

BAB II TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang aplikasi pelaporan kecelakaan lalu lintas menggunakan metode *Location Based Service* berbasis *Progressive Web App* ini menggunakan referensi dari beberapa penelitian yang terkait dengan topik bahasan dalam penelitian ini. Penelitian yang digunakan sebagai bahan studi literatur dalam penelitian ini, antara lain :

1. Penelitian yang dilakukan oleh A. Kurniawan, I. S. Areni, dan A. Achmad (2018) dengan judul “Implementasi *Progressive Web Application* pada Sistem *Monitoring* Keluhan Sampah Kota Makassar”. Penelitian tersebut dilakukan dengan tujuan untuk merancang sebuah sistem yang dapat melakukan proses *caching file* pada konten *website* dengan menggunakan *Progressive Web Apps* dan memanfaatkan *Service Worker*. Dalam penelitian tersebut, data yang bersumber dari sebuah *API (Application Programming Interface)* kemudian ditampilkan dalam keadaan aplikasi *online* atau terhubung ke jaringan *internet*. Hasil dari penelitian tersebut yaitu aplikasi yang disisipkan *Service Worker* mampu melakukan proses *caching* data hingga 500 data keluhan. Meskipun eksekusi waktu yang dibutuhkan dalam mengakses aplikasi lebih lama karena pemasangan *Service Worker*, namun dalam penelitian tersebut diperoleh hasil bahwa aplikasi yang diakses dapat lebih cepat dibandingkan ketika dalam keadaan *offline* atau tidak terhubung kedalam jaringan *internet*. Keterkaitan antara penelitian tersebut dengan topik penelitian yang akan dilakukan dalam penelitian ini yaitu dalam hal implementasi teknologi *Progressive Web Apps* dengan memanfaatkan *Service Worker* untuk melakukan proses *caching file* sehingga aplikasi dapat diakses secara lebih cepat walaupun dalam kondisi jaringan *internet* yang kurang baik maupun dalam keadaan *offline*.
2. Penelitian yang dilakukan oleh M. Z. F. Hanindra, N. Safriadi, H. Anra (2017) dengan judul “Rancang Bangun Aplikasi *SOS Broadcast* Lokasi dan Status Keamanan *User* Sebagai Sarana Cepat Tanggap Tindak Pidana Kejahatan Dini Menggunakan *Location Based Service* Berbasis *Android*”. Penelitian tersebut

bertujuan untuk membuat sebuah aplikasi *android* menggunakan *Location Based Service* sebagai sarana penyebar informasi lokasi pengguna yang sedang dalam keadaan darurat atau dalam ancaman kepada daftar kontak darurat seperti keluarga atau kerabat secara *broadcast*. Keterkaitan antara penelitian tersebut dengan penelitian ini yaitu dalam hal penggunaan *Location Based Service* sebagai sarana memperoleh informasi lokasi antara pengguna dan kontak kerabat atau keluarga yang digunakan dalam penelitian tersebut.

3. Penelitian yang dilakukan oleh Mardani (2014) dengan judul penelitian “Sistem Informasi Geografis Pelaporan Masyarakat (SIGMA) Berbasis Foto *Geotag*”. Penelitian tersebut bertujuan untuk memecahkan masalah keterlambatan arus informasi kepada pihak pemerintah tentang masalah yang terjadi di lingkungan masyarakat yang seharusnya menjadi tanggungjawab dari pihak pemerintah. Dalam penelitian tersebut ditemukan solusi atas permasalahan dengan membuat aplikasi pelaporan masyarakat terhadap masalah-masalah yang terjadi di lingkungan masyarakat kepada pihak pemerintah ataupun lembaga yang terkait dalam menangani masalah tersebut. Contoh masalah yang dapat dilaporkan antara lain kerusakan infrastruktur, permasalahan sosial dan pencemaran lingkungan. Bentuk laporan yang dibahas dalam penelitian tersebut berupa data lokasi yang bersumber dari foto *geotag* yang dikirimkan pelapor atau masyarakat. Keterkaitan antara penelitian tersebut dengan penelitian ini yaitu dalam hal pelaporan yang berbasis lokasi dimana laporan disampaikan dengan memanfaatkan data koordinat lokasi yang dikirimkan oleh pelapor.

