

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2. 1. Tinjauan Pustaka

Dalam penelitian ini dilakukan berdasarkan beberapa Pustaka terdahulu yang menjadi hak dasar dan pedoman pendukung serta pembanding dalam pembuatan skripsi ini diantaranya :

Afif Rizki Kurniawan (2018) dengan judul “Penerapan *Progressive Web Apps*” pada Aplikasi Lowongan Pekerjaan dengan Teknologi *Service Worker* (Studi Kasus: Akakom *Carrer Center*). Penelitian ini menggunakan metode PWA dengan teknologi *Service Worker* yang bertujuan agar dapat membuat web yang *responsive*, web yang dapat berjalan pada kondisi jaringan internet yang buruk atau bahkan dalam keadaan *offline*, dan dapat mempermudah para pelamar pekerjaan (alumni STMIK Akakom Yogyakarta) untuk mendapatkan pekerjaan mereka.

Krisna Viktoria Iriyanto (2019) dengan judul “Implementasi *Progressive Web Apps* pada Aplikasi Persediaan Burung dengan Teknologi *Service Worker* (Studi Kasus: Lintang *Bird Farm*)”. Penelitian ini menggunakan metode PWA dengan teknologi *Service Worker* yang bertujuan agar Aplikasi *inventory* tersebut dapat diakses secara *offline* di beberapa halaman tertentu serta dapat melakukan sinkronisasi secara otomatis dari data *offline* dan data *online* Ketika jaringan internet tersedia.

Tegar Pribadi (2019) dengan judul “Penerapan *Progressive Web Apps* pada Aplikasi *Inventory* Barang menggunakan Teknologi *Service Worker* (Study Kasus Dinas Sosial Kota Yogyakarta)”. Penelitian ini menggunakan metode PWA dengan teknologi *Service Worker* yang bertujuan agar aplikasi *Inventoru* Barang dapat diakses secara *offline* (jaringan buruk) dan *online* (jaringan bagus), serta aplikasi yang dibuat ini secara otomatis akan melakukan sinkronisasi dari data *offline* dan data *online* pada saat jaringan internet terhubung.

Ahmad Mawanto (2021) dengan judul “Penerapan *Progressive Web Apps* pada Aplikasi Pengelolaan Keuangan Siswa dengan Teknologi *Service Worker* (Studi Kasus: SMA Negeri 01 Hulu Gurung)”. Penelitian ini menggunakan metode PWA dengan teknologi *Service Worker* yang bertujuan agar aplikasi yang dibuat dapat menerapkan metode *Progressive Web Apps* (PWA) dan teknologi *Service Worker*. Aplikasi yang dibuat dapat memberikan pengalaman pengguna yang lebih baik, dapat dijalankan layaknya *native app* pada perangkat *mobile*, dan jika diakses melalui *browser* pada perangkat *desktop* atau tidak di *instal* di perangkat *mobile*, maka aplikasi akan memiliki kinerja yang lebih baik dari segi kecepatan akses dibandingkan dengan menggunakan *browser* diperangkat *mobile*.

Bangin Bola Meindika (2020) dengan judul *Progressive Web Application* pada *Digital Library* Perpustakaan Daerah Kabupaten Sleman dengan bahasa pemrograman PHP dan DBMS MySQL. Penelitian ini menggunakan metode PWA dengan teknologi *Service Worker* yang bertujuan agar aplikasi dapat

mengimplementasi Progressive Web Application pada Digital Library untuk Perpustakaan Daerah Kabupaten Sleman.

Tabel 2. 1 Tabel Perbandingan Tinjauan Pustaka

No.	Penulis	Judul	Metode	Hasil
1	Kurniawan (2018)	Penerapan <i>Progressive Web Apps</i> pada Aplikasi Lowongan Pekerjaan dengan Teknologi <i>Service Worker</i> (Studi Kasus: Akakom <i>Carrer Center</i>)	<i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i>	Sistem yang dibuat merupakan web yang <i>responsive</i> , dapat berjalan pada kondisi jaringan internet yang buruk atau bahkan dalam keadaan <i>offline</i> , dan dapat mempermudah para pelamar pekerjaan (alumni STMIK Akakom Yogyakarta) untuk mendapatkan pekerjaan mereka.
2	Iriyanto (2019)	Implementasi <i>Progressive Web Apps</i> pada Aplikasi Persediaan Burung dengan Teknologi <i>Service Worker</i> (Studi Kasus: Lintang <i>Bird Farm</i>)	<i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i>	Aplikasi <i>inventory</i> ini dapat diakses secara <i>offline</i> di beberapa halaman tertentu serta dapat melakukan sinkronisasi secara otomatis dari data <i>offline</i> dan data <i>online</i> ketika jaringan internet tersedia karena menerapkan <i>Progressive Web Apps</i> (PWA) menggunakan

