

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Berikut merupakan beberapa sajian tentang penelitian serupa terkait dengan penelitian ini diantaranya Abdul Hakim (2018), Ridho dkk (2017), Afif Rizki Kurniawan (2018), Adi dkk (2017) dan Frengky Adytama (2008).

Abdul Hakim (2018). Melakukan penelitian “Implementasi *Progressive Web Apps* Pada Organisasi Ikatan Mahasiswa Tanjungbalai Jogjakarta dimana penelitian tersebut, penelitian menjelaskan dan membuat suatu aplikasi berbasis web yang ditujukan untuk organisasi IMTA jogja yang dapat diakses setiap anggota, teknologi yang dibangun menggunakan pendekatan teknologi *progressive web apps* yang dapat berjalan ketika internet tidak stabil bahkan *offline*.

Afif Rizki Kurniawan (2018). Pada penelitian ini, peneliti membangun suatu aplikasi berbasis *progressive web apps* menggunakan *javascript* yang dapat membantu para pencari kerja untuk mendapatkan informasi yang mudah dan cepat, teknologi yang diterapkan yaitu *service worker* yang berfungsi untuk menyimpan *cache* pada website yang nantinya dapat berjalan oleh *browser* dalam kondisi *offline* sekalipun. Diharapkan bisa membantu para pencari pekerjaan di lingkup STMIK AKAKOM untuk melihat informasi pekerjaan walaupun konektivitas terganggu atau *offline*.

Ridho dkk (2017) dalam jurnalnya mengenai perbandingan performa *progressive web apps* dan *mobile web* terkait waktu respon, penggunaan memori dan penggunaan media penyimpanan pada suatu halaman web, Universitas Brawijaya. Pada penelitian yang dilakukan terdapat beberapa poin penting yaitu pada ukuran berkas dan *cache* yang kecil, *mobile web* masih lebih unggul dibandingkan dengan *progressive*

web apps, sedangkan pada ukuran berkas dan *cache* yang cukup besar, *progressive web apps* mampu mengungguli *mobile web*. Untuk performa terkait penggunaan memori, *mobile web* menggunakan memori lebih sedikit dibandingkan dengan *progressive web apps* dikarenakan adanya proses tambahan pada *progressive web apps* yaitu *service worker*. Sedangkan untuk performa terkait media penyimpanan pada *mobile web* tidak menggunakan media penyimpanan sama sekali, sedangkan pada *progressive web apps* penggunaan media penyimpanan menyesuaikan dengan *cache* yang disimpan pada peramban.

Adi dkk (2017) dalam jurnalnya yang meneliti mengenai *Platform E-Learning* untuk pembelajaran pemrograman web menggunakan konsep *progressive web apps* yang dapat membantu proses pembelajaran di Teknik Informatika Institut Teknologi Sepuluh November, Surabaya. Pada penelitian ini *Platform E-Learning* yang dapat diakses oleh mahasiswa untuk melihat kelas yang diikuti melihat daftar kuis, melihat daftar soal dari kuis dan menjawab soal.

Frengky Adytama (2008) dalam penelitian ini, peneliti membangun suatu aplikasi *mobile android* yang dapat mencari buku di perpustakaan STMIK Akakom menggunakan *barcode*

Dalam penelitian ini, peneliti membangun suatu aplikasi berbasis *progressive web apps* menggunakan bahasa pemrograman PHP yang dapat membantu para peminjam untuk melakukan peminjaman buku secara online, mendapatkan informasi yang mudah dan cepat, teknologi yang diterapkan yaitu *service worker* yang berfungsi untuk menyimpan *cache* pada website yang nantinya dapat berjalan oleh *browser* dalam kondisi *offline* sekalipun. Diharapkan bisa membantu para peminjam di lingkup STMIK AKAKOM untuk melihat informasi buku di perpustakaan walaupun konektivitas terganggu atau *offline*.

Tabel 2.1 Acuan Tinjauan Pustaka

Penulis	Objek	Metode	Bahasa Pemrograman	Interface
Hakim Abdul (2018)	Organisasi Ikatan Mahasiswa Tanjungbalai Yogyakarta	Progressive Web Apps	Javascript	GUI
Kurniawan Afif Rizki (2018)	ACC (AKAKOM Career Center)	Progressive Web Apps	PHP	GUI
Ridho dkk (2017)	User Interface, manajemen cache pada aplikasi, struktur berkas dan proses pertukaran data	Progressive Web Apps	-	Teks
Adi dkk (2017)	Mahasiswa ITS Jurusan Teknik Informatika	Progressive Web Apps	PHP	GUI
Frengky Adytama (2008)	Perpustakaan STMIK Akakom Yogyakarta	-	Java, Mysql	Mobile
Putri Ayu (2019)	Perpustakaan STMIK Akakom Yigyakarta	Progressive Web Apps	PHP	GUI

2.2 Dasar Teori

2.2.1 Pengertian Perpustakaan

Kata perpustakaan berasal dari kata pustaka, yang berarti: (1) kitab,buku-buku, (2) kitab primbon. Kemudian kata pustaka mendapat awalan per dan akhiran an, menjadi perpustakaan. Pengertian perpustakaan yang lebih umum menurut Sutarno (2006: 11) adalah mencakup suatu ruangan, bagian dari gedung/ bangunan, atau gedung tersendiri , yang berisi buku-buku koleksi, yang disusun dan diatur demikian rupa, sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pembaca.

