

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Definisi Sistem

Menurut (Hutahaean, 2012:1) sistem merupakan sekumpulan dari elemen-elemen yang bekerja sama atau berinteraksi untuk mencapai suatu tujuan yang telah ditetapkan. Sistem yang dimaksud dalam hal ini menggambarkan suatu kejadian atau peristiwa dan satu kesatuan yang nyata, seperti tempat, benda, dan orang-orang yang benar-benar ada dan terjadi.

Secara garis besar, sistem merupakan definisi dari segala sesuatu baik berbentuk objek yang abstrak maupun objek yang nyata, yang saling terkait dan menjadi satu kesatuan, serta bekerja sama dengan sedemikian rupa untuk mencapai suatu tujuan tertentu. Pada suatu sistem, akan terdapat hubungan antara beberapa unsur elemen yang saling bekerja sama demi mencapai tujuan. Salah satu syarat sistem yang baik adalah sistem tersebut harus dibuat dengan tujuan untuk menyelesaikan suatu masalah.

Didalam suatu sistem, pasti terdapat serangkaian struktur ataupun proses yang akan berjalan didalamnya. Serangkaian proses inilah yang nantinya akan berjalan sesuai dengan apa yang akan dicapai nantinya. Elemen istem terdiri dari elemen masukan (*input*), elemen proses (*process*), dan elemen keluaran (*output*).

2.1.2 GPS (*Global Positioning System*)

Global Positioning System atau biasa disebut dengan GPS, merupakan suatu sistem navigasi yang memanfaatkan teknologi satelit untuk mendapatkan suatu posisi tertentu secara langsung.

Menurut (Milner, 2016:9) dalam penelitiannya yang berjudul “*What is GPS?*”, teknologi GPS bukan hanya digunakan untuk mencari arah, namun dapat juga digunakan untuk berbagai macam hal, yang tentunya berhubungan dengan koordinat posisi, navigasi, dan juga operasi waktu. Teknologi ini juga dapat digunakan dimanapun, di seluruh penjuru bumi, dan juga dapat digunakan dalam kondisi cuaca apapun.

Menurut (Perkasa, 2019:22) dalam penelitiannya yang berjudul “Penggunaan *Global Positioning System (GPS)* Untuk Dasar Survey Pada Mahasiswa”, sistem GPS terbagi menjadi tiga segmen, diantaranya adalah sebagai berikut :

1. Satelit, berfungsi untuk menerima sinyal dan menyimpan segala data yang ditransmisikan oleh stasiun pengontrol. Selain itu, satelit juga memancarkan sinyal dan informasi secara terus menerus ke *receiver* yang dimiliki oleh *user*.
2. Pengontrol, berfungsi untuk mengendalikan satelit dari bumi. Pengontrol dapat melakukan pengecekan terhadap kesehatan satelit, prediksi orbit dan waktu, dan lain sebagainya.
3. Penerima (*receiver*), berfungsi untuk menerima data dari satelit pemancar. Setelah sinyal berhasil ditangkap oleh *receiver*, selanjutnya

receiver akan memproses data tersebut dan menjadikannya sebagai koordinat posisi.

Konsep GPS tidak luput dari istilah *tracking*. *Tracking* adalah suatu aktifitas yang dilakukan oleh seseorang untuk keperluan melacak, melihat, ataupun memantau letak atau posisi dari suatu objek, baik itu benda mati ataupun benda yang hidup. Dengan kata lain, apabila suatu sistem sudah menanamkan teknologi GPS didalamnya, maka sistem tersebut sudah bisa melakukan *trace* atau *tracking* dari posisi suatu objek tersebut. *Tracking* juga bervariasi macamnya, mulai dari *real time tracking*, ataupun *tracking* secara periodik, tergantung pada sistem yang digunakan. *Tracking* secara *real time* sudah sering dijumpai dalam kehidupan sehari-hari, salah satu contohnya adalah penggunaan *Google Maps* untuk menunjukkan posisi suatu bangunan, sedangkan *tracking* secara periodik dapat ditemukan pada aplikasi *e-commerce*.

2.1.3 Aplikasi Berbasis Web

Menurut (Elgamar, 2020:3), *website* adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya.

Dalam penerapannya, *website* dapat dijalankan melalui jaringan lokal (*localhost*), ataupun melalui jaringan internet. Dalam pengaksesannya, kita membutuhkan web browser sebagai aplikasi bantuan untuk pengaksesan halaman *website*.

Suatu *website* pasti memiliki *domain* dan *hosting* dalam implementasinya. *Domain* akan berfungsi sebagai alamat (URL) pada suatu *website*, sedangkan *hosting* akan berfungsi sebagai suatu media untuk menyimpan segala file *website* yang sudah dibuat. Sederhananya, *hosting* akan berperan sebagai tempat ataupun rumah untuk menyimpan segala data, dan *domain* merupakan alamat dari tempat tersebut.

Aplikasi berbasis web, pada umumnya dibangun dari berbagai macam bahasa pemrograman, tergantung dengan *programmer* yang membuatnya. Pada umumnya, *website* menggunakan bahasa pemrograman PHP, dan dikombinasikan dengan penggunaan HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheet*), dan Javascript. PHP akan berfungsi sebagai alur yang menjalankan segala logika, HTML akan berfungsi sebagai pondasi dasar dalam aplikasi web, CSS akan berfungsi sebagai penghias dari halaman website agar dapat terlihat lebih menarik, dan yang terakhir adalah Javascript yang akan berfungsi sebagai bahasa yang dapat memperkaya isi *website* agar lebih *powerfull*.

