

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2. 1 Tinjauan Pustaka

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Adapun hasil-hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian yaitu pemanfaatan pwa dengan framework laravel.

Berdasarkan hasil penelitian yang dilakukan Juanda B (2018) dimana melakukan penelitian mengenai implementasi *progressive web app* untuk meningkatkan kinerja Aplikasi Pengingat Jadwal Taklim. Jenis penelitian yang digunakan adalah metode penelitian deskriptif kualitatif, jenis penelitian deskriptif yang digunakan adalah *Design and Creation*. Sedangkan metode pengumpulan data yang digunakan adalah menggunakan metode pengumpulan data wawancara, dokumentasi, studi literatur dan observasi. Metode perancangan yang digunakan adalah *Waterfall* dan Teknik pengujian menggunakan *blackbox*. Hasil dari pengujian aplikasi menyimpulkan bahwa fungsi yang diharapkan semuanya berhasil.

Bahari dan Yusuf (2019) menyajikan penelitian mengenai penerapan *progressive web apps* pada Aplikasi Lowongan Pekerjaan Dosen Universitas Perjuangan. Pada penelitian ini peneliti menggunakan teknologi *Progressive Web Apps (PWA)* dan menggunakan *framework laravel*. Metode pengembangan yang digunakan untuk aplikasi lowongan pekerjaan dosen Universitas Perjuangan adalah metode *waterfall*. metode *waterfall* memiliki tahapan yang dimulai dari

requirement definition, system and software design, implementation and unit testing, integration and system testing dan *operation and maintenance*. Adapun hasil pengujian dengan menggunakan *black box* sudah sesuai dengan yang diharapkan. Penerapan sistem ini menggunakan teknologi *service worker* sehingga dapat diakses dalam keadaan koneksi internet tidak stabil maupun offline.

Noor dan Pahrul (2020) menyajikan penelitian mengenai Implementasi *Progressive Web Apps (PWA)* Menggunakan *Laravel* Dan *Vue.js* Dalam Pembuatan Aplikasi Penyedia Jasa *Freelance*. Berdasarkan hasil pembahasan tentang Implementasi Teknologi PWA terhadap pembuatan aplikasi penyedia jasa *freelance* ini dapat menjadi sebuah sarana bagi publik dan calon pekerja agar dapat bertemu dan melakukan transaksi. Maka berdasarkan data kuesioner yang didapat dari para responden, dapat diambil kesimpulan bahwa penelitian yang dilakukan berjalan dengan sukses dan dirasa dapat membantu para lulusan SMK 3 Mataram untuk mendapatkan pekerjaan dan juga pengalaman kerja.

Aripin dan Soemantri (2021) menyajikan penelitian mengenai Implementasi *Progressive Web Apps (PWA)* pada Repository *E-Portofolio* Mahasiswa. Pengembangan perangkat lunak yang dilakukan pada penelitian ini menggunakan metode *Waterfall*, Sedangkan perancangan sistem yang digunakan adalah berbasis *Progressive Web Apps (PWA)* yang merupakan teknologi website modern yang membuat website dapat dijalankan seperti aplikasi native. Hasil pengujian menunjukkan sistem sudah berjalan normal, mudah digunakan, dan memenuhi kriteria *progressive web apps* seperti dapat dijalankan secara offline dan dapat diakses melalui home screen.

Pada penelitian ini, Demih (2022) memanfaatkan *Progressive Web App (PWA)* sebagai *landing page profil* Desa Bigwaimod Catubuow, Pegunungan Arfak, Papua Barat dengan tujuan untuk meningkatkan aksesibilitas dan keterjangkauan informasi profil desa. PWA dikembangkan untuk menyediakan konten statis tentang profil Desa Bigwaimod, termasuk informasi berita, *event* dan galeri foto. Dengan mengimplementasikan fitur *caching* (mekanisme yang digunakan untuk menyimpan data atau sumber daya (seperti halaman *web*, gambar, atau data lainnya) secara sementara di lokasi penyimpanan lokal). PWA memungkinkan pengguna untuk mengakses informasi tersebut secara *offline* saat tidak ada koneksi internet, dan data disinkronkan kembali dengan *server* ketika koneksi internet tersedia kembali. Penelitian ini diharapkan dapat memberikan kontribusi positif dalam memperluas akses informasi dan pengenalan potensi desa kepada masyarakat yang tinggal di lokasi dengan keterbatasan akses internet.