Sedangkan definisi perpustakaan yang dikemukakan oleh Perpustakaan Nasional RI (2005:4) adalah unit kerja yang memiliki sumber daya manusia sekurang-kurangnya seorang pustakawan, ruangan/tempat khusus, dan koleksi bahan pustaka sekurang-kurangnya seribu judul dari berbagai disiplin ilmu yang sesuai dengan jenis dan misi perpustakaan yang bersangkutan serta dikelola menurut sistem tertentu untuk kepentingan masyarakat pengguna.

2.2.2 Digital Library

Digital library atau perpustakaan digital berkembang bersamaan dengan perkembangan internet, namun di Indonesia keberadaan digital library masih jauh dari harapan pencari informasi. Berbagai macam kendala menyebabkan keberadaannya tidak segenar perkembangan internet itu sendiri. Kendala seperti sumber daya manusia yang belum memadai, belum cukupnya konsep dan desain yang dilakukan oleh pengelola perpustakaan, sarana prasarana dan dukungan dana yang belum memadai merupakan hal yang banyak dialami oleh perpustakaan perguruan tinggi.

Onno W. Purba pada beberapa makalahnya menyampaikan definisi perpustakaan terotomasi dan perpustakaan digital sebagai berikut :

Perpustakaan terotomasi adalah perpustakaan yang seluruh fungsi kerjanya telah terintegrasi menggunakan sebuah sistem informasi elektronik. Fungsi kerjanya dapat berupa otomasi sirkulasi, otomasi katalog dan sebagainya.

Perpustakaan digital/ Digital Library adalah sebuah sistem yang memiliki berbagai layanan dan obyek informasi yang mendukung pemakai yang membutuhkan obyek informasi tersebut melalui perangkat digital. Layanan perpustakaan digital ini misalnya pencarian informasi di dalam obyek informasi seperti dokumen, gambar dan database. Secara langsung pengguna dapat memanfaatkan informasi dengan mengakses secara elektronik, tetapi secara tidak langsung obyek informasinya masih berupa kertas/buku tetapi metadatanya sudah berformat elektronik.

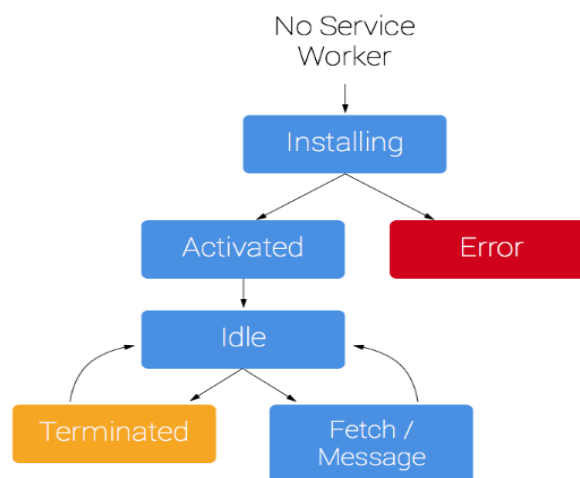
2.2.3 Progressive Web Apps

Progressive web apps adalah sebuah pengalaman yang menggabungkan yang terbaik dari aplikasi web. Seiring dengan pengguna yang semakin berhubungan dengan aplikasi dari waktu ke waktu, *progressive web apps* menjadi sangat *powerful*. Dapat memuat halaman dengan cepat, bahkan pada jaringan yang buruk sekalipun, mengirim *notification* yang *relevant*, memiliki *icon* pada layer awal dan memuat halaman sebagai pengalaman pengguna yang sangat baik. Adapun kelebihan teknologi *progressive web apps* antara lain :

- **Progressive** – bekerja untuk setiap user, terlepas dari apapun jenis browsernya karena itu dibangun dengan *progressive* yang sangat baik.
- **Responsive** – cocok dengan segala *device* desktop, seluler, tablet, dan yang lainnya.
- **Connectivity Independent** – Ditingkatkan dengan *service worker* agar aplikasi bisa *offline* atau di jaringan berkualitas rendah sekalipun.
- **App-like** – Terasa seperti aplikasi, karena model *shell* aplikasi memisahkan fungsi aplikasi dari konten aplikasi.
- **Fresh** – Selalu diperbaharui berkat pembaruan *service worker*.
- **Safe** – Dijalankan melalui *HTTPS* untuk mencegah pengintaian dan memastikan konten belum dirusak.
- **Re-engageable** – Membuat keterlibatan ulang mudah melalui fitur seperti *push notifications*.
- **Installable** – Memungkinkan pengguna menambahkan aplikasi ke *homescreen* mereka tanpa harus kesusahan harus melalui toko aplikasi seperti *playstore* dan *appstore*.
- **Linkable** – Mudah berbagi aplikasi melalui URL, tidak memerlukan instalasi yang rumit. (Lepage Pete, 2018)