2.1.4 PHP (*Hypertext Preprocessor*)

PHP adalah salah satu bahasa pemrograman yang dikhususkan untuk pengembangan aplikasi web. Pada saat ini, penggunaan bahasa pemrograman PHP untuk keperluan pembuatan *website* sudah sangat menjamur. Bahasa pemrograman PHP sangat populer di kalangan *programmer* web. Hal ini dikarenakan sintaks pemrograman PHP mudah untuk dipelajari, terutama bagi para pemula yang ingin belajar mengembangkan aplikasi berbasis web.

PHP bersifat *open source*, artinya bahasa pemrograman ini dapat dipakai secara gratis. Bahasa pemrograman PHP juga mampu dijalankan diberbagai *platform*. PHP sendiri berjalan pada *server side scripting*. Artinya, untuk dapat menjalankan bahasa pemrograman PHP, wajib ada web server sebagai server yang nantinya akan melayani *request* dari *client*.

A. Sejarah PHP

Pada mulanya, bahasa pemrograman PHP adalah kependekan dari *Personal Home Page*. Dengan kata lain, situs yang dikembangkan hanya bersifat personal saja. Bahasa pemrograman PHP pertama kali diciptakan oleh seseorang yang bernama Rasmus Lerdorf pada tahun 1995. Pada awalnya, bahasa pemrograman PHP berwujud sebagai sekumpulan *script* yang dipakai untuk mengolah suatu formulir data dari *website*.

Pada tahun 1997, bahasa pemrograman PHP ditulis dan direvisi ulang oleh sebuah perusahaan yang bernama Zend. Hasilnya, bahasa pemrograman PHP berubah menjadi bahasa yang lebih baik dan lebih cepat. Kemudian pada tahun 1998, tepatnya pada bulan Juni, perusahaan tersebut merilis interpreter baru untuk PHP dan disebut sebagai PHP 3.0. Pada saat itulah singkatan PHP mengalami perubahan menjadi PHP : *Hypertext Preprocessing*. Setelah itu, bahasa pemrograman PHP sudah banyak mengalami perkembangan dan perubahan.

B. Sintaks dasar

Penulisan bahasa pemrograman PHP tidak jauh berbeda dengan sintaks penulisan bahasa pemrograman lain. Bahasa pemrograman PHP memiliki *tag* pembuka dan penutup. Biasanya, bahasa pemrograman PHP ditulis

bersamaan dengan skrip HTML. Dengan kata lain, kita dapat menyisipkan skrip PHP kedalam HTML, begitupun sebaliknya. Berikut ini merupakan struktur penulisan dari sintaks dasar PHP :

```
<?php
    //ini merupakan skrip yang ingin kita tulis
?>
```

2.1.5 HTML (*Hypertext Markup Language*)

Menurut (Winarno et al., 2014:1) HTML atau *Hypertext Markup Language* merupakan suatu jenis bahasa yang digunakan oleh *website* dalam menampilkan konten. Sebuah dokumen yang berbentuk HTML pada dasarnya merupakan suatu jenis dokumen yang dapat di *edit* atau diubah dengan teks *editor* atau biasa disebut dengan aplikasi pengubah teks..

HTML (*Hypertext Markup Language*) merupakan suatu bahasa markah yang digunakan dalam membuat serta merancang struktur dalam aplikasi web. Didalam penerapannya dalam pembuatan web, HTML (*Hypertext Markup Language*) berfungsi sebagai pondasi yang akan menopang keseluruhan aplikasi web. HTML (*Hypertext Markup Language*) di tulis dengan elemen yang dinamakan *tag*. *Tag-tag* ini nantinya akan disusun sedemikian rupa, sehingga terbangunlah struktur aplikasinya. *Tag* ini sendiri terbagi menjadi dua bagian, yaitu *tag* awal dan *tag* akhir. *Tag* awal biasanya ditulis dengan format `<nama tag>`, yang menandakan awal dari *tag* itu sendiri. Sedangkan *tag* akhir biasanya ditulis dengan format `</nama tag>`, yang akan menandakan akhir dari *tag*. Didalam penulisan *tag*,

HTML (*Hypertext Markup Language*) tidak membedakan penggunaan huruf besar maupun huruf kecil.