				teknologi <i>Service Worker</i> ..
3	Pribadi (2019)	Penerapan <i>Progressive Web Apps</i> pada Aplikasi <i>Inventory</i> Barang Menggunakan Teknologi <i>Service Worker</i> (Studi Kasus: Dinas Sosial Kota Yogyakarta)	<i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i>	Aplikasi <i>inventory</i> barang dapat diakses secara <i>offline</i> (jaringan buruk) dan <i>online</i> (jaringan bagus), serta aplikasi yang dibuat ini secara otomatis akan melakukan sinkronisasi dari data <i>offline</i> dan data <i>online</i> pada saat jaringan internet terhubung.
4	Marwanto, (2021)	Penerapan <i>Progressive Web Apps</i> pada Aplikasi Pengelolaan Keuangan Siswa dengan Teknologi <i>Service Worker</i> (Studi Kasus: SMA Negeri 01 Hulu Gurung)	<i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i>	Aplikasi yang dibuat dapat menerapkan metode <i>Progressive Web Apps</i> (PWA) dan teknologi <i>Service Worker</i> . Aplikasi yang dibuat dapat memberikan pengalaman pengguna yang lebih baik, dapat dijalankan layaknya <i>native app</i> pada perangkat <i>mobile</i> , dan jika diakses melalui <i>browser</i> pada perangkat <i>desktop</i> atau tidak

				<p>di <i>instal</i> di perangkat mobile, maka aplikasi akan memiliki kinerja yang lebih baik dari segi kecepatan akses dibandingkan dengan menggunakan <i>browser</i> diperangkat <i>mobile</i>.</p>
5	Meindika (2020)	<p><i>Progressive Web Application</i> pada <i>Digital Library</i> Perpustakaan Daerah Kabupaten Sleman</p>	<p><i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i></p>	<p>aplikasi dapat mengimplementasi <i>Progressive Web Application</i> pada <i>Digital Library</i> untuk Perpustakaan Daerah Kabupaten Sleman.</p>
6	Widyapratama (2024)	<p>Implementasi <i>Progressive Web Apps</i> dengan Teknologi <i>Service Worker</i> pada Aplikasi Kingdom Yogyakarta</p>	<p><i>Progressive Web Apps</i> dengan teknologi <i>Service Worker</i></p>	<p>Hasil yang diharapkan bahwa <i>Fans</i> dan Anggota dapat mengakses Aplikasi Kingdom Yogyakarta dengan mudah tanpa terganggu kendala jaringan internet.</p>

2. 2. Dasar Teori

2.2.1. RRQ Kingdom Yogyakarta

RRQ (*Rex Regum Qeon*) merupakan salah satu tim *E-Sport* terbesar di Indonesia yang menaungi 6 devisi game, diantaranya *Mobile Legends Bang – Bang*, *PUBG Mobile*, *Free Fire*, *Valorant*, dan *Pokemon Unite*. Didirikan sejak 2013 oleh Adrian Pauline sebagai CEO RRQ dan Riki K. Suliawan sebagai Founder RRQ. Karena merupakan salah satu tim *E-Sport* terbesar, mereka membuat suatu wadah yang bernama RRQ Kingdom.

RRQ Kingdom dibagi menjadi beberapa regional, yaitu RRQ Kingdom Jawa Tengah, RRQ Kingdom Jawa Barat, RRQ Kingdom Jawa Timur, RRQ Kingdom JABODETABEK (Jakarta, Bogor, Depok, Tangerang dan Bekasi), RRQ Kingdom Yogyakarta, RRQ Kingdom Bali, RRQ Kingdom SUMBANGSEL (Lampung dan Sumatra Selatan), RRQ Kingdom Sumatra Utara, RRQ Kingdom Sulawesi Utara, RRQ Kingdom Kalimantan Barat, dan RRQ Kingdom Kalimantan Selatan.

