlu mengunduh melalui penyedia layanan aplikasi [5]. Hal ini memungkinkan pengguna untuk menjalankan platform tanpa perlu mengunjungi *website* melalui peramban web [6].

Dalam penelitian [7] menunjukkan bahwa telah berhasil mengimplementasikan PWA dengan menggunakan metode *prototype* dengan hasil implementasi pengguna dapat melakukan instalasi di layar utama, tampilan responsif

dan fitur *offline* pada *website* E-Reses DPRD Kabupaten Sukabumi untuk meningkatkan efisiensi dan pengalaman pengguna melalui pengujian nilai kepuasan dengan stakeholder.

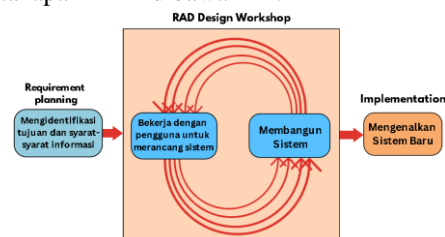
Penelitian selanjutnya [8] membahas tentang perancangan sistem informasi pelayanan jasa laundry sepatu berbasis Progressive Web App (PWA) yang bertujuan untuk mempermudah akses aplikasi secara *online* dan *offline* dengan *service worker* dan *push notification*. Metode penelitian yang digunakan adalah analisis kualitatif dengan fokus pada proses bisnis usaha laundry Sepatu. Sistem Informasi Pelayanan Jasa Usaha Laundry Sepatu berbasis PWA merupakan sistem terkomputerisasi yang menggunakan teknologi terbaru, memudahkan pelayanan kepada pelanggan dengan fitur-fitur yang diperlukan. Penggunaan sistem ini diharapkan dapat meningkatkan kinerja bisnis, image branding, potensi pasar baru, dan loyalitas pelanggan.

Namun, masih terdapat kelemahan dan kekosongan dalam penelitian Implementasi *Progressive Web App* Pada *Website* E-Reses DPRD Kabupaten Sukabumi dengan Metode *Prototype* tersebut, dari penelitian tersebut yang menyatakan bahwa implementasi PWA berhasil, namun tidak ada pengujian yang dilakukan seperti pengujian *installable* dan *optimized* PWA menggunakan pengujian *tool* dari peramban web untuk memastikan apakah PWA sudah memenuhi semua persyaratan di pengujian *tool* dari peramban web dan belum menggunakan *service worker* agar dapat merubah tampilan pemberitahuan *website* yang sedang *offline* ketika tidak ada jaringan *internet*, yang sebelumnya tidak ada di penelitian terdahulu pada *website* E-Reses DPRD Kabupaten Sukabumi. Untuk penelitian pada perancangan sistem informasi pelayanan jasa usaha laundry sepatu berbasis *Progressive Web App* (PWA) masih terdapat kekurangan tidak adanya instalasi menggunakan manifest *JSON* untuk ditambahkan ke layar utama dan tidak ada penggunaan *service worker* untuk merubah tampilan halaman *offline* ketika tidak ada jaringan internet.

Dengan demikian, penelitian ini akan merubah tampilan *offline* ketika tidak ada jaringan internet menggunakan *service worker*, melakukan pengujian *tool* dari peramban web untuk memenuhi syarat-syarat PWA pada platform freelance digital, dan menggunakan manifest *JSON* untuk instalasi PWA ke layar utama tanpa harus mengunduh di penyedia layanan aplikasi pada perangkat pengguna agar memudahkan akses ke platform freelance digital kepada pengguna seperti mitra dan *freelancer* tanpa harus mencari alamat *website* secara berulang setiap ingin mengunjungi platform freelance digital.

II. METODE

Dalam mendukung penelitian ini, digunakan metode *Rapid Application Development* (RAD). Pendekatan ini bertujuan untuk mempersingkat waktu yang dibutuhkan tanpa membuang waktu yang berlebihan [9]. Tahapan yang digunakan dalam implementasi metode RAD melibatkan *Requirements Planning*, *Design Workshop*, dan *implementation* [10]. Berikut adalah tahapan RAD dibawah ini.

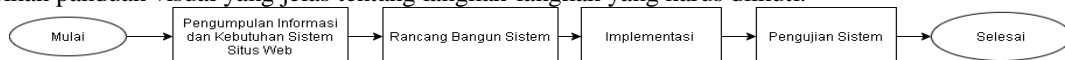


Gambar 1. Metode RAD

- Requirements planning* merupakan proses analisis yang lebih mendalam untuk memahami kebutuhan apa yang digunakan oleh pengguna di platform freelance digital, serta fitur yang akan digunakan sesuai dengan kebutuhan pengguna dan pengembang serta tujuan dari fitur berdasarkan kebutuhan yang telah di analisis [11].
- Design Workshop* merupakan tahap merencanakan arsitektur *design* berdasarkan dari hasil analisis yang telah dilakukan [12].
- Implementation* merupakan tahapan yang menghasilkan keluaran sistem dan merupakan fase di mana program akan diuji menggunakan pengujian *tool* dari peramban web, pengujian *black box*, dan hasil dari program tersebut akan diuji cobakan langsung kepada pengguna berupa kuesioner [13].