2.2 Aplikasi

Yuhefizar (2012) mengungkapkan bahwa aplikasi merupakan suatu program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Selanjutnya

Berdasarkan kedua pendapat ahli di atas, maka dapat disimpulkan bahwa aplikasi merupakan sebuah program perangkat lunak yang dibuat dengan tujuan memenuhi kebutuhan dari aktivitas yang dilakukan oleh pengguna serta mempermudah pekerjaan dari pengguna tersebut.

2.3 Laporan

Menurut Keraf (2006), laporan adalah suatu cara komunikasi di mana penulis menyampaikan informasi kepada seseorang atau suatu badan karena tanggung jawab yang dibebankan kepadanya. Selanjutnya Rama dan Jones (2008), mengungkapkan bahwa laporan merupakan sebuah persentasi data yang telah terformat dan tersusun menjadi sebuah informasi yang jelas yang akan menjadi informasi akan sebuah kegiatan dan juga peristiwa yang terjadi.

Berdasarkan kedua pendapat ahli di atas, dapat disimpulkan bahwa laporan merupakan suatu bentuk penyampaian informasi berdasarkan suatu kegiatan atau peristiwa yang terjadi dan didukung dengan data yang lengkap serta sesuai dengan fakta sehingga informasi yang diberikan dapat dipercaya dan mudah dipahami serta dalam penyampaiannya laporan dapat bersifat lisan maupun tertulis.

2.4 Kecelakaan lalu lintas

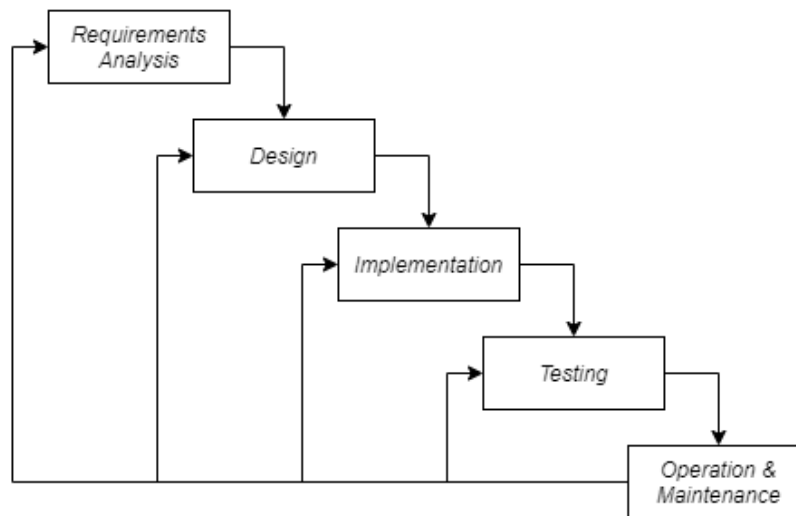
D.A. Colling (1990) yang dikutip oleh (2009) mengungkapkan bahwa kecelakaan merupakan sebuah kejadian yang tidak direncanakan dan terkontrol yang dapat disebabkan oleh manusia, situasi, faktor lingkungan, ataupun kombinasi dari hal-hal tersebut yang mengganggu proses kerja dan dapat menimbulkan cedera ataupun tidak, kesakitan, kematian, kerusakan properti ataupun kejadian yang tidak diinginkan lainnya. Selanjutnya berdasarkan *Undang-Undang Nomor 22 Tahun 2009* tentang Lalu Lintas dan Angkutan Jalan, mengungkapkan kecelakaan lalu lintas adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja yang melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan/atau kerugian harta benda.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa kecelakaan lalu lintas merupakan peristiwa yang terjadi pada lalu lintas jalan yang tidak terduga dan tidak diinginkan yang juga sulit diprediksi kapan dan dimana terjadinya. Kecelakaan lalu lintas juga sedikitnya melibatkan satu kendaraan dengan atau tanpa pengguna jalan lain yang menyebabkan cedera, trauma, kecacatan, kematian dan/atau kerugian harta benda pada pemiliknya.

