BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Afif Rizki Kurniawan (Akakom, 2018). Penelitian ini memberikan informasi untuk mencari lowongan pekerjaam yang terdapat di Akakom Carrer Center dengan dimana pelamar dari alumni STMIK Akakom Yogyakarta dengan menggunakan metode *Progressive Web Apps*.

Penelitian yang dilakukan oleh Awal Kurniawan dkk (2017). Penelitian ini memberikan informasi memanfaatkan fitur *service worker* yang mampu melakukan *caching* file hingga ratusan data keluhan, sehingga aplikasi tetap dapat dijalankan meskipun dalam keadaan *offline*.

Penelitian yang dilakukan oleh Dedie Citra Mahendra (2017). Penelitian ini memberikan informasi mengenai terbentuknya sistem alat yang dapat digunakan untuk melacak posisi serta memberikan sistem *alert* dengan modul *Global Positioning Sistem* (GPS), *Modul Global Sistem for Mobile Communication* (GSM) dan Mikrokontroler yang didukung dengan beberapa komponen elektronika guna mewujudkan *system* pelacakan dan keamanan kendaraan.

Penelitian yang dilakukan oleh Abdul Hakim (2018). Penelitian ini memberikan informasi mengenai hasil pengujian menggunakan *lighthouse* menunjukkan rata-rata nilai 100 pada kriteria *Progressive Web Apps*, nilai 100 pada kriteria *performance*, dan 100 pada kriteria *best practices*. Proses penilaian

dapat dilakukan dalam kondisi *offline* dan sudah memenuji kebutuhan. Aplikasi dapat menampilkan informasi mengenai berita, pengumuman dan anggota menggunakan database.

Penelitian yang dilakukan oleh Ridho dkk (2017). Penelitian ini memberikan informasi mengenai perbandingan performa *Progressive Web Apps* (PWA) dan *mobile* web terkait waktu respon, penggunaan memori dan penggunaan media penyimpanan pada suatu halaman *web*, Universitas Brawijaya. Pada penelitian ini didapat beberapa kesimpulan yaitu pada ukuran berkas dan *cache* yang kecil, *mobile* web masih lebih unggul dibandingkan dengan PWA, sedangkan pada ukuran berkas dan *cache* yang cukup besar PWA mampu mengungguli *mobile web*. Untuk perfoma terkait penggunaan memori, *mobile* web menggunakan memori lebih sedikit dibandingkan dengan PWA dikarenakan adanya proses tambahan pada PWA yaitu *service worker.* Sedangkan untuk performa terkait media penyimpanan, pada *mobile* web tidak menggunakan media penyimpanan sama sekali, sedangkan PWA menggunakan media penyimpanan menyesuaikan dengan *cache* yang disimpan pada peramban.

Tabel 2.1. Perbandingan Tinjauan Pustaka

No.	Penulis	Judul	Tahun	Metode	Hasil
1.	Afif Rizki Kurniawan	Penerapan *Progressive Web Apps* Pada Aplikasi Lowongan Pekerjaan dengan Teknologi *Service Worker* (Studi Kasus Akakom Carrer Center)	2018	*Progressive Web Apps* dengan teknologi *service worker*	Untuk mencari lowongan pekerjaan yang terdapat di Akakom Carrer Center dengan menggunakan metode *Progressive Web Apps*
2.	Awal Kurniawan	Implementasi *Progressive Web Application* pada Sistem Monitoring Keluhan Sampah Kota	2017	*Progressive Web Apps* dengan teknologi *service worker*	Untuk memanfaatkan fitur *service worker* yang mampu melakukan *caching* file hingga ratusan data

		Makassar			keluhan
3.	Dedie Citra Mahendra	Sistem Monitoring Mobil Rental Menggunakan GPS *Tracker*	2017	*Global Positioning Sistem* (GPS), *Modul Global Sistem for Mobile Communication* (GSM).	Untuk melacak posisi serta memberikan sistem alert dengan modul *Global Positioning Sistem* (GPS)
4.	Abdul Hakim	Implementasi *Progressive Web Apps* Pada Organisasi Ikatan Mahasiswa Tanjungbalai Jogjakarta	2018	*Progressive Web Apps*	Aplikasi dapat menampilkan informasi mengenai berita, pengumuman dan anggota menggunakan database
5.	Ridho dkk	Perbandingan Peforma *Progressive Web Apps* dan Mobile Web	2017	*Progressive Web Apps*	Membandingkan peforma *progressive web apps* dengan mobile web
6.	Yang Diusulkan	Implementasi *Progressive Web Apps* Pada Aplikasi Monitoring Service Laptop Dengan Teknologi Service Worker (Studi Kasus Service Laptop Bengkel OS)	2019	*Progressive Web Apps* dengan teknologi *service worker*	Dalam Tahap Proses

2.2 Dasar Teori

2.2.1 Service Laptop Bangkel OS

Service Laptop Bengkel OS yang beralamat di Jalan Suryowijan No.69c, Gedongkiwo, Mantrijeron, Kota Yogyakarta bergerak dalam bidang jasa perbaikan serta pelayanan *service* laptop. Bisnis ini dirintis sendiri oleh Bapak Muhammad Bilal Shah pada tahun 2008 memulai bisnis *service handphone*(HP) dan jual beli HP. Pada tahun 2013 bisnisnya berkembang dengan menambah *service* laptop sampai sekarang. Pada saat ini dapat menerima pengerjaan berupa perbaikan laptop, komputer dan HP.

