BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam tinjauan pustaka ini diawali dengan menelaah penelitian terdahulu yang memiliki keterkaitan serta relevansi dengan penelitian yang akan dilakukan. Dengan demikian maka dapat diambil rujukan pendukung untuk membuat penelitian ini menjadi lebih memadai.

Pada penelitian ini *Afif Rizki Kurniawan (2018)* mengangkat masalah bagaimana agar proses lowongan pekerjaan pada akakom center dapat lebih mudah oleh para calon calon pekerja. Hasil dari penelitian ini adalah sebuah sistem informasi lowongan pekerjaan dapat diakses dimana saja. *Kurniawan, Awal (2017)* mengangkat masalah bagaimana sistem keluhan yang disisipkan *service workers* mampu melakukan proses *caching* data hingga 500 data keluhan. Meskipun eksekusi waktu yang dibutuhkan dalam mengakses aplikasi lebih lama karena pemasangan *service worker*, namun aplikasi yang diakses lebih cepat ketika dalam keadaan *offline*. *Rukoyah , dkk (2017)* Mengangkat masalah bagaimana RSUD Soreang memiliki banyak inventaris alat/barang diantaranya inventaris alat/barang perawatan kedokteran, keseluruhan barang, perangkat keras, buku-buku nota, formulir dan alat tulis kantor (ATK). Berbagai fasilitas inventaris alat/barang pada rumah sakit ini yang perlu dilindungi mencakup banyak karakteristik yang sangat perlu dipahami oleh praktisi keamanan sistem informasi

yaitu pada dasarnya fasilitas inventaris alat/barang rumah sakit yang ingin dijaga. Institusi pemerintahan yang membutuhkan perlindungan inventaris alat adalah rumah sakit sebagai sebuah institusi pelayanan kesehatan. Hasil dari sistem informasi inventaris akan memberikan kemudahan dalam mengelola barang dan mengelola pengajuan kebutuhan. *Susanto (2010)* mengembangkan sistem informasi *inventory* pada pt.dwiwarna inti sejahtera .dalam penelitiannyayang bergerak dalam bidang penyediaan barang agrikultur kesulitan dalam perekapan laporan yang sangat lama, sehingga Membuat kinerja perusahaan berkurang. Penelitian tersebut berpusat pada sistem *inventory* berbasis web yang dapat menyajikan informasi yang dibutuhkan konsumen. input *supplier*, input data sortir, *work order*, laporan barang masuk, laporan barang keluar, laporan pembeli dan laporan-laporan lainnya yang dapat diakses dalam tiga akses utama dalam bagian admin gudang,bagian direktur utama dan bagian pemasaran sehingga mudah mengelola data barang. Sistem ini hanya menyediakan pengaksesan data barang yang terjadi dalam gudang perusahaan tersebut, dan yang bisa mengakses adalah admin gudang, bagian direktur utama dan bagian pemasaran. Untuk metode sistem ini adalah *Rapid Application Development (RAD)* dan *Unified Modelling Language (UML)* sebagai alat pemodelannya. *Pribadi, Tegar (2019)* mengangkat masalah bagaimana pada dinas sosial yogyakarta bagian perlindungan sosial dan penanganan bencana alam mengolah data dengan cara menyalin pengolahan persediaan.Adapun perbandingan tinjauan pustaka tersebut dapat di lihat pada

