BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Afif Rizki Kurniawan (2018) pada penelitianya menekankan pada pembuatan aplikasi lowongan pekerjaan yang terdapat di Akakom Carrer Center yang ditujukan untuk memberikan informasi mencari lowongan pekerjaan yang terdapat di Akakom Carrer Center yang diperuntuk bagi pelamar alumni STMIK Akakom Yogyakarta. Aplikasi yang dibuat menerapkan metode *Progressive Web Apps* (PWA) dan teknologi *service worker*. Hasil dari penelitan tersebut adalah pengguna dapat menemukan informasi tentang lowongan pekerjaan yang terdapat di Akakom Carrer Center.

Ahmad Zulyadaen (2018) pada penelitianya menekankan pada pembuatan aplikasi reservasi obyek wisata di daerah pulau Lombok yang ditujukan untuk memberikan informasi kepada wisatawan mengenai obyek-obyek wisata di pulau Lombok. Aplikasi yang dibuat tersebut berbasis website dengan hasil penelitiannya yaitu pengguna dapat menemukan informasi paket *tour*, tujuan wisata, harga tiket wisata, dan pesan tiket secara *online* yang terdapat di The Lombok Travel.

Candra Arief Syuada (2018) pada penelitianya menekankan pada pembuatan aplikasi reservasi kost di Yogyakarta yang ditujukan untuk memberikan informasi mengenai tempat tinggal kost yang ada di daerah Yogyakarta. Aplikasi yang dibuat tersebut berbasis *website* dengan menggunakan teknologi Framework Codeigniter. Hasil dari penelitan tersebut adalah pengguna dapat menemukan

informasi tentang harga kost, alamat kost, dan pemesanan tempat tinggal kost secara *online*.

Tabel 2.1. Perbandingan tinjauan pustaka

No.	Penulis	Judul	Metode	Hasil
1.	Afif Rizki Kurniawan (2018)	Penerapan *Progressive Web Apps* Pada Aplikasi Lowongan Pekerjaan dengan Teknologi *Service Worker* (Studi Kasus Akakom Carrer Center)	*Progressive Web Apps* dengan teknologi *service worker*	Menyediakan lowongan pekerjaan yang terdapat di Akakom Carrer Center dengan menggunakan metode *Progressive Web Apps*
2.	Ahmad Zulyadaen (2018)	Reservasi Objek Wisata Di Daerah Pulau Lombok Berbasis Web (Studi Kasus The Lombok Travel)	*PHP* dan *MySQL*.	Menyediakan informasi paket *tour*, tujuan wisata, harga tiket, pesan tiket secara *online* dengan menggunakan bahasa pemrograman *PHP* dan *MySQL*.
3.	Candra Arief Syuada (2018)	Aplikasi Reservasi Kost Yogyakarta Berbasis Web Menggunakan Teknologi Framework Codeigniter	Framework Codeigniter	Menyediakan informasi reservasi kost di Yogyakarta dengan menggunakan teknologi framework codeigniter
4.	Ahmad Fauzi Karim (2019)	Implementasi *Progressive Web Apps* Dengan Teknologi *Service Worker* (Studi Kasus Reservasi Homestay Tembi)	*Progressive Web Apps* dengan teknologi *service worker*	Menyediakan reservasi homestay dengan menggunakan metode *Progressive Web Apps*

2.2 Dasar Teori

Penulis mengambil referensi dari berbagai media dan buku untuk mendapatkan dasar teori dan pengetahuan yang lebih kuat dalam penerapan *Progressive Web Apps,* seperti :

2.2.1 Teknologi *Progressive Web Apps* (PWA)

Progressive Web Apps (PWA) adalah pengalaman yang menggabungkan pengalaman yang terbaik dari web dan pengalaman yang terbaik dari aplikasi. Pengalaman ini bermanfaat untuk pengguna dari kunjungan pertamanya di tab browser, tanpa harus melakukan pemasangan seperti di *playstore* maupun di *app store*. Aplikasi dimuat dengan cepat bahkan pada jaringan yang tidak stabil, mengirimkan pemberitahuan push yang relevan, memiliki ikon pada layar beranda, dan dimuat dengn pengalaman tingkat atas selayar penuh. (Lepage, 2017)

Untuk teknologi PWA 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.

2.2.2 Service Worker

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)

2.2.3 HTTPS

Hypertext Transfer Protocol Secure (HTTPS) memiliki pengertian yang sama dengan http hanya saja https memiliki kelebihan fungsi di bidang keamanan (secure). Dengan menggunakan Secure Socket Layer (SSL) atau Transport Layer Secure (TLS) sebagai sublayer di bawah http aplikasi layer yang biasa. Teknologi https protokol mencegah kemungkinan "dicurinya" informasi penting yang dikirimkan selama proses komunikasi berlangsung antara user dengan web server atau sebaliknya.