A. Desain Alur Pengembangan PWA

Flowchart adalah gambaran visual dari alur prosedur penelitian yang dijalankan secara sistematis. Dengan menggunakan *flowchart* ini, rangkaian proses implementasi PWA dapat dilakukan dengan lebih mudah karena memberikan panduan visual yang jelas tentang langkah-langkah yang harus diikuti.



Gambar 2. Diagram Alir Pengembangan PWA

Diagram tersebut menggambarkan alur pengembangan PWA. Dimulai dengan pengumpulan informasi dan kebutuhan sistem, serta penentuan penggunaan fitur yang diperlukan. Setelah itu, rancang bangun sistem dibuat berdasarkan tahapan sebelumnya. Lalu dilakukan implementasi untuk menghasilkan keluaran sistem. Selanjutnya sistem diuji secara menyeluruh untuk memastikan fungsionalitasnya, setelah itu selesai.

III. HASIL DAN PEMBAHASAN

A. Requirements Planning

Tahap ini melakukan pengumpulan data sistem dan kebutuhan sistem *website* serta fitur yang akan diintegrasikan PWA.

1. Pengumpulan informasi dan kebutuhan sistem situs web

Berdasarkan kebutuhan situs web dari hasil observasi yang telah dilakukan dari 2 sudut pandang pengguna dan pengembang, Berikut merupakan daftar kebutuhan sistem situs web :

Tabel 1. Pengumpulan informasi dan kebutuhan sistem web

Sudut pandang	Keterangan
Pengguna	Mempermudah akses ke platform <i>freelance</i> digital berbasis <i>website</i> tanpa perlu mencari alamat website secara berulang.
Pengembang	Memberikan tampilan yang lebih responsif. Mengubah tampilan halaman pemberitahuan <i>offline</i> kepada pengguna ketika tidak ada koneksi <i>internet</i> .

Berikut adalah fitur yang akan digunakan untuk pengembangan PWA berdasarkan kebutuhan pengguna dan pengembang:

a) Manifest JSON

Manifest *JSON* menentukan nama, deskripsi, ikon, warna tema, dan *URL* awal di dalam *manifest JSON*. Manifest *JSON* adalah sebuah berkas *JSON* yang memberitahu *browser* mengenai aplikasi web yang akan dijalankan, termasuk perilaku saat diinstal pada perangkat *android* dan laptop. Biasanya berisi informasi seperti nama aplikasi, deskripsi, ikon, warna tema, dan *URL* awal situs web [14]. Dengan adanya fitur manifest *JSON* di platform *freelance* digital berguna untuk melakukan instalasi PWA di layar utama agar pengguna tidak perlu mencari halaman *website* secara berulang.

b) Bingkai Kerja Tailwind

Memberikan pengalaman mirip dengan aplikasi yang mendukung multi platform dengan membuat tampilan yang responsif pada perangkat *android* dan laptop diperlukan *framework tailwind* yang akan diuji menggunakan pengujian *tool* dari peramban web. *Tailwind CSS* merupakan *framework CSS revolutioner* dengan konsep *utility-first*, di mana Anda tidak perlu lagi terpaku pada nama kelas yang panjang untuk komponen *HTML*. Berbeda dengan *framework CSS* tradisional seperti *Bootstrap*, *Foundation*, atau *Bulma* yang menyediakan komponen siap pakai, *Tailwind CSS* menawarkan kelas-kelas kecil yang fleksibel [15].

c) Service Worker

Service worker memiliki peran penting dalam menjalankan proses PWA di platform *freelance* digital. Saat pengguna pertama kali membuka *website*, *service worker* akan terpasang dan aktif di latar belakang [16]. *Service worker* merupakan alat yang berguna pada PWA yang memberikan penyimpanan konten dalam *cache*, *respons* jika tidak ada *internet*, serta berbagai fungsi lainnya [17]. *Service worker* berguna untuk mengubah tampilan halaman pemberitahuan *offline* kepada pengguna ketika tidak ada koneksi internet digunakan strategi *cache first*.

Strategi *caching cache first* adalah pendekatan di mana data yang diminta akan diambil dari *cache* terlebih dahulu jika tersedia. Jika data tidak ada dalam *cache*, maka permintaan akan dilakukan ke jaringan. *Respons* dari jaringan akan disimpan ke dalam *cache* dan kemudian ditampilkan kepada pengguna [18]. Strategi ini cocok digunakan untuk *cache offline.php* dan *offline.png* untuk mengubah tampilan halaman pemberitahuan *offline* kepada pengguna ketika tidak ada koneksi *internet*.