2.5 Model Waterfall

Terdapat beberapa metodologi *Systems Development Life Cycle (SDLC)*

yang biasa digunakan dalam membangun sebuah sistem, salah satunya adalah model *waterfall*. *Waterfall* merupakan model yang bersifat sistematis dan termasuk dalam model klasik, nama lainnya adalah *Linear Sequential Model* (Pressman, 2002). Tahapan-tahapan model *waterfall* dapat dilihat seperti pada gambar 2.1.



Gambar 2. 1 Model *Waterfall* (Pressman, 2002)

Tahapan-tahapan dari metode *waterfall* tersebut, dapat diuraikan sebagai berikut :

1. *Analysis* (Analisis)

Fase ini merupakan proses analisa terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem, sehingga kebutuhan yang diperlukan untuk sistem baru akan didapatkan.

2. *Design* (Perancangan)

Perancangan merupakan proses penentuan cara kerja sistem dalam hal perancangan antarmuka, *database*, dan perancangan alur program. Perancangan diperlukan untuk menggambarkan sistem baru dengan tujuan memenuhi kebutuhan pengguna.

3. *Implementation* (Implementasi)

Tahapan implementasi yaitu tahap rancangan sistem yang dibentuk menjadi suatu kode program untuk pembuatan sistem.

4. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari error yang

terjadi. Testing juga dilakukan untuk memastikan kevalidan dalam proses input sehingga dapat menghasilkan output yang sesuai.

5. *Maintenance* (Pemeliharaan)

Fase ini yaitu pemeliharaan dan pengembangan sistem yang berguna untuk melihat kemampuannya, mengecek jika masih ada ditemukan error atau ada penambahan fitur-fitur yang belum ada pada sistem tersebut. Pengembangan diperlukan ketika adanya perubahan dari pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.6 Location Based Service

Steinger (2006), mengungkapkan bahwa *Location Based Service (LBS)* merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak. Selanjutnya menurut Mahmoud dalam A.N.Putra, T. D. Tambunan dan K. N. Ramadhan ”*Location-Based Service (LBS)* memberikan layanan personalisasi kepada pengguna perangkat bergerak (*mobile device*) yang disesuaikan dengan lokasi mereka saat ini. *LBS* membuka pasar baru bagi pengembang, operator jaringan seluler, dan penyedia layanan untuk mengembangkan dan memberikan nilai tambah layanan, memberikan informasi kondisi lalu lintas saat ini, menambahkan informasi rute perjalanan, membantu menemukan lokasi wisata terdekat, dan banyak lagi”.

Berdasarkan kedua pendapat ahli diatas, maka dapat disimpulkan bahwa *Location Based Service* merupakan sebuah layanan informasi berbasis lokasi yang memberikan informasi lokasi dari perangkat bergerak atau perangkat *mobile* dan dapat diakses secara *mobile* melalui jaringan telekomunikasi.

2.7 Google Maps API

Google Maps dirilis pada tahun 2005 sebagai layanan untuk aplikasi pemetaan di *World Wide Web* (Hu dan Dai, 2013). Selanjutnya Udeel (2009) mengungkapkan bahwa *Google Maps* dibuat dengan *XHTML (Extensible HTML)* dan diformat dengan *CSS (Cascading Style Sheet)*. *Google Maps* menyediakan *source code* yang bisa digunakan oleh *developer* atau pengembang untuk menerapkan *Google Maps* kedalam aplikasinya dalam bentuk *API (Application*

Programming Interface). *API* merupakan sebuah kumpulan rutin dan protokol yang melakukan spesifikasi bagaimana komponen perangkat lunak dapat saling berinteraksi antara satu dengan yang lain (Akanbi dan Agunbiade, 2013).

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google* (Shodiq, 2009), diantaranya :

- a. *Roadmap*, untuk menampilkan peta biasa 2 dimensi.
- b. *Satellite*, untuk menampilkan foto satelit.
- c. *Terrain*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
- d. *Hybrid*, untuk menunjukkan foto satelit yang di atasnya tergambar apa yang tampil pada *roadmap* (jalan dan nama kota).