Adapun perbandingan dari penelitian-penelitian terdahulu dapat dilihat pada tabel di bawah ini.

Tabel 1. 1 Tabel Perbandingan dengan Penelitian-penelitian Sebelumnya

Peneliti	Parameter		
	Tujuan	Metode	Hasil
Juanda B (2018)	Meningkatkan kinerja Aplikasi Peningat Jadwal Taklim	Metode perancangan menggunakan <i>Waterfall</i> dan Teknik pengujian menggunakan <i>blackbox</i>	Fungsi yang diharapkan semuanya berhasil
Bahari dan Yusuf (2019)	Penerapan <i>progressive web apps</i> pada Aplikasi Lowongan Pekerjaan Dosen Universitas Perjuangan	Metode perancangan menggunakan <i>Waterfall</i> dan Teknik pengujian menggunakan <i>blackbox</i>	Hasil pengujian dengan menggunakan <i>black box</i> sudah sesuai dengan yang diharapkan
Noor dan Pahrul (2020)	Pembuatan Aplikasi Penyedia	Metode perancangan	Penelitian yang dilakukan

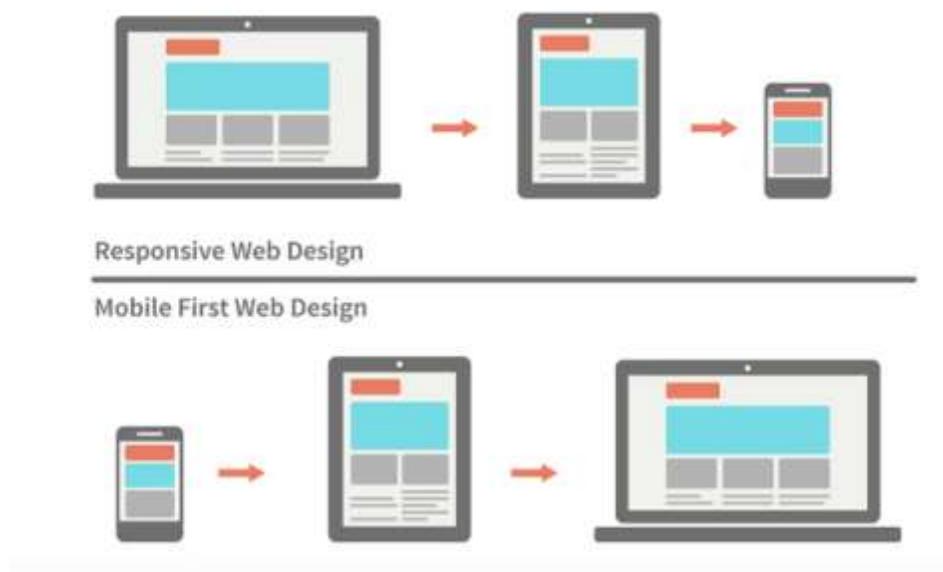
	Jasa <i>Freelance</i>	menggunakan <i>Waterfall</i> dan Teknik pengujian menggunakan <i>blackbox</i>	berjalan dengan sukses dan dapat membantu lulusan SMK 3 Mataram mendapatkan pekerjaan dan juga pengalaman kerja
Aripin dan Soemantri (2021)	Implementasi <i>Progressive Web Apps (PWA)</i> pada Repository <i>E-Portofolio</i> Mahasiswa	Metode perancangan menggunakan <i>Waterfall</i> dan Teknik pengujian menggunakan <i>blackbox</i>	Sistem sudah berjalan normal, mudah digunakan, dan memenuhi kriteria <i>progressive web apps</i> .
Demih (2022)	Memanfaatkan <i>Progressive Web App</i> sebagai <i>landing page profil</i> Desa Bigwaimod, Catubuow, Pegunungan Arfak, Papua Barat	Metode perancangan menggunakan <i>Waterfall</i> dan Teknik pengujian menggunakan <i>blackbox</i>	Diharapkan <i>landing page profil</i> dapat berjalan normal dan dapat dijalankan secara offline.