B. Design Workshop

1. Rancang Bangun Sistem

Rancang Bangun PWA



Gambar 3. Rancang bangun PWA

Pada gambar 3 rancang bangun PWA menjelaskan bahwa pada perangkat *mobile* di *android* dan *laptop* mengakses *web app platform freelance digital* lalu pengguna bisa melakukan *install pwa* dengan manifest *JSON* untuk di *install* di layar utama dengan tampilan yang responsif menggunakan *tailwinds*, lalu *service worker* akan bekerja setelah pengguna mengakses platform *freelance digital* di latar belakang yang akan menyimpan *offline.png* dan *offline.php* di *cache storage* untuk mengubah tampilan halaman pemberitahuan *offline* kepada pengguna ketika tidak ada koneksi internet, ketika kondisi *online* maka akan tampil seperti biasa data dari yang ditampilkan kepada pengguna [19].

C. Implementation

1. Implementasi

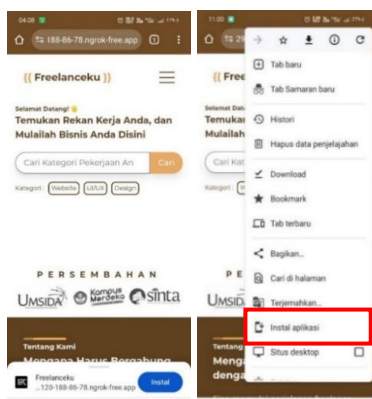
Perangkat yang digunakan untuk Pengujian dan Implementasi:

Perangkat pengujian dan Implementasi dibawah ini digunakan untuk melakukan berbagai pengujian seperti pengujian *tool* dari peramban web, pengujian *add to home screen* atau *install app*, pengujian responsif dan pengujian *service worker*, dan juga digunakan untuk implementasi pada perangkat dibawah ini. Berikut pada tabel 2 perangkat-perangkat yang digunakan untuk pengujian dan implementasi dibawah ini:

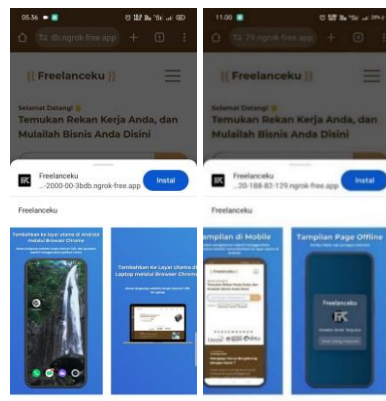
Tabel 2. perangkat pengujian dan implementasi yang digunakan

Perangkat	Sistem Operasi	Browser
Laptop	Windows 10	chrome
Mobile	Android	chrome

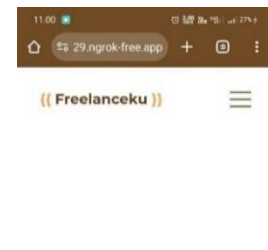
Berikut hasil implementasi PWA:



Gambar 4. Tampilan melakukan instalasi pwa ke layar utama

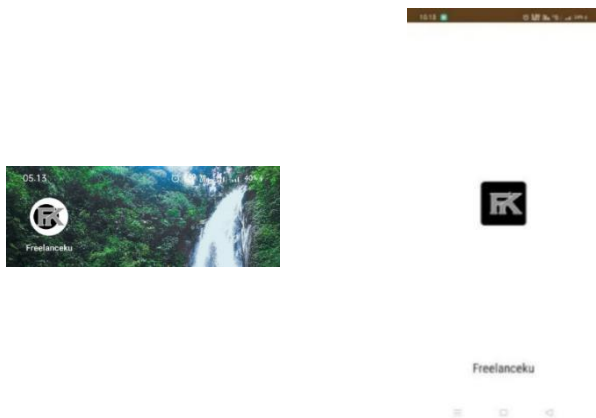


Gambar 5. Tampilan screenshot saat akan melakukan Instalasi



Gambar 6. Tampilan Address bar

Gambar 4 menjelaskan bahwa PWA dapat diinstall di *pop up* seperti pada gambar 4 sebelah kiri, atau di *install* seperti pada gambar 4 di sebelah kanan. PWA dapat di *install* pada perangkat *mobile* di *android* dan *laptop* melalui browser *chrome* di perangkat pengguna dengan menggunakan fitur manifest *JSON* dengan tujuan untuk mempermudah pengguna dalam akses platform *freelance digital* tanpa perlu mengunduh melalui penyedia layanan aplikasi dan tidak perlu mencari alamat *website* secara berulang. Gambar 5 menampilkan 4 gambar *screenshot* di manifest *JSON* saat melakukan instalasi PWA di layar utama untuk memberikan informasi fitur PWA. Pada gambar 6 warna *address bar* berubah sesuai dengan warna yang ditentukan di platform *freelance digital*.



