

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Berikut ini beberapa tinjauan pustaka yang dilakukan oleh beberapa peneliti-peneliti sebelumnya.

Penelitian yang dilakukan oleh Kurniawan, Arif Rizki (2018) dengan judul “Penerapan Progressive Web Apps pada Aplikasi Lowongan Pekerjaan Dengan Teknologi Server Worker (Studi Kasus Akakom Carrer Center)”. Aplikasi dibuat untuk membantu mendapatkan informasi pekerjaan bagi pencari pekerjaan di lingkungan STIMIK AKAKOM.

Penelitian selanjutnya dilakukan oleh Awal Kuriawan (2017) dengan judul “Implementasi Progressive Web App Pada Sistem monitoring Keluhan Sampah Kota Makassar”. Aplikasi berfungsi sebagai sarana monitoring keluhan warga terhadap sampah di Kota Makassar.

Penelitian selanjutnya dilakukan oleh James Riady dkk, (2019) dengan judul “Aplikasi E-Learning Berbasis Progressive Web App Pada Apologetika Indonesia”. pada penelitian tersebut aplikasi dapat menjalankan proses kursus secara online melalui website.

Penelitian selanjutnya dilakukan oleh Leurensius Adi (2017) dengan judul “Platform E-Learning untuk pembelajaran Pemograman Web menggunakan Konsep PWA”. Membuat platform pembelajaran E-learning untuk pemograman Web.

Selanjutnya penelitian yang di lakukan oleh Mandayana Patasik, (2018) dengan judul “Perancangan Progressive Web App Transportasi di Kota Makassar”. Aplikasi ini berfungsi untuk memberikan informasi mengenai transportasi di kota Makassar.

Selanjutnya penelitian yang di lakukan oleh Maikelinus M B Iba, (2020) dengan judul “Implementasi Progressive Web App pada sistem pendaftaran siswa baru”. Aplikasi ini berfungsi untuk melakukan pendaftaran siswa baru secara online.

Berikut ini adalah table mengenai perbedaan penelitian yang dilakukan sebelumnya dibuat dalam table 2.1:

**Tabel 2.1 Tinjau Pustaka**

Nama	Topik	Objek	Metode	Hasil
Kurniawan, Arif Rizki (2018)	Penerapan <i>Progressive Web Apps</i> pada Aplikasi Lowongan Pekerjaan Dengan Teknologi <i>Server Worker</i> (Studi Kasus Akakom Carrer Center)	Lowongan pekerjaan	<i>Progressive Web App</i>	Website pencari pekerja pada Akakom Carrer Center
Awal Kuriawan (2017)	Implementasi <i>Progressive Web App</i> Pada Sistem monitoring Keluhan Sampah Kota Makassar	Sampah	<i>Progressive Web App</i>	Monitoring keluhan sampah
James Riady dkk, (2019)	Aplikasi <i>E-Learning</i> Berbasis <i>Progressive Web App</i> Pada Apologetika Indonesia	Kursus	<i>Server Worker</i>	Aplikasi dapat mewadahi dan menjalankan proses kursus secara <i>online</i>

				melalui website.
Leurensius Adi (2017)	Platform <i>E-Learning</i> untuk pembelajaran Pemograman <i>Web</i> menggunakan Konsep <i>PWA</i>	Metode Pembelajaran	<i>Server Worker</i> <i>dan Manifest</i>	<i>platform e-Learning</i> pembelajaran mata kuliah Pemrograman <i>Web</i>
Mandyana Patasik, (2018)	Perancangan <i>Progressive Web App</i> Transportasi di Kota Makassar	Transportasi	<i>Progressive Web App</i>	memberikan informasi mengenai transportasi yang ada di kota makassar.
Maikelinus M B Iba, (2020)	Implementasi <i>Progressive Web App</i> pada sistem pendaftaran siswa baru	Pendaftaran siswa baru	<i>Progressive Web App</i>	Siswa baru dapat melakukan pendaftaran melalui website
Domi Donatus Meage, (2022)	Penerapan <i>Progressive Web App</i> pada sistem Pemesanan Citra Batik Wamena	Pemesanan toko online	<i>Progressive Web App</i>	Pelanggan dapat melakukan pemesanan melalui website.

## 2.2 Dasar Teori

### 2.2.1 Pengenalan Progressive Web App (PWA)

#### 1. Latar Belakang Progressive Web App (PWA)

*Progressive Web App* adalah sebuah teknologi yang memungkinkan sebuah situs web saluler dapat digunakan seperti asli. Ketika dikembangkan, *PWA* akan menggunakan sejumlah teknologi, *arsitektur app shel*, *service worker*, *API*, serta fitur *web modern* lainnya agar *PWA* ini dapat bekerja seperti aplikasi asli/native pada umumnya. Berbeda dengan *hybrid*, aplikasi *hybrid* lebih menggabungkan antara aplikasi native dan web. *PWA* ini sepenuhnya *web* namun pengguna bisa merasakan pengalaman seperti menggunakan aplikasi native. Seperti bisa dijalankan offline atau koneksi buruk sekalipun, bahkan menggunakan animasi yang smooth seperti aplikasi *native*. Aplikasinya pun tidak perlu diinstal, namun ikon-nya bisa si pasang di *home screen* untuk memudahkan membuka aplikasi. *PWA* sepenuhnya mengandalkan *browser*.