Tabel 2.1 Tinjauan Pustaka

Penulis	Obyek	Bahasa Pemrograman	Hasil
Afif Rizki Kurniawan (2018)	Penerapan Progressive Web Apps Pada Aplikasi Lowongan Pekerjaan Dengan Teknologi Service Worker (Studi Kasus Akakom Carrer Center)	Java Script, HTML5	Sistem ini dibuat menggunakan teknologi service worker dengan tujuan untuk memudahkan pencari lowongan pekerjaan di akakom carrer center
Kurniawan Awal (2017)	Implementasi Progressive Web Application Pada Sistem Monitoring Keluhan Sampah Kota Makassar	Java Script, HTML5	Sistem ini dibuat menggunakan progressive webs apps dengan monitoring keluhan Sampah.
Rukoyah , dkk (2017)	Sistem Informasi Inventaris Barang Pada RSUD Soreang.	PHP,Tanpa menggunakan Progresive Web Apss	Sistem ini dibuat menggunakan teknologi php tapi belum menggunakan progressive webs apps.
Susanto (2010)	Informasi*Inventory* Pada Pt.Dwiwarna Inti Sejahtera	PHP,tanpa Progressive web apps	Sistem ini dibuat menggunakan php dan belum menggunakan progressive webs apps

Ahmad Thaariq (2019)	Implementasi Progessive Web Apps Pada Inventory Obat di apotik hikmah	Java Script, HTML5, CSS, PHP	Sistem ini menggunakan teknologi service worker dengan menerapkan progressive webs apps pada inventory obat di apotik hikmah

Dari beberapa tinjauan pustaka, dapat disimpulkan bahwa Penerapan *Progressive Web Apps* pada implementasi inventory obat pada apotik hikmah dengan teknologi *service worker*, model informasi kemudian berdasarkan hasil penelitian itu akan dirumuskan satu analisis dan desain sistem untuk mengembangkan informasi inventory obat yang sudah ada tersebut ke dalam model aplikasi inventory barang dengan penerapan *PWA*. Adapun beberapa teori yang menyangkut penerapan *PWA*.

2.1 Dasar Teori

2.2.1 Progressive Web Apps (PWA)

Teknologi *PWA* mengambil kelebihan dari teknologi baru yang mengambil bagian dari aplikasi mobile dan aplikasi native. Teknologi ini oleh para developer google pada saat acara *I/O google* di *mountain view California*. Menurut Rica Handayani selaku *StayegicPartner Manager Google* Asia Tenggara, menyatakan bahwa *PWA* akan membuat sebuah situs seperti aplikasi yang akan memberikan performa lebih baik.(Neha Sharma,*2015*).

Progressive Web Apps (PWA) adalah aplikasi *native* yang mendukung *hybrid* secara penuh dan aplikasi ini tidak perlu proses penginstallan terlebih dahulu namun langsung dapat digunakan secara penuh. Program *PWA* memiliki banyak kelebihan yang akan memudahkan

penggunan dalam menyelami sebuah website secara penuh. Apabila dibandingkan dengan *hybrid,PWA* ini penengah antara *native* dan *hybrid* sehingga kondisinya akan lebih stabil namun tetap up to date sesuai kondisi *hybrid* yang sebenarnya. *Icon* dapat dipasang pada bagian *desktop* atau s*creenhome* pada *mobile* agar pengguna dapat melihat notifikasi dengna lebih mudah. Hanya saja untuk saat ini browser yang support dengan *PWA* ini hanya *chrome* diatas 47.

Pada dasarnya teknologi *Progressive Web Apps* bekerja layaknya aplikasi website lainnya,namun yang menjadi perbedaan dengan teknologi aplikasi website lainnya pada *PWA* bekerja dengan konektivitas yang independen. Artinya, aplikasi *PWA* dapat bekerja secara *offline* atau pada jaringan berkualitas rendah dengan adanya *service worker*. Tentu saja selain mampu berjalan di koneksifitas rendah atau *offline, PWA* 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 notifikasi yang artinya pada *PWA* dapat menampilkan pemberitahuan kepada pengguna tentang adanya pembaruan informasi pada aplikasi website tersebut.

PWA akan bekerja dengan meload file *HTML,CSS*dan *Javascript* minimum yang diperlukan untuk membentuk antarmuka 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 antarmuka akan

dimuat dari penyimpanan sementara perangkat lokal yang membuat waktu loading semakin cepat. Penyimpanan sementara secara lokal tersebut menggunakan *service worker* sehingga pada pemuatan berikutnya *PWA* hanya perlu mengambil data yang di butuhkan, daripada memuat semuanya.

