

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Pradana (2017), membuat sistem informasi penyewaan barang pada Kharisma *Sound System* dan Dekorasi yang berbasis Website dengan menggunakan Bahasa pemrograman PHP. Sedangkan Penelitian yang dilakukan oleh Prasetyo (2017), membuat rancangan untuk membangun sebuah website tentang Sistem Informasi rental studio musik yaitu pada studio musik Impact Yogyakarta dimana sama-sama menggunakan Bahasa pemrograman PHP.

Penelitian yang dilakukan oleh Pambudi dan Indriyanti (2015). Membuat aplikasi reservasi studio musik yang berbasis website dengan memanfaatkan teknologi *Framework Laravel*. Sedangkan yang dilakukan oleh Pidekso (2017). Membuat sebuah aplikasi rental studio band yang berbasis website, berbeda dengan yang dilakukan sebelumnya oleh Pambudi dan Indriyanti (2015) yang memanfaatkan teknologi *Framework Laravel* sedangkan penelitian pidekso menggunakan Bahasa pemrograman PHP.

Penelitian yang dilakukan oleh Kusuma (2017). Penelitian ini bertujuan untuk membuat media informasi tentang studio musik YHK *record* yang berada di Kabupaten Lahat, tidak jauh berbeda dengan yang dilakukan Pambudi dan Indriyanti (2015) yang menggunakan teknologi *Framework Laravel*. Penelitian Kusuma (2017) menggunakan teknologi rivalnya yaitu dengan mengimplementasikan *Framework Codeigniter*.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

No.	Penulis	Judul	Metode	Hasil
1.	Pradana (2017)	Sistem Informasi Penyewaan Barang Berbasis <i>Web Mobile</i> Pada Kharisma <i>Sound System</i> dan Dekorasi	<i>PHP</i>	Untuk digunakan sebagai sistem informasi penyewaan dengan berbasis <i>Web Mobile</i>
2.	Prasetyo (2014)	Perancangan Sistem Informasi Rental Studio Musik Pada Studio Musik Impact Yogyakarta Berbasis Web	<i>PHP</i>	Membangun atau merancang sebuah sistem informasi yang digunakan untuk rental studio musik pada studio musik Impact Yogyakarta yang berbasis Web
3.	Pambudi & Indrayanti (2015)	Reservasi Studio Musik Berbasis <i>Web</i>	Implementasi <i>Framework Laravel</i>	Untuk membuat website aplikasi reservasi studio musik berbasis website dengan implementasi <i>Framework Laravel</i>
4.	Pidekso (2017)	Rental Studio Band Berbasis <i>Website</i>	<i>PHP</i>	Untuk membuat website aplikasi rental studio band berbasis <i>Website</i>
5.	Kusuma (2017)	Implementasi Media Informasi Studio Musik YHK Record Studio Di Kabupaten Lahat Menggunakan <i>Framework Codeigniter</i>	Implementasi <i>Framework Codeigniter</i>	Untuk membuat media informasi studio musik pada YHK <i>Record Studio</i> yang berada di Kabupaten Lahat dengan menggunakan implementasi <i>Framework Codeigniter</i>
6.	Prayoga (2020)	Aplikasi Reservasi studio Musik Berbasis <i>Progressive Web Apps</i> Studi Kasus Fusion Musik Yogyakarta	<i>Progressive Web Apps</i>	Membuat aplikasi Reservasi Studio Musik berbasis <i>Progressive Web Apps</i> pada fusion music Yogyakarta

2.2 Dasar Teori

2.2.1 Studio Musik Fusion Yogyakarta

Studio musik Fusion yang beralamat di Jalan Selokam Mataram, Condongcatur, Pringgolayan, Dabag, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta yang bergerak dibidang jasa penyewaan alat musik.

Fusion music memiliki dua studio yaitu satu dilantai bawah dan satu dilantai dua. Tarif penyewaan studio musik yaitu 1 jam Rp.40.000, 2 jam Rp.50.000.