Sebuah *website* dapat dianggap sebagai *PWA* jika memenuhi persyaratan tertentu atau memiliki beberapa fitur diaplikasikan di dalamnya. Beberapa diantaranya seperti adanya fitur *push notifications*, dapat diinstal, bekerja secara *offline*, mudah disinkronkan, dan masih banyak lagi.

## 2. Karakteristik

Selain itu, perlu diketahui bahwa *Progressive Web App* memiliki karakteristik tertentu, yaitu :

### *a. Progressive*

*PWA* masih dapat bekerja di browser lama namun akan bekerja secara optimal jika diakses menggunakan web browser terbaru.

### *b. Discoverable*

Konten didalam *PWA* dapat ditemukan dihalaman search engine.

### *c. Linkable*

Memudahkan dibagikan melalui link URL.

### *d. Network independent*

Tetap dapat diakses secara offline atau dengan koneksi internet yang buruk.

### *e. Installable*

*PWA* dapat ditampilkan pada home screen perangkat.

Selain itu, mampu mempergunakan app switcher untuk beralih berasal dari satu aplikasi ke aplikasi yang lain sehingga mampu merasakan bahwa *PWA* bekerja seperti native yang di unduh dan instal di perangkat.

### *a. Re-engageable*

Meningkatkan keterlibatan bersama dengan pengguna karena punya kemampuan untuk mengirim notifikasi pemberitahuan saat ada info atau konten baru.

***b. Responsive***

Mampu berfungsi bersama dengan baik di perangkat apapun bersama dengan beragam ukuran layar.

***c. Safe***

Menggunakan protocol HTTPS sehingga mampu hindari akses berasal dari pihak yang tidak sah.

**3. Kelebihan dan kekurangan PWA**

Teknologi PWA membawa pengaruh positif pada pengembangan aplikasi web. Namun teknologi PWA ini masih terbilang baru dan masih perlu pengembangan lanjut.

**a. Kelebihan**

**1) Mendapat banyak pengguna**

Perusahaan yang hanya mengembangkan aplikasi *native* mampu beroleh lebih banyak user bersama dengan menyediakan PWA.

**2) Meningkatkan konversasi**

PWA termasuk sangat mungkin perusahaan untuk mengirimkan notifikasi atau pemberitahuan untuk penggunanya.

**3) Tidak perlu instalasi**

Aplikasi *native* pada umumnya perlu diunduh dan dipasang lebih-lebih dahulu berasal dari App Store. PWA bekerja tanpa instalasi dan mampu dibagikan bersama dengan orang lain melalui URL.

**4) Membutuhkan sedikit area penyimpanan**

Aplikasi *native* umumnya membutuhkan data store yang besar saat diinstal diperangkat. Berbeda dengan *PWA* yang membutuhkan sedikit area penyimpanan.

#### 5) **Lebih irit cost dan tenaga**

Sistem *PWA* dikembangkan hanya sekali, sedangkan aplikasi *native* membutuhkan dua pengembangan yang dijalankan secara terpisah yaitu iOS dan Android.

#### **b. Kekurangan**

##### 1) **Beberapa fitur tidak ada di iOS**

Ketika digunakan di iOS, fitur *add to home screen* dan push notification tidak mampu ditampilkan.

##### 2) **Tidak mempunyai fitur sebaik aplikasi native**

Jika dibandingkan dengan *native*, *PWA* punya fitur yang jauh lebih sedikit.

##### 3) **Tidak terdapat di App Store**

Perilaku lebih dari satu besar pengguna masih mencari aplikasi pada app store. Karena *PWA* tidak didistribusikan melalui app store, hal ini membuat *PWA* sedikit sulit untuk di dapatkan.

#### **4. Teknologi Pendukung *Progressive Web App***

Teknologi yang umum digunakan pada *Progressive Web App (PWA)* meliputi :

### a. Service worker

Saat ini ada teknologi *Progressive Web App (PWA)* dengan *Service Worker*-nya yang memungkinkan kita untuk membuka aplikasi web secara *offline*. Ada beberapa kemampuan dari *service worker* yang bekerja untuk Aplikasi Web. Yang paling berperan adalah kemampuan fungsi *offline*.