Didalam HTML (*Hypertext Markup Language*), dapat disisipi Bahasa pemrograman lain yang nantinya akan mendukung aplikasi itu sendiri, sehingga aplikasi yang dibangun dapat lebih kompleks dan lebih *powerful*. Contoh yang paling sering ditemui pada aplikasi web yang menyisipkan bahasa pemrograman PHP dan Javascript dalam penulisannya. Hal ini tentu akan membuat aplikasi tersebut lebih dinamis dan interaktif ketika digunakan, sehingga aplikasi yang dibangun tidak terlihat kaku.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 </head>
6 <body>
7
8 </body>
9 </html>
```

Gambar 2.1 Struktur Dasar HTML

2.1.6 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) merupakan sekumpulan perintah yang digunakan dalam pembuatan aplikasi berbasis web. Kumpulan perintah ini digunakan untuk membuat serta merancang tampilan sebuah situs halaman *website*. Dalam pembuatan *website*, CSS (*Cascading Style Sheets*) berfungsi sebagai komponen yang akan menghiasi halaman web. Tampilan yang akan dihasilkan dalam pembuatan web jika menggunakan CSS (*Cascading Style Sheets*) akan terlihat lebih *soft* dan menarik. Didalam CSS (*Cascading Style Sheets*), kita juga

dapat membuat animasi yang nantinya akan menambah daya tarik dari *website* yang dibuat.

Pada dasarnya, penggunaan CSS (*Cascading Style Sheets*) bukanlah suatu keharusan atau kewajiban dalam membuat aplikasi web. Artinya, suatu halaman web tidak harus menggunakan CSS (*Cascading Style Sheets*) dalam pembuatannya. Namun, jika dalam pembuatan web tidak dibantu oleh CSS (*Cascading Style Sheets*), tampilan web akan terlihat sangat tidak menarik dan kaku. Segala pewarnaan, peletakan objek, akan menjadi sulit dan terlihat klasik. Itulah sebabnya, mengapa *programmer* wajib menggunakan CSS (*Cascading Style Sheets*) dalam membuat aplikasi web, karena selain membuat aplikasi itu terlihat menarik, CSS (*Cascading Style Sheets*) juga dapat menambah daya tarik pengunjung web dalam membuka aplikasi web itu sendiri.

Dalam menyisipkan atau menggunakan CSS kedalam web yang dibuat, terdapat beberapa metode yang berbeda. Pada dasarnya, metode-metode yang ada sama saja dan memiliki tujuan yang sama pula. Hanya saja yang menjadi perbedaannya adalah pada cara peletakannya saja. Berikut ini merupakan tiga cara dari penyisipan dokumen CSS :

A. Metode Internal CSS

Metode internal CSS dapat juga disebut sebagai *Embedded CSS*. Ciri utama dari penggunaan metode ini dapat terlihat dari *tag* `<style>` yang terdapat pada dokumen HTML, baik itu disisipkan pada *tag* `<head>` ataupun pada *tag* `<body>`. Berikut merupakan struktur penulisan dari metode internal CSS :


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Contoh Penerapan Metode Internal CSS</title>
5   <!-- penulisan internal css dalam tag head -->
6   <style type="text/css">
7     p{
8       font-family: serif;
9       line-height: 1.75em;
10      font-size: 18px;
11    }
12    i {
13      font-family: sans;
14      color: orange;
15    }
16  </style>
17 </head>
18
19 <body>
20 <!-- penulisan internal css dalam tag body -->
21 <style type="text/css">
22   h2 {
23     font-family: sans;
24     color: #333;
25   }
26 </style>
27 <h2>Ini judul website yang kita buat</h2>
28 <p>ini adalah contoh dari isi website yang kita buat dalam bentuk paragraf.</p>
29 </body>
30 </html>

```

Gambar 2.2 Metode Internal CSS

B. Metode Eksternal CSS

Penerapan atau penggunaan metode eksternal CSS berbeda dengan internal CSS. Sesuai dengan namanya, eksternal CSS berarti file CSS yang dipakai berada pada luar file HTML. File ini akan diberi nama dengan ekstensi (.css). selanjutnya, apabila ingin menyambungkannya pada file HTML yang dibuat, kuncinya adalah pada pemanggilan file css tersebut.

Berikut merupakan struktur penggunaan metode eksternal CSS :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ini adalah Contoh Eksternal CSS</title>
5   <!-- tepat dibawah ini merupakan perintah untuk memanggil file
6   eksternal css -->
7   <link rel="stylesheet" type="text/css" href="style-ku.css">
8 </head>
9
10 <body>
11   <h2>Ini judul website yang dibangun</h2>
12   <p>Ini adalah contoh dari isi website yang berbentuk paragraf
13   </p>
14 </body>
15 </html>

```

Gambar 2.3 Metode Eksternal CSS

C. Metode *Inline* CSS

Metode inline CSS merupakan metode yang kodenya langsung ditulis didalam atribut elemen yang terdapat pada tag HTML. Setiap tag yang ada pada HTML dapat memanggil atribut style, sehingga penulisan coding CSS dapat langsung ditulis pada tag itu juga. Berikut merupakan struktur penulisan dari metode inline CSS :

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Contoh metode Inline CSS</title>
5  </head>
6
7  <body>
8
9    <h2 style="color:red;font-family:sans">Ini Judul bakal jadi
    warna merah karena ada CSS nya</h2>
10   <p style="color:maroon">paragraf ini akan tampil dengan warna
    merah maroon!</p>
11 </body>
12 </html>

```

Gambar 2.4 Metode *Inline* CSS

2.1.7 MYSQL Database

Menurut (Musiafa, 2019:8) *database* adalah sekumpulan data yang berhubungan serta terintegrasi dengan baik antara satu sama lain. *Database* berisi kumpulan dari data-data yang disimpan secara terstruktur. Didalam dunia pemrograman, penggunaan *database* sangat diperlukan karena *database* digunakan sebagai media penampungan data. Data yang tersimpan didalam *database* akan akan diolah kembali menjadi suatu informasi.

Untuk dapat mengelola ataupun mengolah *database*, diperlukan sebuah *software* atau aplikasi dalam pengelolaannya. Aplikasi ini biasanya disebut sebagai DBMS (*Database Management System*). DBMS merupakan suatu *software* aplikasi

atau perangkat lunak yang memungkinkan *user* untuk mengelola data, seperti menambahkan, menghapus, ataupun merubah segala isi data yang terdapat didalam *database*. Dengan kata lain, DBMS yaitu suatu perangkat lunak yang mampu untuk melakukan proses manajemen dari sistem *database*. Kehadiran dari DBMS ini akan membuat *user* lebih praktis dan efisien dalam mengolah data yang terdapat pada *database*.

Pada dunia pemrograman, terdapat banyak sekali jenis DBMS. Berikut merupakan contoh dari beberapa aplikasi DBMS yang sering ditemui atau cukup populer dalam kalangan *programmer* :

1. MYSQL
2. Microsoft Access
3. Microsoft SQL Server
4. Oracle
5. PostgreSQL
6. MariaDB

Salah satu DBMS yang cukup banyak digunakan di dunia pemrograman adalah MYSQL. Dalam penerapan dan penggunaanya, MYSQL menggunakan sintaks dasar SQL (*Structured Query Language*). *Database* MYSQL juga bersifat *open source*, artinya dapat digunakan secara gratis tanpa harus membeli lisensi yang ada, baik untuk keperluan pribadi ataupun komersil.