Studio musik Fusion memiliki alat musik yang lengkap, dimana setiap studio (atas,bawah) memiliki perlengkapan musik yang berkelas. Seperti Gitar Gibson, Drum Yamaha dll. Fasilitas pada masing-masing studio terdapat dua gitar, satu bass, satu keyboard(piano), satu set drum dan dua microphone.

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 konektivitas 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 *reload* file *HTML*, *CSS* dan *JavaScript* minimum 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.

Berikut ini adalah beberapa ciri-ciri website sudah terpasang PWA :

1. Instant loading

Hal ini lah yang dinamakan *service worker* dimana kita dapat mengakses aplikasi / website meskipun dalam keadaan *offline* sekalipun.

2. Connectivity Independent

Masih berhubungan dengan poin sebelumnya, fitur ini dapat mempermudah kita mengakses di kondisi koneksi apapun.

3. App-like Mode.

Dengan menggunakan PWA ini, seolah kita membuat aplikasi native pada smartphone yang tidak harus diinstal pada smartphone kita. Ini memungkinkan karena PWA sendiri dibuat dengan model App Shell dalam full screen mode.

4. Fresh dan Safe

PWA lebih smooth dan dapat diakses dengan cepat selain itu juga dapat diakses dengan aman melalui perangkat apapun.

5. Discoverable

PWA bisa teridentifikasi sebagai “aplikasi” oleh mobile device, dengan menggunakan W3C Web App Manifest dan Service Worker registration scope yang memungkinkan mesin pencari bisa menemukannya. Padahal kita tidak menginstal aplikasi tapi diakses melalui PWA.

6. Re-engageable, Installable, dan Linkable

Ketiga poin ini menggambarkan PWA mempunyai fitur yang sangat beragam sehingga akan lebih praktis digunakan. seperti yang kita ketahui, saat kita mengakses suatu website yang terintegrasi dengan PWA, akan memunculkan notifikasi apakah kita ingin menambahkan ke Homescreen, jika iya, maka aplikasi akan muncul di homescreen tanpa harus melewati proses instalasi, dan untuk

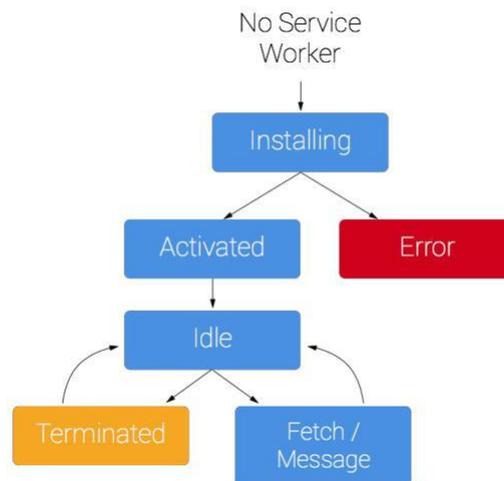
mengaksesnya tinggal melakukan klik icon aplikasi pada homescreen.

2.2.3 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 diteruskan 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.



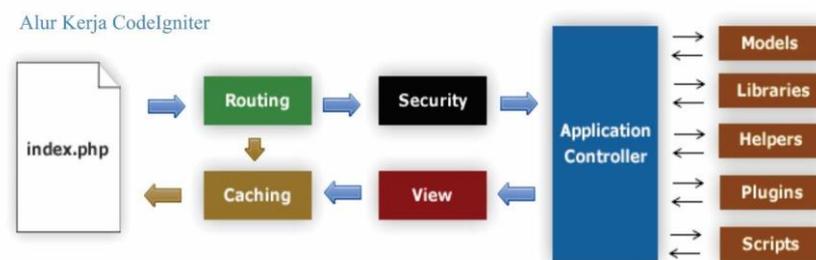
Gambar 2.1 Siklus *Service Worker*

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.