Gambar 7. Muncul ikon setelah instalasi

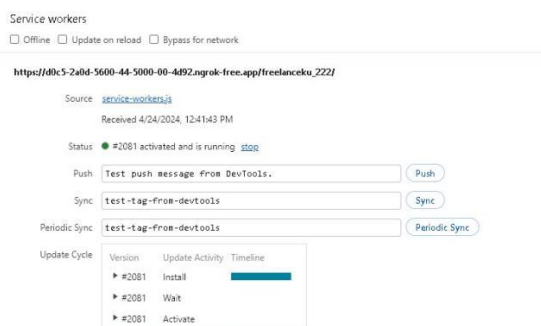


Gambar 8. Tampilan *splash screen*

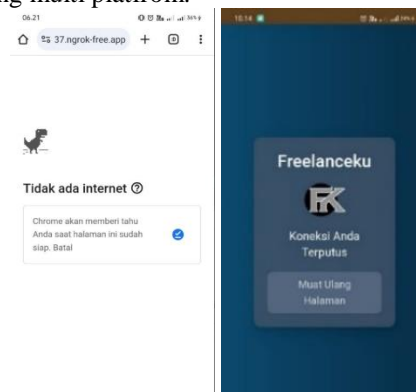


Gambar 9. Tampilan responsif pada perangkat *mobile* di *android*

Setelah melakukan instalasi ke layar utama pada gambar 7, maka akan muncul ikon untuk membuka platform *freelance* digital yang otomatis di *maskable*, dan mempermudah pengguna dalam akses platform *freelance* digital tanpa perlu mengunduh melalui penyedia layanan aplikasi dan tidak perlu mencari *URL website*, cukup menekan ikon yang telah diinstal maka platform *freelance* digital bisa diakses. Pada Gambar 8, *splash screen* secara otomatis terpasang dengan memenuhi syarat yaitu dengan menambahkan nama, warna tema dan ikon di manifest *JSON*. Saat PWA telah dibuka melalui ikon yang telah di instalasi ke layar utama pada perangkat *mobile* di *android* melalui *browser chrome* pada gambar 9, maka pengguna akan mendapatkan tampilan yang responsif di perangkat *mobile* dan mendapatkan pengalaman mirip dengan aplikasi yang mendukung multi platform.

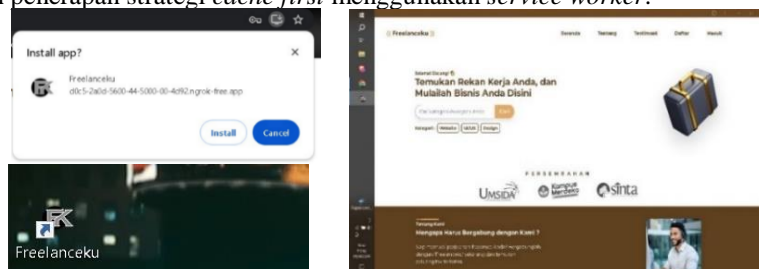


Gambar 10. Tampilan *service worker* yang telah aktif



Gambar 11. Tampilan *website* sebelum dan sesudah ketika tidak ada jaringan internet

Pada gambar 10 menjelaskan bahwa, *service worker* yang telah terinstal dan aktif maka akan menyimpan *cache* yang diminta yaitu *offline.php* dan *offline.png* di dalam *cache storage* untuk mengubah tampilan halaman pemberitahuan offline kepada pengguna ketika tidak ada koneksi *internet*. Pada gambar 11 diatas adalah hasil dari *custom offline page* yang bertujuan untuk memberikan pemberitahuan *offline* kepada pengguna ketika tidak ada jaringan *internet*, untuk gambar 11 di sebelah kiri adalah sebelum menerapkan *service worker* dan gambar 11 di sebelah kanan setelah penerapan strategi *cache first* menggunakan *service worker*.



Gambar 12. Tampilan Instalasi, ikon, dan tampilan responsif di laptop

Tampilan pada gambar 12 merupakan tampilan melakukan instalasi di layar utama di laptop melalui browser *chrome*, setelah itu membuka ikon pwa yang telah di *install* dan menyajikan tampilan yang responsif.

Hasil Implementasi

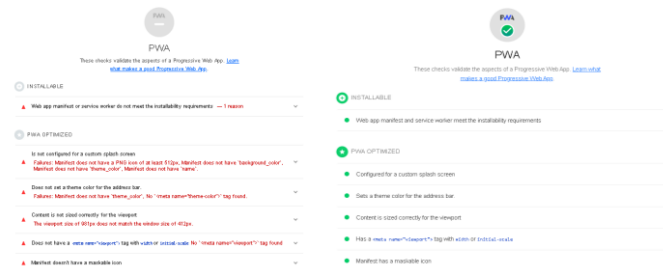
Hasil implemementasi menunjukkan bahwa PWA telah berhasil di implementasikan dengan baik menggunakan fitur manifest *JSON* yang dapat dinstal di layar utama melalui *android* dan laptop, menampilkan *screenshot* saat akan diinstal, merubah warna *adress bar*, munculnya ikon di layar utama, dan menampilkan *splash screen*, sehingga memudahkan pengguna mengakses platfrom *freelance* digital dengan instalasi dan munculnya ikon di layar utama. Tampilan menjadi responsif di *android* dan laptop menggunakan *tailwinds*. *Service worker* berhasil terinstal dan aktif di layar belakang serta merubah tampilan halaman pemberitahuan *offline* kepada pengguna ketika tidak ada jaringan *internet*.

2. Pengujian Sistem

Pada tahap ini pengujian PWA dilakukan menggunakan pengujian *tool* dari peramban web, pengujian manual instalasi di layar utama, pengujian responsif PWA yang telah di *install*, pengujian *service worker* dengan *black box*, dan uji respon pengguna.