Pengertian dari *service* itu sendiri adalah usaha untuk mengembalikan kondisi dan fungsi dari suatu benda atau alat yang rusak akibat pemakaian alat

tersebut pada kondisi semula . Proses perbaikan tidak menuntut penyamaan sesuai kondisi awal, yang diutamakan adalah alat tersebut bisa berfungsi normal kembali. Perbaikan memungkinkan untuk terjadinya pergantian bagian alat/*spare part*. Terkadang dari beberapa produk yang ada dipasaran tidak menyediakan *spare part* untuk penggantian saat dilakukan perbaikan, meskipun ada, harga *spare part* tersebut hampir mendekati harga baru satu unit produk tersebut. Hal ini yang memaksa user/pelanggan untuk membeli baru produk yang sama.

Tidak setiap perbaikan dapat diselesaikan dengan mudah, tergantung tingkat kesulitan dan kerumitan *assembling*/perakitan alat tersebut, mulai dari tingkatan jenis bahan hingga tingkat kecanggihan fungsi alat tersebut. Tingkat kesulitan tersebutlah yang menumbuhkan perbedaan jenis perbaikan, mulai jenis perbaikan ringan, perbaikan sedang dan perbaikan yang sering dinamakan *service* berat. Dari jenis *service* diatas ditentukan biaya perbaikan sesuai tingkat kesulitannya.

2.2.2 Teknologi *Progressive Web Apps* (PWA)

Saat ini teknologi aplikasi *web* sudah banyak mengalami perubahan fungsi. Awal mula perkembangan teknologi *web* dimulai dari web 1.0 yang diperkenalkan tahun 1990 yang masih bersifat statis hingga menjadi aplikasi web yang dapat menangani masalah pengecekan status baterai, penggunaan *mode offline*, hingga *speech recognition*. Salah satu teknologi yang tengah banyak diperhatikan saat ini adalah teknologi *Progressive Web App* (PWA).

Progressive Web Apps (PWA) adalah sebuah istilah untuk aplikasi berbasis *web* yang menggunakan teknologi web paling mutakhir. PWA sebenarnya

hanyalah aplikasi berbasis *web* biasa, tapi memanfaatkan fitur perambanan yang modern agar tampil seperti aplikasi asli. PWA digambarkan sebagai kumpulan dari teknologi, konsep desain dan *WEB* API (*Application Programming Interface*) yang bekerja secara bersama untuk memberikan sentuhan aplikasi pada sebuah *mobile web*. Hal ini termasuk berbagai rekomendasi yang tidak spesifik pada desain aplikasi web untuk perangkat mobile, seperti preferensi *HTTPS* melalui *HTTP* dan desain yang *responsive*. Hal ini juga akan membawa kebutuhan pada API baru untuk peningkatan kualitas pengguna, seperti *Web App Manifest, Service Workers* ataupun *Payment Request API.*

Untuk teknologi *Progressive Web Apps* ini pada dasarnya sama seperti aplikasi website lainnya, tetapi yang menjadi perbedaan antara teknologi PWA dengan teknologi aplikasi website yang lainnya yaitu pada PWA bekerja dengan konektifitas yang independen. Artinya aplikasi PWA dapat bekerja secara *offline* atau pada jaringan berkualitas buruk dengan adanya *service worker*. Tentu juga menggunakan teknologi instant loading yang membuat aplikasi website tersebut berjalan dengan cepat, *screenhome* dimana aplikasi website tersebut dapat dijadikan *icon* pada *desktop* atau *homescreen* dan pada PWA dapat menampilkan notifikasi pemberitahuan kepada pengguna tentang adanya pembaruan informasi pada aplikasi website tersebut.

PWA akan bekerja dengan me*load* file *HTML, CSS* dan *JavaScript* minumum yang diperlukan untuk membentuk antar muka pengguna PWA dan juga merupakan salah satu komponen yang memastikan website dapat berjalan sangat cepat dan langsung di simpan sementara ke perangkat lokal dalam *browser*

untuk nantinya jika setiap kali pengguna membuka aplikasi website, file antar muka akan dimuat dari penyimpanan sementara perangkat lokal yang membuat *loading* semakin cepat. Penyimpanan sementara secara lokal tersebut menggunakan *service worker* sehingga pada pemuatan berikutnya PWA hanya perlu mengambil data yang diperlukan, daripada memuat semuanya.