2.8 Haversine Formula

Haversine Formula merupakan persamaan yang digunakan dalam sistem navigasi yang digunakan untuk mengukur jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan garis bujur dan lintang. Formula haversine merupakan metode untuk mengukur jarak antara dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar, melainkan sebuah bidang yang memiliki derajat kelengkungan. Penggunaan *haversine formula* mengasumsikan pengabaian efek elipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Rumus haversine formula dapat dijabarkan sebagai berikut :

$$x = (long2 - long1) * \cos \left(\frac{lat2 + lat1}{2} \right)$$

$$y = (lat2 - lat1)$$

$$d = \sqrt{x * x + y * y} * R$$

Keterangan Rumus

x = Longitude / Lintang

y = Latitude / Bujur

d = Jarak

R = Radius Bumi = 6371 km

1 derajat = 0,0174532925 radian

2.9 Global Positioning System (GPS)

Global Positioning System atau sering disebut *GPS* adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi (Winardi, 2006). *GPS* pertama kali dikembangkan oleh Amerika Serikat dengan tujuan untuk membantu pasukan Amerika Serikat dalam menghadapi perang Vietnam. Fungsi utama dari *GPS* adalah untuk memberikan informasi tentang posisi, kecepatan dan waktu secara akurat. Akurasi atau ketepatan perlu mendapat perhatian bagi penentuan koordinat sebuah titik atau lokasi. Koordinat posisi ini akan selalu mempunyai ‘faktor kesalahan’, yang lebih dikenal dengan ‘tingkat akurasi’. Misalnya alat tersebut menunjukkan sebuah titik koordinat dengan akurasi 3 meter, artinya posisi sebenarnya bisa berada dimana saja dalam radius 3 meter dari titik lokasi. Tingkat akurasi *GPS* sangat dipengaruhi oleh faktor sekeliling yang mengurangi kekuatan sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit, diantaranya:

1. Kondisi geografis, selama kita masih dapat melihat langit yang cukup luas, kekuatan sinyal semakin baik pula.
2. Hutan, makin lebat hutannya maka makin berkurang sinyal yang didapat.
3. Saat menyelam ke dalam air, *GPS* tidak akan berfungsi.
4. Berada diantara gedung tinggi atau di dalam gedung juga dapat mengganggu kekuatan sinyal.

GPS memungkinkan *developer* mengakses lokasi pengguna kapan pun. Sistem bisa melacak (*tracking*) lokasi ketika pengguna mengubah lokasi, bisa menentukan lokasi ponsel kemudian mengakses *web* untuk menentukan lokasi publik yang dekat dengan pengguna (Kasman, 2013).

Berdasarkan pendapat ahli tersebut maka dapat disimpulkan bahwa *GPS* merupakan suatu sistem yang digunakan untuk menentukan data posisi berdasarkan garis lintang dan bujur bumi dengan memanfaatkan 24 satelit yang mengirimkan sinyal makro ke bumi.

2.10 Teknologi Pendukung

Teknologi pendukung yang digunakan dalam penelitian ini dapat diuraikan sebagai berikut :

2.10.1 *Progressive Web App*

Pada tahun 2015 seorang *designer* Frances Berriman dan seorang *engineer* Alex Russel dari *Google Chrome* menciptakan istilah *Progressive Web App (PWA)* yang menjelaskan bagaimana aplikasi dapat memiliki fitur baru dari *modern browser* yaitu *service worker* dan *web app manifest*, yang kemudian mengajukan *developer* atau pengembang aplikasi *website* untuk mulai meng-*upgrade* aplikasi mereka ke dalam bentuk *PWA*. Konsep *Progressive Web App* yang ditawarkan oleh Frances Berriman dan Alex Russel yaitu aplikasi *website* yang dapat memberikan *user experience* dalam keandalan (*reliability*), kecepatan (*speed*), dan keterlibatan pengguna (*user engagement*).