Pengujian dengan *black box* bertujuan untuk memastikan apakah instalasi di layar utama, pengujian responsif pwa yang telah di *install* pada perangkat di *android* dan laptop untuk memastikan instalasi dan responsif sudah bekerja dengan baik, serta pengujian *service worker* untuk memastikan *service worker* terinstal dan aktif di latar belakang, dan memastikan mengubah tampilan halaman pemberitahuan *offline* ketika tidak ada koneksi internet.

a) Pengujian *tool* dari peramban web



Gambar 13. Hasil pengujian *tool* dari peramban web sebelum dan sesudah.

Pengujian ini bertujuan untuk mengetahui apakah platform *freelance* digital telah memenuhi semua syarat pada PWA menggunakan pengujian *tool* dari peramban web. Seperti pada gambar 13 di sebelah kiri yang belum menerapkan PWA dan belum memenuhi semua persyaratan PWA di pengujian *tool* dari peramban web pada platform *freelance* digital, dan di sebelah kanan pada gambar 13 setelah menerapkan PWA dan memenuhi semua syarat PWA di pengujian *tool* dari peramban web. Berikut beberapa syarat-syarat pwa pada tabel 3 didalam pengujian *tool* dari peramban web:

Tabel 3. Hasil PWA di pengujian *tool* dari peramban web

No	Nama Pengujian	Hasil yang di harapkan	Sebelum	Setelah
Installable				
1.	Web App Manifest dan <i>Service Worker</i> telah terpasang	Fitur manifest <i>JSON</i> terpasang dan dapat di instal di laptop dan <i>android</i> . <i>Service worker</i> telah terpasang dan aktif di latar belakang.	Belum Terpenuhi	Terpenuhi
PWA Optimized				
2.	<i>Splash screen</i> pada perangkat <i>android</i> .	Tampil <i>splash screen</i> saat membuka ikon di perangkat <i>mobile</i> di <i>android</i> .	Belum Terpenuhi	Terpenuhi
3.	Mengatur warna tema pada <i>address bar</i> .	Warna <i>address bar</i> muncul dan sesuai dengan yang di atur di dalam platform <i>freelance</i> digital.	Belum Terpenuhi	Terpenuhi
4.	Konten memiliki ukuran yang benar untuk <i>viewport</i> .	Tampilan konten <i>website</i> memiliki ukuran yang benar sesuai dengan layar atau memiliki tampilan responsif di laptop dan <i>android</i> .	Belum Terpenuhi	Terpenuhi
5.	Memiliki <i>tag</i> <code><meta name="viewport"></code> dengan lebar atau skala awal.	<i>Tag</i> <code><meta name="viewport"></code> telah terpasang dengan benar sehingga membuat tampilan di laptop dan <i>android</i> menjadi responsif.	Belum Terpenuhi	Terpenuhi
6.	Ikon manifest dapat di <i>maskable</i> .	Ikon telah otomatis <i>maskable</i> saat sudah terinstal di layar utama	Belum Terpenuhi	Terpenuhi

Hasil pengujian *tool* dari peramban web diatas menunjukkan bahwa semua persyaratan PWA telah terpenuhi di perangkat *android* dan laptop, yang sebelumnya belum terpenuhi dan berdampak pada beberapa fitur PWA yang tidak berfungsi.

b) Pengujian manual melakukan Instalasi di layar utama:

Setelah PWA berhasil memenuhi semua syarat di pengujian *tool* dari peramban web pada gambar 13. Pada pengujian manual ini dilakukan Instalasi di layar utama pada perangkat yang telah di tentukan yaitu laptop dan *android*. Berikut pada tabel 4 pengujian manual dibawah ini.

Tabel 4. Pengujian *black box* penerapan PWA melakukan Instalasi di layar utama

No	Bentuk Pengujian	Hasil yang diinginkan	Hasil uji	Simpulan
1.	Melakukan Instalasi di layar utama melalui <i>chrome browser</i> di <i>android</i>	Terdapat tombol <i>install</i> di <i>browser chrome</i> dan Ikon tampil setelah melakukan instalasi di <i>android</i> melalui <i>browser chrome</i>	Berhasil	Berhasil melakukan instalasi di <i>browser chrome</i> dan menampilkan ikon setelah instalasi pada perangkat <i>android</i>
2.	Melakukan Instalasi di layar utama melalui <i>chrome browser</i> di Laptop.	Terdapat tombol <i>install</i> di <i>browser chrome</i> dan Ikon tampil setelah melakukan instalasi di laptop melalui <i>browser chrome</i>	Berhasil	Berhasil melakukan instalasi di <i>browser chrome</i> dan menampilkan ikon setelah instalasi pada perangkat laptop

Dari hasil pengujian *black box* diatas penerapan pwa untuk melakukan instalasi di layar utama telah berhasil berjalan dengan baik di *android* maupun laptop melalui *browser chrome* dengan menampilkan tombol *install* dan tampilnya ikon di perangkat pengguna.

c) Pengujian responsif pwa yang telah terinstal:

Pada pengujian responsif pwa yang telah *terinstall* ini dilakukan Instalasi di layar utama pada perangkat yang telah di tentukan yaitu laptop dan *android* lalu membuka pwa yang telah terinstal di layar utama. Berikut pada tabel 5 pengujian responsif PWA yang telah terinstal dibawah ini.