2.1.8 UML (*Unified Modeling Language*)

UML atau *Unified Modeling Language*, merupakan suatu standar pemodelan yang digunakan dengan tujuan untuk membantu *programmer* dalam memvisualisasikan sebuah sistem atau *software* perangkat lunak yang akan dibuat. UML (*Unified Modeling Language*) menggunakan konsep OOP (*Object Oriented Programming*) dalam menerapkan bahasa pemodelannya. Dalam penggunaannya, UML (*Unified Modeling Language*) tidak hanya digunakan untuk memodelkan perangkat lunak saja, melainkan dapat juga digunakan untuk pemodelan semua bidang yang membutuhkan design pemodelan dalam pembuatannya.

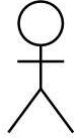



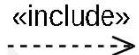
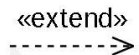
UML (*Unified Modeling Language*) terdiri dari beberapa jenis diagram. Setiap diagram tentunya memodelkan bagian yang berbeda-beda pula. Berikut ini merupakan jenis-jenis diagram yang terdapat pada UML (*Unified Modeling Language*) :

A. *Use Case* Diagram

Use Case diagram merupakan suatu teknik yang digunakan untuk menjelaskan interaksi yang terjadi antar “aktor” yang saling berinteraksi atau beraktivitas pada suatu sistem. Selain itu, *use case* diagram juga akan menjelaskan tentang tipe interaksi bagaimana yang terjadi antara aktor selaku pengguna sistem dan sistem yang sedang berjalan. *Use case* diagram akan menggambarkan secara garis besar urutan proses yang akan berjalan pada sistem yang nantinya akan dibuat. Proses yang digambarkan adalah proses yang menunjukkan interaksi antar pengguna didalam sistem. Melalui gambar pada diagram ini, maka dapat diketahui apa saja fungsi yang dapat digunakan

pada sistem yang dibangun. Dalam *use case* diagram, terdapat beberapa jenis simbol atau komponen, dan berikut merupakan rinciannya :

Tabel 2.1 *Use Case* Diagram






SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Peran seseorang yang nantinya akan berinteraksi dengan sistem yang berjalan
	<i>Use Case</i>	Suatu bentuk dari gambaran fungsional sistem
	<i>Association</i>	Penghubung antara sistem dengan aktor
	Generalisasi	Elemen spesialisasi dari elemen lainnya
	<i>Include</i>	Petunjuk bahwa suatu use case seluruhnya merupakan fungsionalitas dari <i>use case</i> lain
	<i>Extend</i>	Petunjuk bahwa suatu use case merupakan tambahan fungsionalitas dari <i>use case</i> lainnya apabila kondisi tertentu terpenuhi

B. *Activity* Diagram

Activity diagram merupakan diagram yang menggambarkan aktivitas yang terjadi dalam sistem. *Activity* diagram menggambarkan *workflow* (aliran kerja) dari sebuah sistem. *Activity* diagram akan menjelaskan ataupun memperlihatkan serangkaian urutan mengenai aktivitas yang terjadi pada sistem dari awal hingga akhir. Dalam pembuatannya, *activity* diagram dibuat

berdasarkan *usecase* diagram. Didalam *activity* diagram, terdapat sejumlah komponen, dan berikut merupakan rinciannya :

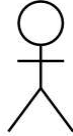
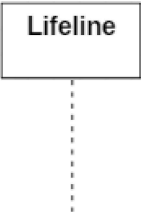
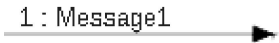


Tabel 2.2 *Activity* Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Initial State/Status Awal</i>	Awal dimulainya aliran kerja
	<i>Activity/Aktivitas</i>	Suatu bentuk pekerjaan yang dilakukan pada aliran kerja
	<i>Decision</i>	Suatu bentuk percabangan dimana terdapat pilihan kondisi yang lebih dari satu
	<i>Final State/Status Akhir</i>	Bagian akhir dari suatu aliran kerja
	<i>Transition</i>	Sebagai penghubung antara aktivitas sebelumnya dan aktivitas selanjutnya

C. *Sequence* Diagram

Sequence diagram merupakan diagram yang menggambarkan atau menjelaskan mengenai interaksi yang terjadi antara objek satu dengan objek yang lain pada sistem. *Sequence* diagram ini berfungsi untuk menampilkan serangkaian pesan yang berjalan pada sistem, dengan kata lain pesan antar objek yang sedang berjalan akan dijelaskan pada diagram ini. Sesuai dengan namanya, pesan yang ditampilkan akan disajikan secara berurutan (sekuensial). Didalam *sequence* diagram, terdapat sejumlah komponen. Berikut merupakan rinciannya :

Tabel 2.3 *Sequence Diagram*








SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Peran seseorang yang nantinya akan berinteraksi dengan sistem yang berjalan
	<i>Lifeline</i>	Menghubungkan <i>sequence</i> selama objek (<i>Message</i> dikirim dan diterima serta aktivasinya.)
	<i>Message</i>	Sebuah pesan yang mendefinisikan komunikasi tertentu antara <i>Lifelines</i> dari sebuah interaksi
	<i>Reply Message</i>	Jenis pesan yang mewakili informasi yang dikirimkan kembali ke pengirim pesan berdasarkan pesan sebelumnya
	<i>Include</i>	Jenis pesan yang mewakili permohonan pesan dari <i>Lifeline</i> yang sama