2.2.4 Service Worker

Service worker adalah *script* yang dijalankan oleh browser di latar belakang, terpisah dari halaman web. Dimasa depan, *service worker* mungkin mendukung hal-hal lain seperti sinkronisasi berkala atau *geofencing*. Fitur inti dalam pembahasan ini yaitu untuk mengatasi dan mengelola permintaan jaringan termasuk mengelola *cache* dari sebuah *response*. *Service worker* bertindak sebagai perantara browser dan jaringan internet. Sebelum berjalan di jaringan internet, *request* terlebih dahulu akan berjalan melalui *service worker*. Pada kondisi ini, *web developer* dapat mengambil control penuh atas *request*. Begitu pula sebaliknya, ketika *response* jaringan, *response* tersebut akan melalui *service worker* terlebih dahulu sebelum ke browser. Berikut *life cycle* dari *service worker* :



Gambar 2.2 *Life Cycle Service Worker*

Gambar 2.2 menjelaskan *life cycle service worker* pada instalasi pertama, berikut penjelasannya :

Untuk memasang *service worker* ke situs web, maka harus mendaftarkannya. Setelah mendaftarkan *service worker*, maka browser akan melakukan instalasi *service worker* di latar belakang. Selama instalasi, bisa dilakukan penyimpanan beberapa file statis dari web. Jika semua file berhasil di *cache*, maka *service worker* akan di *install*.

Jika ada file yang gagal diunduh dan disimpan di *cache*, langkah penginstalan akan gagal dan *service worker* tidak akan diaktifkan atau tidak akan di *install*.

Ketika di *install*, langkah selanjutnya yaitu aktivasi dan disini lah akan menangani *cache* lama yang sudah tersimpan jika ada.

Setelah langkah aktivasi, *service worker* akan mengontrol semua halaman yang termasuk dalam ruang lingkungannya, meskipun halaman yang mendaftarkan *service worker* untuk pertama kali tidak akan dikontrol sampai halaman dimuat lagi. Setelah *service worker* kembali diaktifkan, ia akan berada dalam satu dari dua kondisi, baik *service worker* akan dihentikan untuk menghemat memori atau akan menangani pengambilan data dan pesan yang terjadi ketika ada yang terjadi dari jaringan atau ada pesan yang dimuat dari sebuah halaman website (Gaunt Matt, 2018).

2.2.5 HTTPS

Hypertext Transfer Protocol (HTTP) adalah protokol yang mengatur komunikasi antara *client* dan *server*. Yang menjadi *client* adalah *web browser* atau *device* lain yang dapat mengakses, menerima dan menampilkan konten web.

Pada umumnya cara komunikasi antara *client* dan *server* adalah *client* melakukan request ke *server*, kemudian *server* mengirimkan respon terhadap *client*. Respon yang dimaksud dapat berupa file HTML yang akan ditampilkan di *browser* ataupun data lain yang direquest oleh *client*. Semua kegiatan tersebut diatur oleh suatu protocol yaitu *HTTP*.

Sedangkan *Hypertext Transfer Protocol Secure (HTTPS)* adalah versi *secure* dari *HTTP* yang dikembangkan oleh *Netscape Communications Corp*. HTTPS dapat menjamin keamanan data yang ditransmisikan antara *client* dengan *server*. Ada 3 aspek yang ditangani oleh *HTTPS*, yaitu

- **Autentikasi Server**, dengan adanya autentikasi *server*, pengguna yakin sepenuhnya bahwa sedang berkomunikasi dengan *server* yang dituju.

- **Kerahasiaan Data**, data yang ditransmisikan tidak akan bisa dipahami oleh pihak lain, karena data yang ditransmisikan sudah dienkripsi.
- **Integritas Data**, data yang sedang ditransmisikan tidak dapat diubah oleh pihak lain, karena akan divalidasi oleh *message authentication code (MAC)*. (Lepage Pete, 2018)

2.2.6 JavaScript

JavaScript adalah bahasa paling populer di web dan ekosistemnya merupakan open source. <http://github.info/> memetakan jumlah repositori aktif dan secara keseluruhan bahasa program di GitHub selama beberapa tahun terakhir adalah bahasa pemrograman *JavaScript*. Meskipun web browser adalah *platform* yang paling banyak digunakan untuk *JavaScript*, *database* modern seperti MongoDB dan CouchDB menggunakan JavaScript sebagai *scripting* dan bahasa kueri. JavaScript telah menjadi *platform* yang penting di luar browser. (Antani.2016)

JavaScript adalah bahasa *script* yang ringan dan mudah digunakan. JavaScript dapat membuat halaman web tidak sekedar menjadi halaman dan informasi saja, tetapi juga data menjadi suatu program aplikasi dengan antarmuka web.