Tabel 5. Pengujian responsif PWA yang telah *terinstall* dengan *black box*

No	Bentuk Pengujian	Hasil yang diinginkan	Hasil uji	Simpulan	
1.	Membuka pwa yang telah <i>terinstall</i> di laptop	Bentuk menyesuaikan perangkat	aplikasi ukuran	Berhasil	Berhasil menampilkan aplikasi yang responsif sesuai bentuk perangkat
2.	Membuka pwa yang telah <i>terinstall</i> di <i>android</i>	Bentuk menyesuaikan perangkat	aplikasi ukuran	Berhasil	Berhasil menampilkan aplikasi yang responsif sesuai bentuk perangkat

Dari hasil pengujian *black box* diatas untuk responsif PWA yang telah *terinstall*, berhasil berjalan dengan baik di *android* maupun laptop melalui *browser chrome* dengan menampilkan aplikasi yang responsif sesuai bentuk perangkat pengguna.

d) Pengujian Service Worker:

Pada pengujian dibawah ini menunjukkan pengujian *service worker* dengan *black box*.. Berikut hasil pengujian *service worker* pada tabel 6.

Tabel 6. Pengujian *black box* penerapan *service worker*

No	Bentuk Pengujian	Hasil yang diinginkan	Hasil uji	Simpulan
1.	Bentuk <i>service worker</i> ketika telah <i>terinstall</i> dan aktif di sebuah <i>browser</i>	<i>Service worker terinstall</i> dan aktif di <i>browser</i>	Berhasil	Berhasil Menginstall dan mengaktifkan <i>services worker</i> pada <i>browser</i>
2.	Halaman yang memberikan peringatan jika sedang <i>offline</i>	Tampil halaman <i>offline page</i>	Berhasil	Berhasil menampilkan halaman <i>offline page</i> ketika tidak ada koneksi <i>internet</i>

Dari hasil pengujian *black box* diatas untuk pengujian *black box* penerapan *service worker* telah berhasil berjalan dengan baik dengan terinstalnya *service worker* dan aktifnya *service worker* di latar belakang sehingga menampilkan halaman *offline* yang telah dirubah tampilanya di perangkat pengguna saat tidak ada koneksi internet.

d) Hasil Uji Respon Pengguna

Pada tahap uji respon pengguna dilakukan untuk mengukur kepuasan penggunaan PWA pada platform freelance digital. Metode pengujian melibatkan penyebaran kuesioner berupa *Google Form* kepada 10 responden yang berasal dari kalangan masyarakat umum. Proses perhitungan dilakukan dengan menggunakan skala Likert untuk menganalisis tanggapan responden terhadap PWA di platform freelance digital. Langkah-langkah yang diambil dalam perhitungan uji respon pengguna mengikuti rumus skala likert sebagai berikut [20].

$$P = \frac{N \times R}{I} \times 100\%$$

Keterangan :

P = Nilai Presentase Tiap Soal

N = Nilai setiap jawaban pada masing-masing instrumen

R = Frekuensi Nilai Terjawab

I = Nilai jawaban tertinggi dikali dengan jumlah peserta



Gambar 14. Jumlah Jawaban Responden

Telah didapatkan jawaban dari 10 responden melalui *Google Form*. Hasil dari kuesioner tertera dan dapat dilihat pada Tabel 7.

Tabel 7. Hasil Kuesioner

Pertanyaan	1	2	3	4	5
Apakah PWA memudahkan akses dalam menggunakan platform <i>freelance</i> digital tanpa mencari <i>URL</i> dan tanpa perlu mengunduh melalui penyedia layanan aplikasi?	0	0	1	3	6
Apakah fitur PWA sudah berfungsi dengan baik seperti pada informasi <i>screenshot</i> ketika ingin melakukan instalasi PWA?	0	0	0	3	7
Apakah tampilan platform <i>freelance</i> digital sudah responsif di perangkat <i>android</i> dan <i>laptop</i> ?	0	0	0	2	8
Apakah tampilan pemberitahuan <i>offline</i> sudah berubah ketika tidak ada jaringan <i>internet</i> ?	0	0	1	1	8

Dari hasil kuesioner, kriteria interpretasi akan dicari berdasarkan perhitungan interval seperti yang terdokumentasikan dalam Tabel 8.