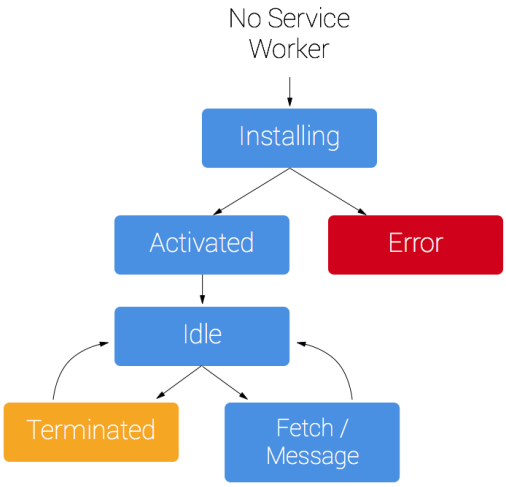
2.2.2 Service Worker

Salah satu Konsep yang dibangun oleh PWA adalah *Service Worker*. *Service worker* adalah *script* yang berjalan di belakang *browser* pengguna. *Service worker* tidak membutuhkan sebuah halaman ataupun interaksi dari pengguna untuk menjalankan tugasnya, dengan begitu *service worker* akan terus berjalan walaupun halaman web tidak terbuka.

Service Worker adalah *script* yang berjalan di latar belakang *browser* pengguna, yang tidak memerlukan halaman web atau interaksi dari pengguna. *Service worker* pada dasarnya adalah berkas *JavaScript* yang berjalan pada thread yang berbeda dengan main thread *browser*, menangani *network request*, *caching*, dan mengembalikan *resource* dari *cache*, dan bisa mengirim *push message*. *Service worker* bekerja sebagai pengatur event *fetch* dari browser, lalu *service worker* memutuskan apakah *request* akan ditersukan ke server atau ke *cache* berdasarkan kondisi jaringan *online* atau *offline*. (Gaunt, 2017).

Yang membuat *API* ini menarik adalah karena memungkinkan pengguna mendukung pengalaman *offline*. Sebelum *service worker,* ada satu API lain yang memberikan pengguna pengalaman *offline* di website, yaitu *AppCache*. Namun pada *AppCache* jumlah *gotcha* yang ada serta fakta bahwa meskipun desain

bekerja dengan sangat baik, untuk laman aplikasi web tunggal, namun ternyata tidak begitu baik untuk situs multi-laman. *Service Worker* telah didesain untuk menghindari titik-titik menyulitkan yang sudah umum tersebut.



Gambar 2.1 Siklus *Service Worker*

Langkah-langkah dalam melakukan konfigurasi dasar service worker sebagai berikut :

1. Daftarkan *service worker* melalui URL (*Uniform Resource Locator*) fungsi *serviceWorkerContainer.register*().
2. Jika berhasil, *service worker* dijalankan di *ServiceWorkerGlobalScope*.
3. *Service worker* telah siap untuk memproses *event.*
4. Instalasi *service worker* dicoba ketika *service worker* mengontrol halaman yang diakses setelah dan sebelumnya. *Event install* akan selalu dikirim pertama kali ke *service worker.*
5. Ketika *handler oninstall* selesai, *service worker* dipasang.

6. Proses aktivasi. Ketika *service worker* terpasang, selanjutnya akan menerima *event activate*. Penggunaan utama dari *onactivate* ini adalah untuk membersihkan sumber daya yang digunakan sebelumnya.
7. *Service control* sekarang dapat mengontrol halaman, tapi hanya dibuka setelah *register*() telah sukses seperti dokumen mulai aktif dengan atau tanpa *service worker* dan menjaganya selama masih digunakan. Jadi dokumen harus dimuat ulang agar benar-benar terkontrol.

2.2.3 HTTPS

Pada dasarnya HTTPS (*Hypertext Transfer Protocol Secure*) memiliki pengertian yang sama dengan HTTP, hanya saja pada HTTPS memiliki kelebihan dalam fungsi keamanannya. HTTPS bukan protokol yang terpisah, tetapi mengacu pada kombinasi dari interaksi HTTP normal melalui *Socket Layer* terenkripsi SSL (*Secure*) atau *Transport Layer Security* (LTS) mekanisme transportasi.

Hal ini menjamin perlindungan yang wajar dari penyadap dan (asalkan dilaksanakan dengan benar dan otoritas sertifikasi tingkat atas melakukan pekerjaan mereka dengan baik) serangan.

Secara teknis, website yang menggunakan HTTPS akan melakukan enkripsi terhadap informasi (data) mengunakan teknik enkripsi SSL. Dengan cara ini meskipun seseorang berhasil “mencuri” data tersebut selama dalam perjalanan *user* web *server,* orang tersebut tidak akan bisa membacanya karena sudah diubah oleh teknik enkripsi SSL. Umumnya website yang menggunakan HTTPS ini adalah website yang memiliki tingkat kerawanan tinggi yang berhubungan dengan

masalah keuangan dan privasi dari pelanggannya seperti *website* perbankan dan investasi.

HTTPS dienkripsi dan deskripsi dari halaman yang diminta oleh pengguna dan halaman yang di kembalikan oleh web *server*. Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks.*