2. 2 Dasar Teori

2. 2. 1 *Progressive Web App*

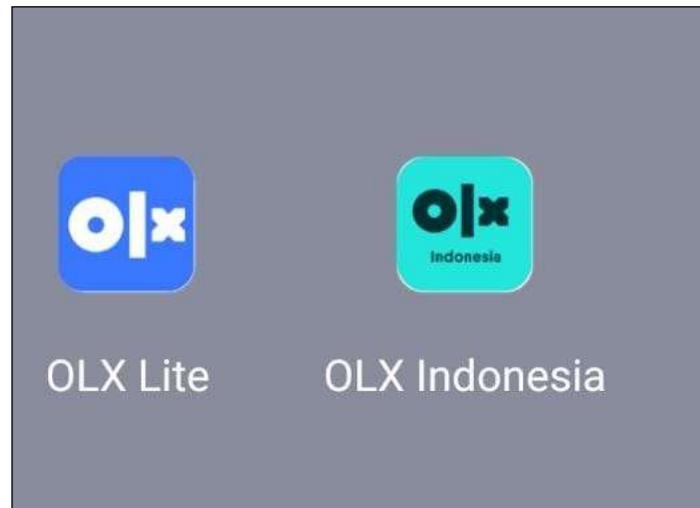
Progressive Web Application (PWA) adalah sebuah *website* yang dibangun menggunakan teknologi *web* modern, namun dapat berlaku seperti sebuah *mobile app* (Santosa 2019 : page 3-5). Berikut poin penting yang membuat aplikasi *web* bisa disebut sebagai *Progressive Web App* diantara sebagai berikut:

- a. *Responsif (Responsive)*: PWA akan menyesuaikan diri dengan berbagai ukuran layar, termasuk ponsel, tablet, dan desktop.



Gambar 2. 1 *Contoh Responsif (Responsive) tampilan mobile, tablet dan desktop*

- b. Seperti Aplikasi (*App Like*): PWA memberikan pengalaman yang mirip dengan aplikasi *native*, termasuk ikon beranda, akses *offline*.



Gambar 2. 2 Contoh App Like pada OLX

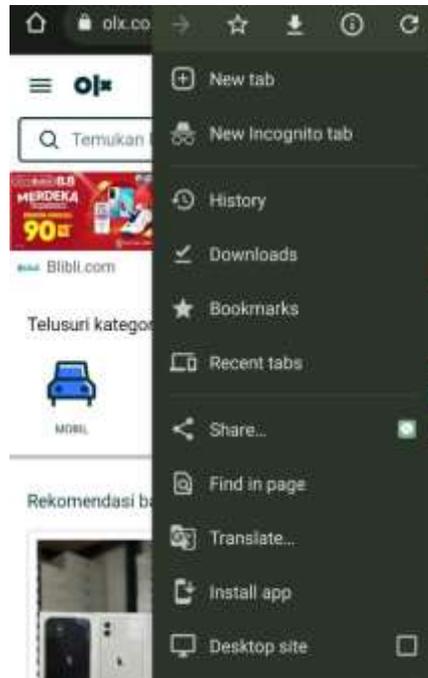
Olx lite adalah versi PWA yang telah di *install* dengan *install app* via *browser* sedangkan *olx* Indonesia adalah aplikasi *native*.

- c. Akses *Offline*: PWA dapat menyimpan data dan konten sebelumnya, memungkinkan pengguna untuk mengakses aplikasi bahkan ketika mereka tidak terhubung ke internet.



Gambar 2. 3 Contoh akses online pada PWA

- d. *Instalable*: Pengguna dapat "menginstal" PWA pada layar beranda perangkat mereka, sehingga mereka dapat mengaksesnya seperti aplikasi seluler tanpa mengunjungi situs web lagi.



Gambar 2. 4 Contoh *instalable* pada PWA

Pwa ini akan di gunakan untuk halaman *landing page* yang bisa diakses *public* sedangkan fitur yang dikelola admin tidak menggunakan pwa. Seperti dijabarkan diatas PWA mampu mengatasi saat akses data yang sudah tersimpan di *local storage* bias diakses secara *offline* dan ketika *online* data terupdate, sehingga masih dapat membaca informasi walaupun tidak ada internet.

2. 2. 2 Website

Website adalah kumpulan halaman *web* yang diletakan dalam satu tempat atau *site* maupun situs. Jadi di dalam halaman-halaman *web*, beserta *file –file* pendukungnya, seperti gambar, video dan *file* digital lainnya. Sebuah *website* pasti memiliki alamat tersendiri, yang disebut dengan nama *domain (domain name)* dan alamat IP (*IP address*). (Tim EMS, 2012).