RRQ Kingdom Yogyakarta adalah salah satu komunitas yang menjadi wadah untuk para *fans* yang berada di Regional Daerah Istimewa Yogyakarta dibawah komunitas RRQ (*Rex Regum Qeon*) *Kingdom*. Komunitas ini berfungsi untuk menjalin kebersamaan dan kekeluargaan antar fans dimana *fans* bisa saling berkomunikasi, berbagi informasi dan mengekspresikan rasa cinta dan dukungannya kepada Team RRQ. Komunitas ini memiliki kegiatan yaitu Nobar

(Nonton Bareng), Kopdar (Kopi Darat), dan beberapa Kegiatan lainnya.

2.2.2. Website

Menurut Aziz Sholechul (2013) “Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses diseluruh dunia selama terkoneksi dengan jaringan internet. Website juga merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk di kunjungi”

2.2.3. Progressive Web Apps

Progressive Web Apps merupakan metode pengembangan perangkat lunak yang memungkinan pengguna merasakan pengalaman menggunakan aplikasi *mobile* melalui *browser*. Teknologi ini menggabungkan *Mobile Web* dan *Mobile Apps*. *Mobile Web* memiliki kelebihan dalam pembuatannya yang lebih cepat dan dapat diakses dengan mudah oleh pengguna melalui *browser* dalam berbagai *device*.

Sedangkan, *Mobile Apps* memiliki keunggulan dalam fitur yang disediakan yaitu dapat lebih kaya dan lebih interaktif, tetapi pengguna perlu melakukan installasi pada *device mobile* yang digunakan.

Pada Mei 2016, Frances Berriman dan Alex Russel merancang *Progressive Web Apps* dengan mempunyai 2 inti teknologi yaitu *Service Worker* dan *Web Apps Manifest*.

2.2.4. Web Apps Manifest

Web Apps Manifest adalah sebuah file yang berformat **.json** yang berfungsi untuk mengontrol apa saja hal yang akan terlihat pada pengguna. Struktur *Web Apps Manifest* terdiri dari beberapa bagian yaitu:

- a. *display* - Memberitahukan *browser* bahwa Website dapat dijalankan sendiri atau berdiri sendiri (*Standalone*).
- b. *start_url* - Menentukan halaman utama Ketika Web Apps dijalankan.
- c. *short_name* or *name* – *short_name* Digunakan untuk menamai Web App yang nanti akan diletakkan pada *Home Screen* atau *Launcher* atau ditempat terbatas. Sedangkan *name* digunakan pada saat muncul *Question Box* pada saat *Web App* akan diinstal.
- d. *icons* – Berfungsi untuk saat *Web App* diinstal Pengguna dapat melihat icon dari Web App pada *Home Screen*.
- e. *background_color* - Warna yang akan muncul pada *Splash Screen* ketika *Web App* pertama kali diluncurkan.
- f. *orientation* - Memberitahukan *browser* agar *Web App* mendukung Orientasi atau perputaran perangkat *Landscape* dan *Potrait*.

- g. `scope` – Mendefinisikan set URL *Web App*, halaman mana saja yang akan ditampilkan sebagai *Web App* dan secara default `scope` adalah semua URL halaman *Web App*.
- h. `theme_color` – Untuk menentukan warna *tool bar*, dan dapat dilihat dalam Aplikasi *Task Switcher*.
- i. `prefer_related_applications` – Jika diset *true*, *browser* akan meminta user menginstal *Native App* yang didefinisikan pada *property related_applications*. Jika diset *false*, *browser* akan melakukan instalasi *Web App*.
- j. `related_applications` - Berisi *array* yang didalamnya terdapat informasi aplikasi *Native* yang akan diinstal.

```

{
  "name": "nama aplikasi",
  "short-name": "nama pendek dari aplikasi",
  "start_url": "/index.html",
  "scope": ".",
  "display": "standalone",
  "background_color": "#fff",
  "theme_color": "#3A1AB1",
  "description": "deskripsi dari apps",
  "dir": "ltr",
  "lang": "en-US",
  "orientation": "portrait-primary",
  "icons": [
    {
      "src": "src/img/a48.jpg",
      "type": "image/jpeg",
      "size": "48x48"
    },
    {
      "src": "src/img/a64.jpg",
      "type": "image/jpeg",
      "size": "64x64"
    }
  ],
  "prefer_related_applications": "true",
  "related_applications": [
    {
      "platform": "play",
      "url": "https://play.google.com/store/.....",
      "id": "com.example.app"
    }
  ]
}