Selanjutnya, Santoso (2019) mengatakan bahwa *Progressive Web App (PWA)* merupakan sebuah *website* yang dibangun dengan menggunakan teknologi pembuatan *website* yang modern, namun *website* tersebut dapat berlaku sebagai sebuah aplikasi *mobile*. *Progressive Web App* membawa pengaruh positif dalam pengembangan aplikasi *website*. Teknologi *PWA* ini masih terbilang baru, sehingga terdapat beberapa kelebihan dan kekurangan dari *PWA*.

Berikut beberapa kelebihan dan kekurangan dari penerapan teknologi *PWA* pada aplikasi berbasis *web* (Santoso, 2019) :

Kelebihan *PWA* :

1. Respon seperti *Mobile App*

Pada aplikasi *PWA*, seluruh konten *web* di-*cache* di dalam “*App-shells*” sehingga tampak seperti *Mobile App*. Dengan adanya *App-shell*, akan terbentuk tampilan dan nuansa aplikasi *mobile* reguler yang lengkap dengan elemen *User Interface* dan animasi. Hal ini yang kemudian membuat *PWA* menjadi sangat *user-friendly*.

2. *Layout responsive*

PWA telah menggunakan *responsive Layout* sehingga dapat menyesuaikan dengan ukuran layar pada perangkat *mobile* yang digunakan oleh pengguna aplikasi, walaupun ini bukanlah kelebihan utama dari aplikasi *PWA*.

3. Tersedia di mode *offline*

Setelah sebuah halaman *web* dibuka di *browser*, maka *cache* dari situs *web* tersebut tersimpan secara otomatis di dalam *web browser*. Hal ini nantinya membuat situs *web* tersebut dapat diakses secara *offline* dengan konten yang dimuat sebelumnya pada saat *online*.

4. *Add to Home Screen(A2HS)*

Fitur ini seperti fitur *bookmark page* pada *web browser*, pengguna dapat menambahkan aplikasi *PWA* ke layar *home* untuk akses lebih cepat saat membuka aplikasi dengan menggunakan *icon*.

5. *Push Notification*

Setelah membuka aplikasi *PWA* di *web browser*, pengguna akan diminta untuk mengizinkan pemberitahuan “*Allow push notification*” yang memungkinkan anda mendapatkan pesan dan peringatan cepat dari situs *web*.

Kekurangan *PWA* :

1. Perlu protokol *HTTPS*

Untuk menjalankan *Push Notification* dan “*in prompt build*”, aplikasi *web PWA* membutuhkan koneksi jaringan yang aman (*secure TL connection/SSL*).

2. Dukungan *web browser* modern

Saat penelitian ini dilakukan, aplikasi *PWA* baru dapat berjalan secara sempurna menggunakan *browser Google Chrome* dan *Opera*, sedangkan *Mozilla Firefox* baru sebagian fitur *PWA* yang berjalan dengan baik.

3. Dukungan pengguna

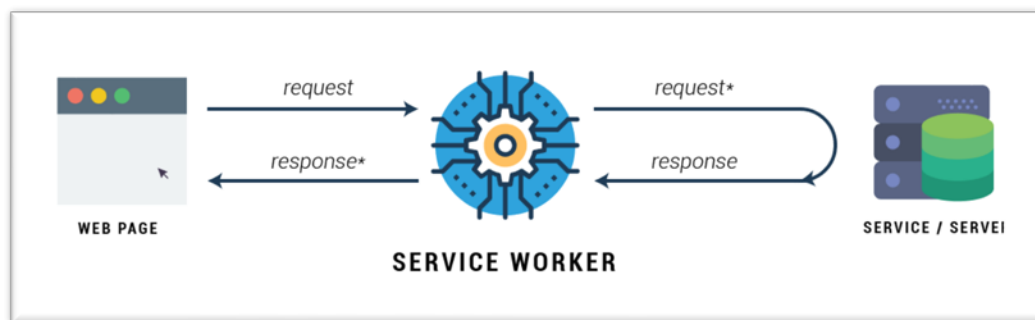
PWA didesain untuk mendapatkan perhatian banyak orang, namun masih perlu banyak dukungan lagi dari pengguna aplikasi *web* maupun *native*. Dari segi biaya, pengembangan aplikasi *web progressive* jauh lebih efisien dibandingkan aplikasi *native*.