2.2.2 Service Worker

Progressive Web Apps harus cepat ,yang berarti bahwa tetap berfungsi saat *online,offline*, dan pada koneksi yang lamban serta tidak stabil. Untuk mencapainya,*PWA* harus menyimpan sementara tampilan antarmuka dari aplikasi website dengan menggunakan *service worker*, sehingga selalu tersedia dengan cepat. *Service worker* adalah skrip yang dijalankan browser yang berjalan di latar belakang yang terpisah dari laman web, yang membuka pintu ke berbagai fitur dimana tidak memerlukan laman web ataupun interaksi pengguna. Saat ini*service worker* sudah menyertakan berbagai fitur seperti pemberitahuan, push, dan sinkronisasi latar belakang. *Service worker* juga berkemampuan mencegah dan menangani permintaan jaringan termasuk mengelola *cache* respons melalui program.

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.

Untuk memasang *service worker*, perlu mendaftakannya sebagai file *JavaScript*. Mendaftarkan *service worker*akan menyebabkan browser memulai langkah pemasangan *service worker* di latar belakang. Biasanya selama langkah pemasangan, perlu menyimpan sementara beberapa asset statis. Jika semua file berhasil di simpan sementara, maka *service worker*akan terpasang. Jika ada file yang gagal di unduh dan disimpan sementara, maka langkah pemasangan akan gagal dan *service worker* tidak akan diaktifkan sehingga membuat aplikasi website tersebut tidak dapat berjalan secara *offline*. Setelah *service worker* aktif, maka secara bersamaan akan menangani manajemen penyimpanan sementara yang sebelumnya. *service worker* akan mengontrol semua laman yang berada dalam cakupannya, walaupun laman yang mendaftarkan *service worker* untuk pertama kali tidak akan di control hingga dimuat ulang.jika *service worker* sudah terkontrol,*service worker* akan berada dalam salah satu dari dua keadaan,yaitu : *service worker* akan dihentikan untuk menghemat memori, atau s*ervice worker* akan menangani pesan untuk menjalankan permintaan saat jaringan buruk. (Anonim, 16 april 2020)

2.2.3 HTTPS

Pada dasarnya *HTTPS (Hypertext Transfer Protocol Secure)* memiliki pengertian yang sama dengan *HTTP*, hanya saja pada *HTTPS* memiliki kelebihan dalam gungsi keamanannya. *Https* bukan protokol yang terpisah, tetapi mengacu pada kombinasi dari interaksi *HTTP* normal melalui Socket Layer terenkripsi *SSL (Secure)* atau *Transport Layer Security (TLS)* 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. (Niagahoster, 5 februari 2018)

2.2.4 PHP

PHP merupakan bahasa *scripting server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, server-lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan di kirim kepada *client* yang melakukan permintaan. (Didik Dwi Prasetyo, 2004).

2.2.5 Java Script

Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan. (Sun *Microsystem*, dalam buku M. Shalahudin dan Rosa A.S, 2010).

Java berdiri diatas sebuah mesin penterjemah (*interpreter*) yang diberi nama*Java Virtual Machine (JVM). JVM* inilah yang akan membaca kode bit *(bytecode)* dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa *Java* disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, jika pada sistem operasi tersebut terdapat *JVM*.Alasan utama pembentukan bahasa *Java* adalah membuat aplikasi-aplikasi yang dapat

diletakkan di berbagai macam perangkat elektronik, sehingga *Java* harus bersifat tidak bergantung pada *platform* (*platformindependent*). Itulah yang menyebabkan dalam dunia pemograman *Java* dikenal adanya istilah "*write once, run everywhere*", yang berarti kode program hanya ditulis sekali, namun dapat dijalankan dibawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

2.2.6 MySQL

MySQL merupakan *RDBMS* (*server database*) yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak *user*.(Raharjo, 2011).

MySQL adalah sebuah *software open source* yang digunakan untuk membuat sebuah database (Kadir,2008).