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.

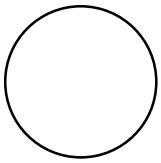
HTTPS dienkripsi dan deskripsi dari halaman yang diminta oleh pengguna dan halaman yang di kembalikan oleh web *server*. Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks.* HTTPS menyediakan tiga jaminin keamanan seperti berikut ini (Palmer, 2017):

1. Otentikasi server memungkinkan pengguna untuk memiliki keyakinan bahwa mereka berbicara dengan server aplikasi yang benar. Tanpa jaminan ini, tidak ada jaminan kerahasiaan atau integritas.
2. Kerahasiaan data berarti penyadap tidak dapat memahami isi komunikasi antara browser pengguna dan server web, karena data dienkripsi.
3. Integritas data berarti penyerang jaringan tidak dapat merusak atau mengubah konten komunikasi antara browser pengguna dan server web, karena mereka divalidasi dengan kode otentikasi pesan kriptografi.

2.2.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat untuk membuat diagram yang serbaguna. Data flow diagram terdiri dari notasi penyimpanan (data *store*), proses (*process*), aliran data (*flow* data), dan sumber masukan (*entity*). Seperti yang ditunjukan pada Tabel 2.2.

Tabel 2.2. Simbol-Simbol Diagram Alir Data

Simbol	Arti	Deskripsi
	External Entity	Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.
	Proses	Aktifitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasanya berupa manual maupun terkomputerisasi.
	Data Flow	Suatu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.
	Data Store	Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam satu store. Aliran data di-update atau ditambahkan ke data store.

2.2.5 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah suatu diagram untuk menggambarkan desain konseptual dari model suatu basis data relasional. ERD juga merupakan gambaran yang menghubungkan antara objek satu dengan objek

yang lain dalam dunia nyata. Bisa dikatakan bahwa bahan yang akan digunakan untuk membuat ERD adalah dari objek dunia nyata. (Utami dan Hartanto, 2012).

a. Komponen *Entity Relationship Diagram* (*ERD*)

Adapun bentuk symbol grafis atau komponen dari *Entity Relationship Diagram* (*ERD*) sebagai berikut (Priyadi, 2014):

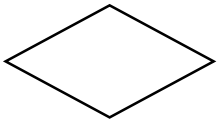
1. Entitas

Entitas merupakan notasi untuk mewakili suatu objek dengan karateristik sama, yang dilengkapi oleh atribut, berfungsi memberikan identitas sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan lainya. Entitas di lambangkan dengan bentuk persegi panjang sepeti Gambar 2.1.

Gambar 2.1 Simbol Entitas

2. Relasi

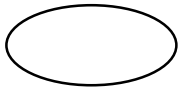
Relasi merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Dan fungsinya untuk mengetahui jenis hubungan antara 2 *file*, relasi berbentuk belah ketupat seperti Gambar 2.2.



Gambar 2.2 Simbol Relasi

3. Atribut

Berbentuk lingkaran atau elip yang menyatakan atribut (atribut yang berfungsi sebagai key digaris bawahi). Atribut digambarkan pada Gambar 2.3.



Gambar 2.3 Simbol Atribut

4. Garis Penghubung

Garis penghubung merupakan notasi yang menghubungkan antara atribut dengan entitas dan entitas dengan relasi yang digunakan dalam Diagram E-R. Garis penghubung berbentuk garis seperti tertera pada gambar 2.4.

Gambar 2.4 Simbol Garis Penghubung

b. Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat beberapa jenis kardinalitas yang menggambarkan relasi antar entitas, adapun beberapa kardinalitas tersebut sebagai berikut (Fathansyah, 2012):

1. Relasi satu-ke-satu *(one-to-one)*

Entitas hanya boleh berhubungan dengan satu entitas kedua dan sebaliknya yang dijelaskan pada Gambar 2.5.

Gambar 2.5 *Relasi one to one*

2. Relasi satu-ke-banyak *(one-to-many)*

   Entitas pertama boleh banyak berhubungan dengan entitas kedua, tetapi entitas kedua hanya boleh berhubungan dengan satu entitas atau sebaliknya yang dijelaskan pada Gambar 2.6.

Gambar 2.6 *Relasi one to many*

3. Relasi banyak-ke-*satu (many-to-one)*

   Entitas pertama hanya berhubungan dengan satu entitas kedua, tetapi entitas kedua boleh berhubungan dengan banyak entitas atau sebaliknya yang dijelaskan pada Gambar 2.7.

Gambar 2.7 *Relasi many to one*

4. Relasi banyak-ke-banyak *(many-to-many)*

   Entitas pertama boleh banyak berhubungan dengan entitas kedua atau sebaliknya yang dijelaskan pada Gambar 2.8.

Gambar 2.8 *Relasi many to many*