2.10.2 *Service Worker*

Menurut Santoso (2019), *Service Worker* merupakan salah satu jenis *web worker*, *javascript* yang berjalan di belakang layar (*background*) tanpa mempengaruhi kinerja halaman *web*. *Service Worker* pada dasarnya adalah file *javascript* yang berjalan *client side* secara terpisah dari rangkaian *browser* utama

yang berfungsi untuk mencegah permintaan jaringan, melakukan *cache* atau mengambil sumber daya dari *cache* dan mengirimkan pesan.

Secara teknis, *Service Worker* menyediakan *script* “*network proxy*” di *web browser* untuk mengelola permintaan *web* (*HTTP request*) secara terprogram. *Service Worker* menggunakan mekanisme *cache* secara efisien dan memungkinkan perilaku *error-free* selama periode *offline*. Mekanisme kerja dari *Service Worker* dapat dilihat pada gambar 2.2.



Gambar 2. 2 Mekanisme kerja *Service Worker*

(Sumber : <https://itnext.io/service-workers-your-first-step-towards-progressive-web-apps-pwa-e4e11d1a2e85>)

Pada gambar di atas dijelaskan mengenai mekanisme kerja dari *service worker* yaitu menerima *request* dari halaman *web* kemudian mengirimkan ke *server*. Selanjutnya *service worker* bertugas menerima *response* dari *server* untuk diteruskan kembali kepada halaman *web*.

2.10.3 PHP

Welling dan Thomson, (2009) mengungkapkan bahwa *PHP* atau *hypertext preprocessor* merupakan bahasa pemrograman *server-side script* yang dirancang untuk pengembangan *web*. Selanjutnya Arief (2011) mengungkapkan bahwa *PHP* merupakan sebuah bahasa *server-side-scripting* yang menyatu dengan *HTML* yang berfungsi untuk membuat halaman *web* yang dinamis. *PHP* merupakan *server-side-scripting* sehingga sintaks dan perintah-perintah dalam *PHP* akan dieksekusi di *server*, kemudian hasilnya akan dikirimkan ke *browser* dalam format *HTML*.

Berdasarkan kedua pendapat ahli tersebut, dapat disimpulkan bahwa *PHP* merupakan bahasa yang bersifat *server-side-scripting* yang berguna dalam pembuatan *web* dinamis yang sintaks dan perintahnya dieksekusi didalam *server* untuk kemudian dikirimkan hasilnya ke *web browser* dalam bentuk format *HTML*.

2.10.4 *MySQL*

Welling dan Thomson (2005), mengungkapkan bahwa *MySQL* merupakan *DBMS* yang disebarluaskan secara gratis. Server *MySQL* mengontrol akses ke dalam data agar banyak pengguna bisa mengakses data tersebut secara bersamaan dan memastikan bahwa hanya pengguna tertentu yang dapat mengakses data tersebut. Sedangkan Arief (2011), mengungkapkan bahwa *MySQL* adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya.

Berdasarkan kedua pendapat ahli tersebut maka dapat disimpulkan bahwa *MySQL* merupakan sebuah *DBMS* yang berfungsi untuk mengontrol akses data ke dalam *database* serta melakukan pengelolaan terhadap data yang ada di dalam *database*.

2.10.5 *JavaScript*

Menurut Sigit (2011), *Javascript* merupakan bahasa *scripting* yang bekerja di sisi *Client/Browser* sehingga *website* bisa lebih interaktif. Selanjutnya Suryana (2014), mengungkapkan bahwa “*Javascript* adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen *HTML*. Dimana objek tersebut dapat berupa suatu *window*, *frame*, *URL*, dokumen, *form*, *button*, atau *item* yang lain”.

Berdasarkan kedua pendapat ahli tersebut maka dapat ditarik kesimpulan bahwa *Javascript* merupakan bahasa *scripting* berorientasi objek yang bekerja pada sisi *Client/Browser* untuk mengendalikan aspek interaksi pada atribut *HTML* dengan tujuan membuat *website* lebih interaktif.