*Service Worker* adalah salah satu jenis *Web Worker*, *JavaScript* yang berjalan di-background tanpa mempengaruhi kinerja halaman web. *Service worker* pada dasarnya adalah file *JavaScript* yang berjalan di client side secara terpisah dari rangkaian browser utama, berfungsi mencegat permintaan jaringan, melakukan cache atau mengambil sumber daya dari cache, dan mengirimkan pesan.

### b. Web App Manifest

*Web app manifest* merupakan W3C specification yang mendefinisikan *JSON based manifest* untuk memberikan developer tempat untuk meletakkan metadata dari aplikasi web, terdiri dari :

- 1) Nama aplikasi web
- 2) Link ke icon atau obyek gambar *web app*
- 3) Target *URL* untuk membuka (*launch*) *web app*
- 4) Data konfigurasi web app seperti display, orientasi, dan lainnya

Dengan setting file *web manifest*, *developer* memungkinkan pengguna untuk menciptakan pengalaman seperti aplikasi mobile native melalui aplikasi web progresif. Penambahan detail tentang *web app manifest* akan dibahas tersendiri.

### **2.2.2 Framework Codeigniter**

Framework atau dalam bahasa Indonesia dapat diartikan sebagai “Kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal. Alasan mengapa menggunakan Framework.

1. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam framework (dengan syarat programmer mengikuti pola standard yang ada).
2. Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll).
3. Mempercepat dan mempermudah membangun sebuah aplikasi web.
4. Lebih bebas dalam pengembangan jika dibandingkan CMS

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis.

CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi codenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework”.

### **2.2.3 Bootstrap Framework**

Bootstrap adalah kerangka kerja CSS yang bersifat open source dan digunakan untuk kebutuhan pembuatan tampilan desain visual dari aplikasi web atau situs website. Kerangka kerja yang digunakan berbentuk template desain berbasis HTML dan CSS untuk kebutuhan pengembangan navigasi, tombol, tipografi, formulir, dan komponen antarmuka yang lainnya.

Selain itu, *Bootstrap* juga memiliki fitur yang mencakup *library* dari *JavaScript*. Untuk penggunaan dari *framework* ini digunakan untuk membantu dalam menyusun program aplikasi pada sisi front end (*client – side*). Untuk sekarang, *Bootstrap* sangat diminati oleh berbagai

pengembang web melalui platform *Github* untuk membantu proses pembuatan desain aplikasi atau website yang lebih komprehensif dan modern.

#### **2.2.4 JavaScript**

*JavaScript* adalah bahasa *script* yang berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen *HTML*. Dimana objek tersebut dapat berupa suatu *window*, *frame*, *URL*, dokumen, *form*, *button* atau item yang lain. Yang semuanya itu mempunyai properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai dan atribut lain.

#### **2.2.5 PHP**

*PHP* atau *Hypertext Preprocessor* adalah bahasa pemrograman yang berjalan pada *server side scripting* dan bersifat open source (sumber terbuka). Untuk penggunaan dari bahasa ini sering digunakan untuk pembuatan aplikasi berbasis website yang berjalan secara dinamis, sehingga dapat terintegrasi dengan basis data (database).

PHP banyak digunakan untuk pembuatan website untuk kebutuhan ecommerce, sistem informasi, maupun landing page. Karena tergolong dalam bahasa pemrograman berbasis *server-side*, maka scrip yang digunakan akan diproses oleh server. Jenis server yang sering dipakai adalah Apache, Nginx, dan LiteSpeed.

### **2.2.6 MySQL**

*MySQL* adalah sebuah *DBMS (Database Management System)* menggunakan perintah *SQL (Structured Query Language)* yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. *MySQL* dibagi menjadi dua lisensi, pertama adalah *Free Software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya.

*MySQL* termasuk dalam *RDBMS (Relational Database Management System)*. Sehingga, menggunakan table, kolom, baris, di dalam struktur database-nya. Jadi, dalam proses pengambilan data menggunakan metode relational database. Dan juga menjadi penghubung antara perangkat lunak dan database server.

### **2.2.7 Unified Modeling Language (UML)**

*UML* adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awalmulanya, *UML* diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997.

*UML* juga dapat didefinisikan sebagai suatu bahasa standard visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standard penulisan blueprint sebuah software.

UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk factor-faktor scalability, robustness, security, dan sebagainya.

### **2.2.8 XAMPP**

XAMP adalah sebuah paket perangkat lunak (*software*) computer yang sistem penamannya diambil dari akronim kata *Apache*, *MySQL* (dulu) / *MariaDB* (sekarang), *PHP*, dan *Perl*. sementara huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai symbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti *OS Linux*, *OS Windows*, *Mac OS*, dan *Solaris*.

Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui Xampp control panel, atau istilah website offline.

### **2.2.9 JSON**

*JSON (JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar Edisi ke-3 - Desember 1999. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C

termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data. *JSON* terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