```

Gambar 2. 1 Contoh Source Code manifest.json

File *manifest.json* harus ditambahkan pada setiap html yang akan menggunakan fitur PWA dan disisipkan pada tag `<head>`

```

1. <link rel="manifest" href="/manifest.json">

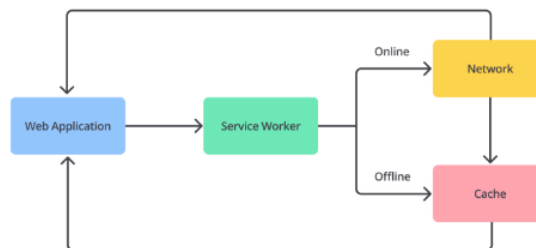
```

Gambar 2. 2 Potongan Source Code tag `<head>` pada html

2.2.5. Service Worker

Service Worker adalah salah satu Teknologi yang digunakan pada *Progressive Web Apps* yang memungkinkan *Web App* dapat

dijalankan melalui semua *browser* yang ada. Prosesnya simple dan transparan. Untuk menggunakan teknologi *Service Worker* ini, pembuat *web* perlu melakukan registrasi menggunakan *Javascript* yang ada pada suatu halaman *web* yang sedang diakses.



Gambar 2. 3 Cara Kerja Service Worker

Pada gambar tersebut kita dapat melihat bahwa *service worker* menjadi bagian yang mengatur *flow request* dan *response*. Salah satu kemampuan yang menarik dari *service worker* adalah mampu untuk mengatur strategi ketika berkomunikasi dengan internet.

2.2.6. Laravel

Laravel dirilis oleh Taylor Otwell pada tahun 2011 sebagai proyek pribadi dengan tujuan menyempurnakan *CodeIgniter* yang pada saat itu merupakan *Framework* yang terkenal dan banyak digunakan oleh para pengembang aplikasi.

Laravel adalah *framework* PHP *Open Source* yang kuat dan mudah di pahami, hal ini mengikuti pola desain model *View-Controller*. Laravel menawarkan serangkaian fungsi yang menggabungkan fitur dasar kerangka PHP seperti *CodeIgniter*, *Yoo* dan Bahasa Pemrograman lainnya seperti *Ruby on Rails*.

Laravel bertujuan untuk membuat proses pengembangan menjadi menyenangkan bagi pengembang tanpa mengorbankan fungsionalitas aplikasi. Penggunaan Laravel dalam pengembangan web dapat mencegah beberapa serangan web.

Ada beberapa fitur pada Laravel, diantaranya:

- a. *MVC Architecture* yang digunakan sebagai fitur yang paing signifikan untuk membuat lebih mnarik bagi pengembang untuk meningkatkan efisiensi halaman web dan aplikasi.
- b. *Dynamic Template* yang digunakan agar pengembang dapat memanfaatkan templete dinamis untuk merancang tata letak yang memuaskan, mengintegrasikan CSS dengan lancer dan menggabungkan kerangka kerja *Javascript* yang kuat.
- c. *Object Relation Mapping (ORM)* yang memiliki fungsi sederhana namun penting dalam pengimplementasian kode aktif PHP. Hal ini dapat memudahkan pengembang

membangun halaman web dan aplikasi online yang kuat dan skalabel.

- d. *Modular Library* yang menawarkan berbagai fitur yang mudah digunakan saat membangun halaman web dan aplikasi.
- e. *Data Migration* yang dapat dimanfaatkan untuk mengubah struktur database secara efektif menggunakan kode PHP, dikarenakan migrasi data dari satu program ke program lainnya merupakan persyaratan umum bagi pengembang.
- f. *Integrated Tools* yang dapat membuat pengembang dapat menghindari proses pemrograman yang memakan banyak waktu. Fitur ini bertujuan menyederhanakan tugas – tugas sulit, seperti memelihara beberapa sistem database dan mengelola migrasi database.
- g. *Unit Testing* yang memiliki tujuan untuk mampu menjalankan beberapa pengujian unit, menjamin semua update diimplementasikan dengan benar tanpa mengubah kode.