2.10.6 *HTML*

HTML (HyperText Markup Language) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di dalam halaman *web* (Arief, 2011). Selanjutnya menurut Sibero (2011), *Hyper Text Markup Language* atau *HTML* adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk melakukan pertukaran dokumen *web*. Dokumen *HTML* terdiri dari komponen yaitu *tag*, *elemen* dan *attribute*. *Tag* adalah tanda awal < dan tanda akhir > yang digunakan sebagai pengapit suatu elemen. Elemen adalah nama penanda

yang diapit oleh tag yang memiliki fungsi dan tujuan tertentu pada dokumen *HTML*. Elemen dapat memiliki elemen anak dan juga nilai. Elemen anak adalah suatu elemen yang berada didalam elemen pembuka dan elemen penutup induknya. Nilai yang dimaksud adalah suatu teks atau karakter yang berada diantara elemen pembuka dan elemen penutup. Atribut adalah properti elemen yang digunakan untuk mengkhususkan suatu elemen. Elemen dapat memiliki atribut yang berbeda pada tiap masing-masingnya.

Berdasarkan beberapa pendapat ahli tersebut, maka dapat disimpulkan bahwa *HTML* merupakan bahasa yang digunakan dalam pembuatan halaman *web* yang berisi *tag*, *element* dan *attribute* yang memiliki fungsi masing-masing dalam halaman *web*.

2.10.7 CSS

Salah satu bahasa desain *web* yang dapat mengatur format tampilan sebuah halaman *web* dengan perancangan desain *text* berupa *font*, *color*, *margins*, *size* dan lain-lain. Menurut Kadir dan Triwahyuni (2013:323) “*CSS* adalah kode yang dimaksudkan untuk mengatur tampilan halaman *web*”. Kemudian, Arief (2011:11) menyatakan bahwa “*Client side scripting* adalah salah satu jenis bahasa pemrograman web yang proses pengolahannya dilakukan disisi *client*”.

Selanjutnya Sibero (2013:112) menyatakan bahwa, “*Cascading Style Sheets* memiliki arti Gaya Menata Halaman Bertingkat, yang artinya setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format element induknya”. Adapun contoh skrip dari *CSS* adalah sebagai berikut (Sibero, 2013:112) :

Kode Program 2. 1 Contoh skrip dari *CSS*

```
<div style= "font-weight:bold">
  <p>Contoh Paragraf dalam DIV dengan style font-weight:bold</p>
  <span> Contoh SPAN dalam DIV dengan style font weight:bold</span>
</div>
<div>
  <p>Contoh Paragraf dalam DIV tanpa style</p>
  <span> Contoh SPAN dalam DIV tanpa style</span>
</div>
```

Dari beberapa pendapat ahli diatas dapat disimpulkan bahwa *CSS*

(*Cascading Style Sheets*) merupakan salah satu jenis bahasa pemrograman untuk mengatur proses pengolahan pada komponen tampilan web menjadi bentuk web yang lebih indah dan menarik.

2.10.8 Codeigniter

Menurut Budi Raharjo (2015:3), “*CodeIgniter* adalah *framework web* untuk bahasa pemrograman *PHP* yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. EllisLab adalah suatu tim kerja yang berdiri pada tahun 2002 dan bergerak di bidang pembuatan *software* dan *tool* untuk para pengembang *web*”.

CodeIgniter memiliki banyak fitur atau fasilitas yang membantu para pengembang (*developer*) *PHP* untuk dapat membuat aplikasi *web* secara mudah dan cepat. Jika dibandingkan dengan *framework web PHP* lainnya, harus diakui bahwa *CodeIgniter* memiliki desain yang lebih sederhana dan bersifat fleksibel (tidak kaku). *CodeIgniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. *CodeIgniter* merupakan sebuah toolkit yang ditujukan untuk orang yang ingin membangun aplikasi *web* dalam bahasa pemrograman *PHP*.