Secara umum, situs *web* digolongkan menjadi 3 jenis yaitu sebagai berikut:

- a. *Website Statis* adalah *web* yang mempunyai halaman tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari situs tersebut.
- b. *Website Dinamis* merupakan *website* yang secara struktur diperuntukan untuk *update* sesering mungkin. Biasanya selain halaman utama yang bisa diakses oleh *user* pada umumnya, juga disediakan halaman admin untuk mengedit konten.
- c. *Website Interaktif* adalah *web* yang saat ini sedang booming. Salah satu contoh *website interaktif* adalah *blog* dan forum. Di *website* ini user bisa berinteraksi dan beradu argumen mengenai apa yang menjadi pemikiran mereka.

2. 2. 3 API

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau

dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan developers untuk membuat aplikasi. Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat *fitur* yang serupa. Penerapan *API* akan sangat terasa jika *fitur* yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: *integrasi* dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk *sistem operasi*, *library*, dan *web* (Sandi, 2018).

API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan layer dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi. Contoh yang dapat menggambarkan spesifikasi tersebut adalah *POSIX* (*Portable Operating System Interface*). Dengan menggunakan standar *POSIX*, aplikasi yang *di-compile* untuk bekerja pada sistem operasi tertentu juga dapat bekerja pada sistem lain yang memiliki kriteria yang sama. *Software library* juga memiliki peran penting dalam menciptakan *compatibility* antar sistem yang berbeda.

Aplikasi yang berinteraksi dengan *library* harus mengikuti serangkaian aturan yang ditentukan oleh *API*. Pendekatan ini memudahkan *software developer* untuk membuat aplikasi yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut *API 25* yang sama. Kelebihan lain dari metode ini menunjukkan betapa mudahnya

menggunakan *library* yang sama dengan bahasa pemrograman yang berbeda. Seperti namanya, *Web API* dalam diakses melalui *protokol HTTP*, ini adalah konsep bukan teknologi. Kita bisa membuat *Web API* dengan menggunakan teknologi yang berbeda seperti *PHP*, *Java*, *.NET*, dll. Misalnya *Rest API* dari *Twitter* menyediakan akses *read* dan *write* data dengan mengintegrasikan *twitter* kedalam aplikasi kita sendiri.

2. 2. 4 Service Worker

Pada umumnya sebuah *website* tidak dapat ditampilkan secara *offline* namun tidak demikian jika *website* menerapkan *service worker* pada *website*. *Service worker* adalah kode *Javascript* yang dijalankan oleh *browser* di latar belakang, dan terpisah dengan skrip lainnya. *Service worker* memiliki kemampuan untuk menyimpan *aset-aset* sebuah *website* seperti *file HTML*, *CSS*, *Javascript*, dan gambar di penyimpanan *browser*, sehingga ketika *website* tersebut diakses tanpa koneksi internet *service worker* akan mengembalikan *aset website* yang sudah disimpan sebelumnya, hal ini membuat sebuah *website* dapat bekerja layaknya aplikasi *native Android* dan *desktop*. Dengan kemampuan menyimpan *asset website* secara lokal, *service worker* juga dapat meningkatkan kecepatan akses sebuah *website* dan menghemat data (Kurniawan, 2020).

2. 2. 5 Framework Laravel

Laravel adalah *framework* pembuatan *website* yang dibuat oleh *Taylor Otwell* yang dibangun dengan konsep *MVC (Model- View-Controller)*, dan dilengkapi juga *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging*

bundle dan instalasi *bundle* melalui *command prompt*. *Laravel* dibangun menggunakan bahasa pemrograman PHP yang cukup populer di Indonesia. Selain itu Jika dibandingkan dengan *framework* PHP lainnya *Laravel* memiliki banyak keunggulan seperti: *auth scaffolding* yang memudahkan pembuatan *fitur login* dan *register* secara otomatis tanpa harus menulis kode program, dan *Eloquent Orm* untuk mengelola *database* tanpa menggunakan bahasa *SQL*, dan cocok untuk membuat layanan *restfull API* dan *Repository online* (Yudhanto dan Helmi, 2018)

2. 2. 6 Landing Page

Landing page pada dasarnya adalah sebuah halaman *website* yang ditemui oleh pengunjung *internet* pada saat pertama kali seseorang ketika akan melakukan sebuah aksi penting yang diinginkan oleh pemilik *website* (Ash, et al., 2012). *Landing page* biasanya membuat pengguna fokus pada suatu informasi. Bisa dikatakan bahwa *landing page* berfungsi untuk menampilkan suatu informasi dengan akhir supaya pengguna tersebut melakukan aksi tertentu.