D. *Class Diagram*

Class diagram atau disebut juga dengan diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* menggambarkan struktur dan deskripsi *class* serta hubungan antar kelas. Diagram *class* akan memetakan bagaimana struktur dari sistem yang sedang dibuat, dengan memvisualisasikan hubungan antar objek dengan memodelkan kelas dan segala atribut yang ada

didalamnya. Berikut merupakan komponen yang terdapat didalam *class* diagram :

Tabel 2.4 Class Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

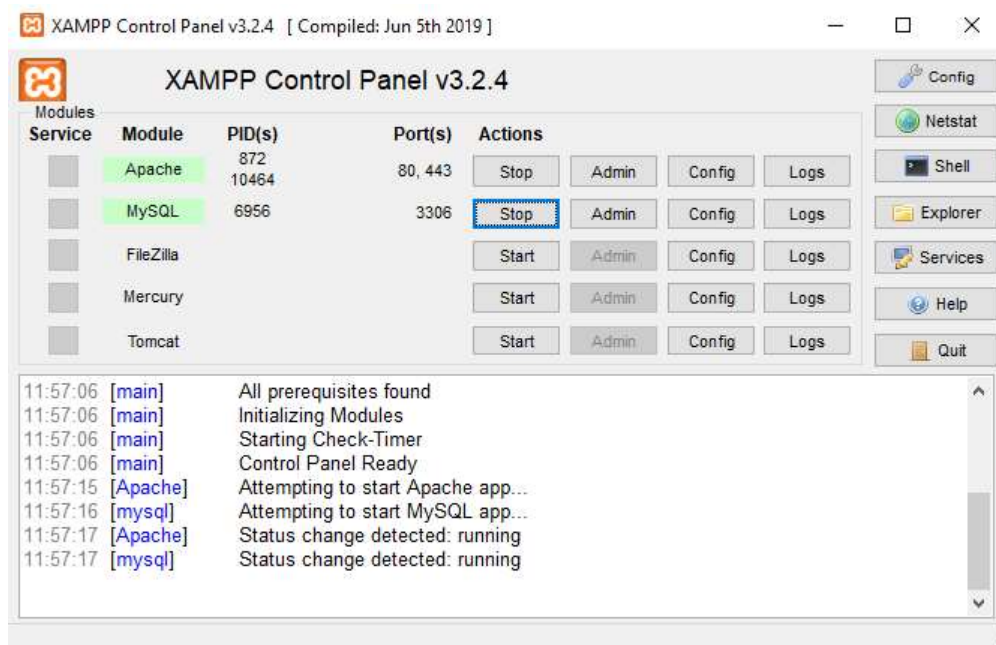
2.2 Software Pendukung

Pada pembuatan suatu aplikasi, tentunya diperlukan *software* tertentu dalam proses pembuatannya. *Software* yang digunakan tentunya beragam, sesuai dengan keperluan *programmer* itu sendiri. Didalam pembuatan aplikasi ini, peneliti menggunakan beberapa *software* atau aplikasi pendukung yang akan dijelaskan pada perincian berikut ini :

2.2.1 XAMPP

XAMPP merupakan sebuah perangkat lunak (*software*) yang biasanya digunakan sebagai server untuk melakukan kompilasi dari program web. XAMPP adalah perangkat lunak (*software*) yang bersifat gratis dan dapat digunakan oleh siapa saja tanpa harus membeli lisensi (*freeware*). XAMPP nantinya akan berfungsi sebagai *server standalone (localhost)* pada saat ingin membuat sebuah program. Artinya, pada saat membuat sebuah perangkat lunak berbasis web, setiap *programmer* wajib memiliki aplikasi yang menjalankan web *server*, agar *programmer* dapat melakukan pengujian sistem secara lokal (*offline*).

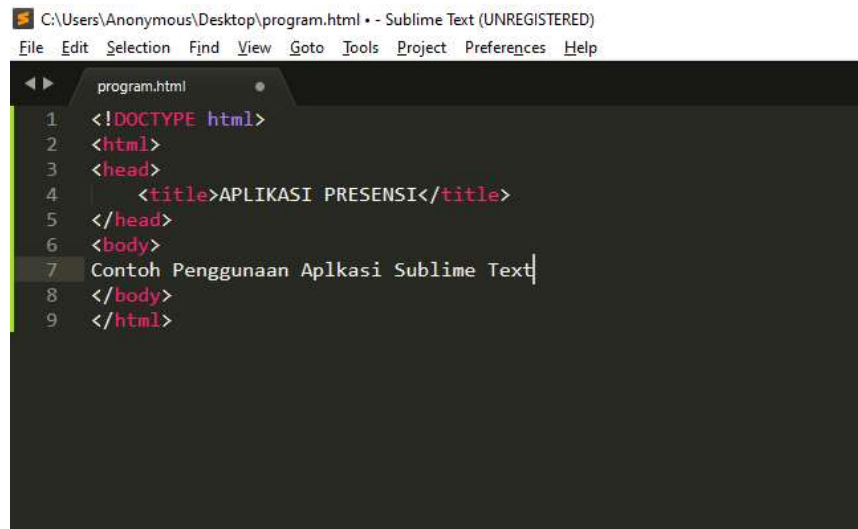
Dalam kalangan *programmer* web, XAMPP cukup populer dan sering digunakan, karena bersifat gratis dan mudah untuk dioperasikan. Selain itu, XAMPP dapat berjalan pada sistem operasi *cross platform*. Artinya, aplikasi ini dapat berjalan disegala sistem operasi, baik itu Windows, Linux, ataupun MacOS. XAMPP mencakupi semua program yang berguna dalam membangun suatu *website*, seperti Apache, MYSQL/MariaDB, PHP, dan Perl. Walau aplikasi ini tergolong cukup lengkap, namun aplikasi ini juga termasuk kedalam aplikasi yang ringan dan sederhana.