Beberapa keunggulan yang ditawarkan oleh *CodeIgniter* sebagai berikut :

1. *CodeIgniter* adalah *framework* yang bersifat *free* dan *open-source*.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses instalasi, *framework CodeIgniter* hanya berukuran kurang lebih 2MB (tanpa dokumentasi atau jika direktori *user_guide* dihapus). Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6MB.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.
4. *CodeIgniter* menggunakan pola desain *Model-View-Controller (MVC)* sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara di kemudian hari.
5. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.
6. *CodeIgniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *CodeIgniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

2.10.9 *Bootstrap*

Bootstrap merupakan sebuah *framework CSS* yang memudahkan pengembang untuk membangun website yang menarik dan responsif. *Bootstrap* adalah *CSS* tetapi dibentuk dengan *LESS*, sebuah *pre-processor* yang memberi fleksibilitas dari penggunaan *CSS* biasa. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *web* yang mengutamakan desain (Otto, 2011). Selanjutnya Fauzi (2008:14), mengungkapkan bahwa “*Bootstrap* merupakan suatu metode berbasis komputer yang sangat potensial untuk dipergunakan pada masalah ketidakstabilan dan keakurasian, khususnya dalam menentukan interval konfendesi”.

Berdasarkan kedua pendapat ahli tersebut, dapat disimpulkan bahwa *bootstrap* merupakan sebuah *framework CSS* yang dirancang untuk membantu *developer* dalam melakukan desain terhadap rancangan *website* yang akan dibangun agar dapat lebih menarik, interaktif dan *responsive*.

2.11 Pengujian Perangkat Lunak

Pengujian merupakan elemen kritis dari jaminan terhadap kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Karakteristik umum dari pengujian perangkat lunak adalah sebagai berikut (Sukamto, 2009) :

1. Pengujian dimulai pada level modul dan bekerja keluar kearah integrasi pada sistem berbasiskan komputer.
2. Teknik pengujian yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. Pengujian diadakan oleh *software developer* dan untuk proyek yang besar oleh *group testing* yang *independent*.
4. *Testing* dan *debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomodasikan pada setiap strategi *testing*.

Selanjutnya, metode pengujian perangkat lunak ada 3 jenis, yaitu (Sukamto, 2009) :

1. *White Box / Glass Box* - pengujian operasi.
2. *Black Box* - untuk menguji sistem.

3. *Use Case* - untuk membuat input dalam perancangan *black box* dan pengujian *statebased*.

Dalam penelitian ini pengujian dilakukan dengan menggunakan metode skala likert, black box, dan pengujian kompatibilitas aplikasi.

2.11.1 *Pengujian Skala Likert*

Perhitungan hasil dari Kuesioner dilakukan dengan menggunakan cara *Skala Likert*. *Skala Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono, 2014). Untuk setiap pilihan jawaban diberi skor, maka responden harus menggambarkan, mendukung pertanyaan dengan jawaban yang dipilih. Dengan *skala likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak ukur menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan.

Tabel 2. 1 Skala Penilaian Untuk Pertanyaan Positif dan Negatif

Nilai	Kriteria
1	Sangat Buruk
2	Buruk
3	Cukup
4	Baik
5	Sangat Baik

2.11.2 *Pengujian Black Box*

Menurut Pressman (2010), *black box testing* juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Selanjutnya Shalahuddin dan Rosa (2011) mengungkapkan bahwa *black box testing* merupakan pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Berdasarkan kedua pendapat ahli di atas, dapat disimpulkan bahwa pengujian *black box* merupakan pengujian perangkat lunak dari segi fungsional

suatu program atau aplikasi yang bertujuan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari program atau aplikasi tersebut telah sesuai dengan spesifikasi yang dibutuhkan.

2.11.3 *Pengujian Kompatibilitas*

Pada tahap pengujian kompatibilitas ini, pengujian dilakukan dengan mengakses dan memasang aplikasi yang telah dibuat pada beberapa *device* dan *browser* yang berbeda. Dalam pengujian ini, akan terlihat apakah aplikasi dapat berjalan dapat digunakan pada berbagai *browser* dengan versi yang berbeda. Berikut contoh tabel pengujian kompatibilitas aplikasi.

No	Tipe Perangkat	Jenis Browser	Keterangan Pengujian	Screenshot Hasil
1.				
2.				
3.				
...				
Dst.				