Tabel 8. Interval Interpretasi Indeks

Nilai Interval	Hasil
0% - 19,99%	Sangat Kurang
20% - 39,99%	Kurang
40% - 59,99%	Cukup
60% - 79,99%	Bagus
80% - 100%	Sangat Bagus

Indeks interpretasi hasil perhitungan dari setiap pertanyaan terdapat pada table berikut :

Tabel 9. Interval Interpretasi Indeks

Pertanyaan	Nilai Interval	Hasil
Apakah PWA memudahkan akses dalam menggunakan platform <i>freelance</i> digital tanpa mencari <i>URL</i> dan tanpa perlu mengunduh melalui penyedia layanan aplikasi?	90%	Sangat Bagus
Apakah fitur PWA sudah berfungsi dengan baik seperti pada informasi <i>screenshot</i> ketika ingin melakukan instalasi PWA?	94%	Sangat Bagus
Apakah tampilan platform <i>freelance</i> digital sudah responsif di perangkat <i>android</i> dan <i>laptop</i> ?	96%	Sangat Bagus
Apakah tampilan pemberitahuan <i>offline</i> sudah berubah ketika tidak ada jaringan <i>internet</i> ?	94%	Sangat Bagus

Berdasarkan hasil uji respon pengguna menggunakan kuesioner telah didapat presentase rata rata 93,5%. Berdasarkan Nilai Interval Interpretasi Indeks presentase tersebut berada di kategori "Sangat Bagus".

IV. SIMPULAN

Penelitian ini bertujuan untuk mengimplementasikan *Progressive Web Apps* (PWA) pada platform *freelance* digital dengan menggunakan metode *Rapid Application Development* (RAD). Hasil penelitian menunjukkan bahwa PWA telah berhasil diimplementasikan pada platform *freelance* digital. Hal ini dibuktikan dengan hasil pengujian *tool* dari peramban web, pengujian manual instalasi di layar utama, pengujian responsif, dan pengujian *service worker* dengan *black box*. Hasil uji respon pengguna menunjukkan bahwa PWA diterima dengan baik oleh pengguna. Hal ini dibuktikan dengan hasil kuesioner yang menunjukkan bahwa presentase rata-rata kepuasan pengguna adalah 93,5% dengan kategori "Sangat Bagus". Penelitian ini juga berhasil mengatasi kekurangan yang ada dalam penelitian sebelumnya, seperti pada penelitian Implementasi Progressive Web App pada Website E-Reses DPRD Kabupaten Sukabumi yang tidak melakukan pengujian terkait *installable* dan *optimized* PWA menggunakan pengujian *tool* peramban web. Dalam penelitian ini, pengujian tersebut dilakukan untuk memastikan bahwa PWA memenuhi semua persyaratan di pengujian *tool* peramban web, termasuk fitur *installable* dan *optimized* PWA. Selain itu, penelitian ini juga memperbaiki kekurangan dalam penelitian sistem informasi pelayanan jasa usaha laundry sepatu berbasis PWA yang tidak menggunakan manifest *JSON* untuk instalasi ke layar utama dan tidak menggunakan *service worker* untuk merubah tampilan pemberitahuan *offline* ketika tidak ada jaringan *internet*. Implementasi PWA dalam penelitian ini mencakup penggunaan manifest *JSON* untuk instalasi di layar utama tanpa harus mengunduh melalui penyedia layanan aplikasi dan memanfaatkan *service worker* untuk mengubah tampilan halaman *offline* ketika tidak ada koneksi internet, serta berhasil memenuhi semua syarat PWA menggunakan pengujian *tool* peramban web.

V. Saran

Penelitian ini dapat dikembangkan dengan menguji PWA pada perangkat yang lebih beragam. Saran yang kedua, PWA juga dapat dikembangkan dengan menambahkan fitur-fitur baru yang lebih menarik. Saran yang ketiga, Platform *freelance* digital lainnya dapat menerapkan PWA untuk meningkatkan kemudahan akses dan kepuasan pengguna.