Gambar 2.5 XAMPP Control Panel

2.2.2 Sublime Text

Sublime Text merupakan salah satu *software* yang berfungsi sebagai *text editor*. Dalam dunia pemrograman, aplikasi ini sangat banyak dijumpai. Hal ini dikarenakan aplikasi ini cukup canggih jika digunakan dalam membuat sebuah program (*coding*). Aplikasi ini juga dapat digunakan untuk berbagai macam bahasa pemrograman, salah satunya adalah PHP. Selain itu, aplikasi ini juga memungkinkan kita untuk memilih tema sesuai dengan yang kita inginkan. Hal ini tentu akan menjadi daya tarik tersendiri dalam menggunakan aplikasi ini untuk melakukan *coding*.



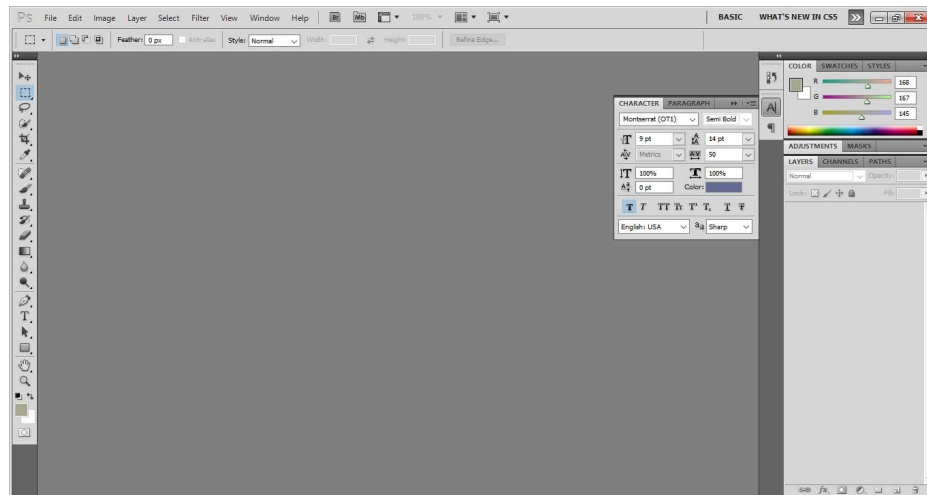
```
C:\Users\Anonymous\Desktop\program.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

program.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>APLIKASI PRESENSI</title>
5 </head>
6 <body>
7 Contoh Penggunaan Aplkasi Sublime Text|
8 </body>
9 </html>
```

Gambar 2.6 *Sublime Text*

2.2.3 Photoshop CS5

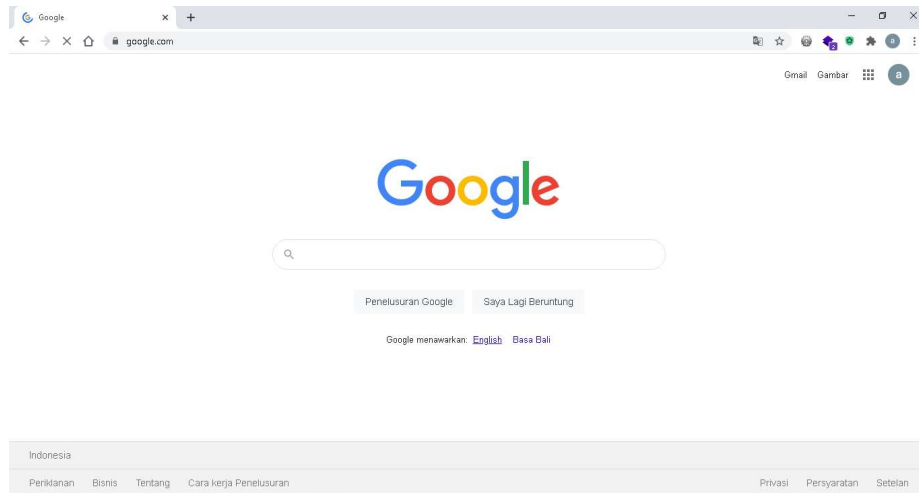
Adobe Photoshop atau sering dikenal dengan Photoshop, merupakan salah satu aplikasi yang berfungsi untuk desain grafis. Photoshop merupakan aplikasi *editing* yang sangat *powerful* dalam melakukan pengeditan gambar. Banyak sekali fitur-fitur yang dapat digunakan didalamnya. Aplikasi ini sangat sering digunakan dalam dunia *editing*, karena terbukti dapat menghasilkan *output* gambar yang sangat menarik. Aplikasi ini tentunya juga berkaitan didalam pembuatan web. Aplikasi ini digunakan untuk medesain ataupun mengedit objek gambar yang nantinya akan dimasukan kedalam *website* yang dibangun. Bukan hanya untuk desain web saja, aplikasi ini juga dapat digunakan oleh beberapa profesi seperti fotografer, seniman kreatif, serta desainer.



Gambar 2.7 Photoshop CS5

2.2.4 Google Chrome

Google Chrome merupakan salah satu *software* atau aplikasi yang dibuat oleh salah satu perusahaan teknologi yang cukup besar didunia, yaitu Google. Aplikasi ini merupakan jenis aplikasi *web browser* yang digunakan untuk kepentingan pengguna dalam melakukan *browsing* atau berselancar pada dunia maya. Aplikasi ini sama seperti aplikasi *web browser* pada umumnya, yaitu berfungsi untuk menampilkan struktur halaman web. Namun, Google Chrome saat ini tidak hanya berfungsi untuk menampilkan halaman web saja. Google Chrome juga dapat digunakan untuk menampilkan beberapa jenis *file* yang ada dikomputer, seperti *file* PDF, *file* foto, dan lain sebagainya. Saat ini, Google Chrome sudah banyak digunakan didunia, karena selain gratis, aplikasi ini juga terbilang cukup mudah untuk digunakan, dan tersedia pada beberapa jenis *platform* (*multi-platform*).