2.2.4 Framework *CodeIgniter*

Framework CodeIgniter merupakan sebuah aplikasi yang bersifat opensource yang digunakan untuk membuat sebuah aplikasi dengan bahasa pemrograman PHP. Codeigniter menggunakan konsep MVC (Model, View, Controller) dalam membangun aplikasi website dinamis. Hal tersebut akan mempermudah kinerja para programmer dalam membuat aplikasi website dinamis. Selain ringan dan cepat, codeigniter juga memiliki dokumentasi manual yang sangat lengkap.



Gambar 2.2 Alur Kerja *CodeIgniter*

Dibawah ini merupakan keterangan dari gambar 2.2 :

1. Index.php: Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
2. The Router: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
3. Cache File: Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
4. Security: Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.

5. Controller: Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
6. View: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

2.2.5 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) menggunakan 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*.

2.2.6 MySQL

MySQL adalah sistem manajemen database relasional *open source* (RDBMS) dengan *client-server* model. Sedangkan RDBMS merupakan software untuk membuat dan mengelola database berdasarkan pada model relasional.

Sebelum dibahas lebih lanjut, ada baiknya bagi kita untuk mengetahui sejarah singkat MySQL. MySQL dibaca MY-ES-KYOO-EL [maɪ,ɛs,kjuːˈɛl]. Beberapa orang bahkan membaca MySQL seperti sedang menyebutkan “my sequel”. MySQL AB, sebuah perusahaan asal Swedia, menjadi yang pertama dalam mengembangkan MySQL di tahun 1994. Hak kepemilikan MySQL kemudian diambil secara menyeluruh oleh perusahaan teknologi Amerika Serikat, Sun *Microsystems*, ketika mereka membeli MySQL AB pada tahun 2008. Di tahun 2010, Oracle yang adalah salah satu perusahaan teknologi terbesar di Amerika Serikat mengakuisisi *Sun Microsystems*. Semenjak itulah, MySQL sepenuhnya dimiliki oleh Oracle.

Di bawah ini kami akan membahas satu per satu mengenai sejumlah software yang terkait dalam RDBMS:

1. Database

Dalam bahasa yang sederhana, database adalah sekumpulan data yang terstruktur. Anggap saja Anda sedang berpose *selfie*: yang dilakukan pastilah menekan tombol kamera depan, lalu mengabadikan potret diri. Foto *selfie* yang diambil adalah data, sedangkan galeri foto merupakan database. Database adalah tempat untuk menyimpan dan mengelola data. Kata “relasional” berarti data yang disimpan di dataset dikelola sebagai tabel.

Setiap tabel saling terkait. Jika software tidak mendukung model data relasional, maka yang dipanggil hanyalah DBMS.

2. *Open Source*

Jika suatu *software* atau *tool* dikatakan *open source*, maka itu berarti Anda bebas menginstall, menggunakan, bahkan memodifikasinya. Anda pun dapat mempelajari dan mengkustomisasikan *source code*-nya agar *software* bisa diatur dan diubah sesuai keinginan dan kebutuhan Anda. Hanya saja, aktivitas yang Anda lakukan tak hanya tergantung pada situasi dan kondisi, tapi juga ditentukan oleh GPL (*GNU Public License*). Lisensi berbayar tersedia bagi Anda yang menginginkan fleksibilitas terhadap kepemilikan dan bantuan tingkat lanjut (*premium*).

3. *Client-server Model*

Komputer yang memasang dan menjalankan *software* RDBMS disebut sebagai *client*. Agar bisa mengakses data, komputer harus terhubung dengan server RDBMS terlebih dulu. Keadaan seperti inilah yang disebut *client-server*.

4. MySQL

MySQL adalah salah satu pilihan *software* RDBMS. Terkadang RDBMS dan MySQL dianggap sama karena popularitas MySQL. Aplikasi web terkenal, seperti Facebook, Twitter, YouTube, Google, dan Yahoo! menggunakan MySQL untuk menyimpan data. Pada awalnya MySQL dibuat untuk penggunaan terbatas saja, tapi sekarang *software* ini sudah kompatibel dengan berbagai platform computing, seperti Linux, macOS, Microsoft Windows, dan Ubuntu.