REFERENSI

- [1] N. Dalimunthe and M. A. Fajri, 'Analisis Status Pekerja Freelance dalam Perjanjian Kerja Waktu Tertentu (PKWT) dalam Perspektif Hukum Ketenagakerjaan di Indonesia', *AHKAM*, vol. 2, no. 3, pp. 482–497, Jun. 2023, doi: 10.58578/ahkam.v2i3.1297.
- [2] H. L. Aqil, 'Ilusi Kemitraan dalam Wacana Ekonomi Berbagi', *Jurnal Dinamika*, vol. 4, no. 1, Art. no. 1, Jul. 2023, doi: 10.18326/dinamika.v4i1.1-12.
- [3] N. P. Kiswanto, S. D. E. Paturusi, and V. Tulenan, 'Aplikasi E-Log Book Penangkapan Ikan Menggunakan Progressive Web App', *Jurnal Teknik Informatika*, vol. 15, no. 2, Art. no. 2, Jun. 2020, doi: 10.35793/jti.15.2.2020.29597.
- [4] T. Karli, A. Muawwal, and M. S. S. Thayf, 'IMPLEMENTASI PROGRESSIVE WEB APPLICATION PADA WEBSITE FRIZFOO MENGGUNAKAN EXPRESS.JS', *KHARISMA Tech*, vol. 18, no. 2, Art. no. 2, Oct. 2023, doi: 10.55645/kharismatech.v18i2.445.
- [5] H. A. Billah and I. K. D. Nuryana, 'Penerapan Progressive Web Apps untuk Pengembangan Fitur Push Notification dan Multi-Platform Installable pada Aplikasi Beasiswa', *Journal of Informatics and Computer Science (JINACS)*, vol. 5, no. 01, pp. 7–15, Jul. 2023.
- [6] A. Amrullah, Y. Salim, and A. R. Manga, 'Implementasi Progressive Web App Terhadap Aplikasi E-Commerce Sebagai Solusi Untuk Meningkatkan Kinerja Aplikasi Berbasis Web', *Buletin Sistem Informasi dan Teknologi Islam (BUSITI)*, vol. 2, no. 3, Art. no. 3, Aug. 2021, doi: 10.33096/busiti.v2i3.912.
- [7] M. D. Ramdhani, A. Asriyanik, and A. Pambudi, 'Implementasi Progressive Web App Pada Website E-Reses DPRD Kabupaten Sukabumi dengan Metode Prototipe', *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 22, no. 2, Art. no. 2, Aug. 2023, doi: 10.53513/jis.v22i2.8426.
- [8] M. J. Budiman and N. Yudianthi, 'PERANCANGAN SISTEM INFORMASI PELAYANAN JASA USAHA LAUNDRY SEPATU BERBASIS PROGRESSIVE WEB APP (PWA)', *Prosiding Seminar Nasional Penelitian LPPM UMJ*, vol. 1, no. 1, Art. no. 1, Oct. 2022, Accessed: May 22, 2024. [Online]. Available: <https://jurnal.umj.ac.id/index.php/semnaslit/article/view/14249>
- [9] R. T. Aldisa, 'Penerapan Metode RAD (Rapid Application Development) Pada Sistem Informasi Promosi dan Pemesanan Makanan Berbasis Website Studi Kasus Restoran Waroenk Anak Kuliah', *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 3, Art. no. 3, Dec. 2021, doi: 10.47065/bits.v3i3.1137.
- [10] M. Y. Putra and R. W. R. Lolly, 'Sistem Aplikasi Penjualan Souvenir Berbasis Web Menggunakan Metode Rapid Application Development (RAD)', *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS: Journal of Information System*, vol. 5, no. 2, pp. 151–160, Jun. 2021, doi: 10.51211/isbi.v5i2.1548.

- [11] D. Susanti and E. Elmiyati, 'Perancangan Website Media Informasi dan Pemesanan pada PT. Trita Musi Prasada dengan Metode RAD', *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 1, Art. no. 1, Sep. 2020, doi: 10.30812/matrik.v20i1.723.
- [12] M. P. Puteri and H. Effendi, 'Implementasi Metode RAD Pada Website Service Guide "Tour Waterfall South Sumatera"', *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 7, no. 2, Art. no. 2, Sep. 2018, doi: 10.32736/sisfokom.v7i2.570.
- [13] S. Suhardi, A. R. Rizqiah, N. Ichsan, R. Ratnawati, L. Hakim, and S. Ayumida, 'SISTEM INFORMASI ALUMNI BERBASIS WEB DENGAN METODE RAPID APPLICATION DEVELOPMENT', *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, Art. no. 1, Jan. 2024, doi: 10.31602/tji.v15i1.13587.
- [14] M. Hapif, S. Kosasi, S. M. Kuway, and T. Wijaya, 'Pembuatan dan Pengujian Website Penjualan Elektronik Berbasis Progressive Web Application', vol. 10, no. 2, 2021.
- [15] R. J. Romadhondaru and A. Basuki, 'Visualisasi Topologi Jaringan berdasarkan Data Routing Border Gateway Protocol', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, Art. no. 9, Sep. 2022.
- [16] R. P. Piro, Y. Sholva, and H. Novriando, 'Aplikasi Pelaporan Kecelakaan Lalu Lintas Menggunakan Metode Location Based Service Berbasis Progressive Web App', *JURISTI (Jurnal Riset Sains dan Teknologi Informatika)*, vol. 1, no. 1, pp. 56–68, Feb. 2023.
- [17] N. Nurwanto, 'Penerapan Progressive Web Application (PWA) pada E-Commerce', *Techno.Com*, vol. 18, no. 3, Art. no. 3, Aug. 2019, doi: 10.33633/tc.v18i3.2400.
- [18] M. A. Fikri, A. Primajaya, and M. Jajuli, 'PENERAPAN PROGRESSIVE WEB APP PADA PEMBUATAN WEBSITE MAGANG STUDI KASUS PRODI INFORMATIKA UNSIKA', *INFOTECH journal*, vol. 9, no. 2, Art. no. 2, Oct. 2023, doi: 10.31949/infotech.v9i2.7059.
- [19] N. M. Adelin, 'Implementasi Progressive Web Apps (PWA) Untuk Meningkatkan Kinerja dan Performa Situs Maritimepreneur', *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Art. no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3904.
- [20] I. A. Kautsar and M. Ruslianor Maika, 'The use of User-centered Design Canvas for Rapid Prototyping', *J. Phys.: Conf. Ser.*, vol. 1764, no. 1, p. 012175, Feb. 2021, doi: 10.1088/1742-6596/1764/1/012175.

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.