Gambar 2.8 Google Chrome

2.3 Penelitian Terdahulu

Penelitian ini mengacu pada penelitian terdahulu untuk mempermudah peneliti dalam mengumpulkan data dan juga sebagai acuan peneliti dalam melakukan penelitian. Peneliti mengangkatnya sebagai bahan referensi dalam memperkaya bahan kajian pada penelitian ini. Berikut ini merupakan penelitian terdahulu yang terkait dengan judul penelitian ini :

1. Penelitian Syaiful Amrial Khoir, Anton Yudhana, dan Sunardi (2020) berjudul “Implementasi GPS (*Global Positioning System*) Pada Presensi Berbasis Android DI BMT Insan Mandiri”, yang terbit dalam Jurnal Sains Komputer & Informatika (J-SAKTI), Volume 4 Nomor 1 Maret 2020, pp. 9-17, ISSN : 2548-9771 / eISSN : 2549-7200. Permasalahan yang diangkat pada penelitian tersebut adalah terdapatnya kendala yang dialami oleh para karyawan yang bekerja sebagai marketing pada perusahaan tersebut. Para tenaga marketing

dinilai sulit untuk melakukan presensi hadir, presensi pulang, maupun presensi pada saat keluar kantor karena tingginya aktivitas marketing yang berhubungan langsung dengan anggota ataupun nasabah diluar kantor. Hal tersebut tentu akan membuat tenaga marketing mengalami pemotongan gaji, ataupun mendapatkan surat teguran dari manager operasional karena tidak melakukan absensi. Dengan adanya permasalahan tersebut, maka peneliti menganggap perlu dilakukan sebuah inovasi dan solusi dalam proses absensi, yaitu melalui aplikasi android.

2. Penelitian Febrian Murti Dewanto, Bambang Agus Herlambang, dan Aris Tri Jaka Harjanta (2017) berjudul “Pengembangan Sistem Informasi Absensi Berbasis *Radio Frequency Identification* (RFID) Terintegrasi dengan Sistem Informasi Akademik”, yang terbit dalam Jurnal Informatika:Jurnal Pengembangan IT (JPIT), Vol. 02, No. 02, Juli 2017, ISSN : 2477-5126 / eISSN : 2548-9356. Permasalahan yang diangkat dalam penelitian tersebut adalah diawali dengan banyaknya mahasiswa yang datang terlambat dalam perkuliahan. Hal ini tentunya akan menjadi pemicu timbulnya sistem titip absen. Oleh karena itu, tentu dosen akan susah untuk mengetahui jumlah akurat mahasiswa yang hadir. Berdasarkan permasalahan tersebut, maka peneliti tersebut ingin membuat sebuah sistem absen berbasis *Radio Frequency Identification* (RFID). Sistem yang dikembangkan ini nantinya akan terhubung dengan sistem akademik yang digunakan. Dalam

pembuatannya, peneliti menggunakan bahasa pemrograman PHP.

3. Penelitian Rizky Pratama Putra, Haeruddin, dan Novianti (2019) berjudul “Aplikasi Absensi Perkuliahan Berbasis Android Studi Kasus Absensi FKTI Universitas Mulawarman”, yang terbit dalam Jurnal Teknologi Informasi (JURTI), Vol. 3 No. 1, Juni 2019, ISSN : 2579-8790. Permasalahan yang diangkat didalam penelitian tersebut adalah dikarenakan aktivitas pada absensi perkuliahan pada Universitas Mulawarman masih manual menggunakan kertas. Hal ini tentunya akan menjadi masalah, karena beresiko rusak atau hilangnya data absen tersebut. Selain itu, mahasiswa juga dapat melakukan penitipan absen jika menggunakan metode lama ini. Oleh karena itu, peneliti melakukan penelitian tersebut, dengan membuat suatu aplikasi absen. Hal ini didasari atas pemanfaatan teknologi yang tentunya akan memudahkan pihak akademik dalam mengolah data absen secara cepat dan akurat. Pada sistem tersebut juga dilengkapi dengan fitur *Global Positioning System* (GPS), untuk membantu mengetahui lokasi pengguna dalam melakukan absensi.
4. Penelitian Muhammad Yusuf, R. V. Hari Ginardi dan Adhatu Solichah A. berjudul “Rancang Bangun Aplikasi Absensi Perkuliahan Mahasiswa dengan Pengenalan Wajah”, yang terbit dalam Jurnal Teknik ITS, Vol. 5, No. 2, 2016, ISSN : 2337-3539. Permasalahan yang diangkat didalam penelitian tersebut adalah proses absen yang

dilakukan pada kampus terkait masih menggunakan metode absen secara manual. Hal ini tentunya kurang efektif, karena tentu akan menjadi pemicu terjadinya penitipan absen oleh mahasiswa. Selain itu, proses pengolahan data absensi akan memakan waktu karena masih menggunakan metode yang serba manual. Oleh karena itulah peneliti ingin membangun sebuah sistem absensi yang akan membantu proses absen serta memudahkan proses pengolahan data oleh pihak akademik. Teknologi yang tertanam pada aplikasi ini menggunakan sistem pengenalan wajah. Sistem ini menggunakan metode *Eigenface*.

5. Penelitian Subiantoro dan Sardiarinto (2018) berjudul “Perancangan Sistem Absensi Pegawai Berbasis Web Studi Kasus : Kantor Kecamatan Purwodadi”, yang terbit dalam Jurnal Swabumi, Vol. 6 No2, 2 September 2018, ISSN : 2355-990X / eISSN : 2549-5718. Permasalahan yang terdapat pada penelitian tersebut adalah masih digunakannya absensi secara manual pada kantor kecamatan Purwodadi. Absensi dilakukan dengan mengisi buku harian. Hal tersebut tentu akan berdampak pada efisiensi dan efektifitas pendataan, pencarian data, dan pengolahan data absen akan memakan waktu yang relatif lama. Disamping itu, kemungkinan terjadinya kesalahan data dan kehilangan data juga semakin besar. Dengan hadirnya sistem absensi ini, tentu akan memberikan kemudahan pada sisi pegawai dalam melakukan absensi, serta kemudahan pada sisi admin dalam mendapatkan laporan.

6. Hasil penelitian Al Husain, Abdul Haqy Aji Prastian, dan Andre Ramadhan (2017) berjudul “Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi”, yang terbit dalam Technomedia Journal (TMJ), Vol. 2 No. 1, Edisi Agustus 2017, eISSN : 2528-6544 / pISSN : 2620-3383. Menurut peneliti, suatu pembaruan terhadap suatu sistem baru sangat perlu dilakukan karena akan berdampak sangat baik bagi perusahaan. Jika memiliki sistem yang bagus, maka kinerja pada suatu perusahaan akan jauh lebih efisien, efektif, dan cepat. Salah satu sistem yang dapat diperbarui adalah sistem absensi. Sistem online digunakan untuk mempercepat absen para karyawan karena hanya dilakukan melalui smartphone masing-masing karyawan. Sistem absen yang dimaksud membutuhkan jaringan lokal yang berjarak hanya didalam lingkungan kerja, sehingga karyawan tidak dapat melakukan absensi pada area luar kantor untuk meminimalisir kecurangan.

Hasil penelitian Eni Dwi Wardihani, S.T., M.T. , Evanendra Nur Wisnu Putra, Nouri Aldila, dan Saroni Widodo, S.T., M.Kom. (2017) berjudul “*Design And Development Of Presence System System Using Rfid Based On Raspberry Pi*”, yang terbit dalam Journal of Applied Information and Communication Technologies, Vol. 2 No. 1, 2017. Permasalahan yang terdapat pada penelitian tersebut adalah sistem absen pada Politeknik Negri Semarang masih menggunakan cara yang

manual. Hal ini tidak akan menutup kemungkinan adanya kesalahan dalam pengolahan data absensi, serta rentannya terjadi kecurangan absen mahasiswa dalam memalsukan kehadirannya. Ditambah lagi, perkembangan sistem saat ini sudah sangat pesat sehingga semua dibuat serba otomatis. Berdasarkan hal itulah peneliti ingin mengembangkan sistem absen menggunakan Raspberry Pi berbasis RFID, dimana PBM yang nantinya akan mengakses data absensi mahasiswa. Database yang dipakai pada sistem tersebut menggunakan MYSQL. Admin PBM nantinya akan mengolah data absensi melalui *website* yang telah disediakan.

8. Hasil penelitian dari J. Chandramohan, R. Nagarajan, M. Ashok Kumar, T. Dineshkumar, G. Kannan, dan R. Prakash (2017) berjudul “*Attendance Monitoring System of Students Based on Biometric and GPS Tracking System*”, yang terbit dalam *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, Vol. 3, 2017, ISSN : 2454-1311. Penelitian tersebut membahas mengenai sistem biometrik yang dianggap sangat penting dan sangat bermanfaat dalam kehidupan sehari-hari, terutama ketika memanfaatkannya sebagai sistem absensi berbasis *fingerprint*. Selain aman, sistem biometrik dianggap susah untuk ditiru dan susah untuk dimanipulasi. Dalam penelitian tersebut, peneliti menggunakan sistem biometrik sidik jari dan sensor wajah dalam mengidentifikasi kehadiran siswa. Alat tersebut nantinya akan dihubungkan dengan GSM dengan bantuan

teknologi mikrokontroler, sehingga dapat terhubung dengan GPS. Apabila siswa melakukan absensi, peneliti menyebutkan bahwa hasil dari informasi kehadiran siswa dapat dikirim secara otomatis kepada orang tua siswa.

9. Hasil penelitian dari I Gede Tika Permana, Denny Sagita Rusdianto, dan Lutfi Fanani (2019) berjudul “Pengembangan Sistem Presensi Berbasis Lokasi Menggunakan *Geofence WiFi* dan *REST API* pada Fakultas Ilmu Komputer Universitas Bramawijaya”, yang terbit pada Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, No. 9, September 2019, e-ISSN : 2548-964X. Permasalahan yang diangkat pada penelitian tersebut adalah masih manualnya proses rekap absensi yang terjadi pada Universitas Bramawijaya. Penggunaan mesin *fingerprint* pada kampus tersebut menyebabkan proses rekapitulasi kehadiran masih manual. Selain itu, permasalahan lainnya adalah sering terjadi ketidaksesuaian data antara hasil absen dari kehadiran *staff* ataupun dosen. Dari permasalahan tersebutlah peneliti mencoba untuk mengembangkan suatu aplikasi yang dapat dipakai pada *smartphone* untuk melakukan absensi. Pada penelitian tersebut, peneliti memanfaatkan sensor *fingerprint*, kamera, dan *WiFi* dalam proses absensi.
10. Hasil penelitian dari Nandang Hermanto, Nurfaizah, dan Nur Rahmat Dwi Riyanto (2019) berjudul “Aplikasi Sistem Presensi Mahasiswa Berbasis Android”, yang terbit pada Jurnal SIMETRIS, Vol. 10, No. 1

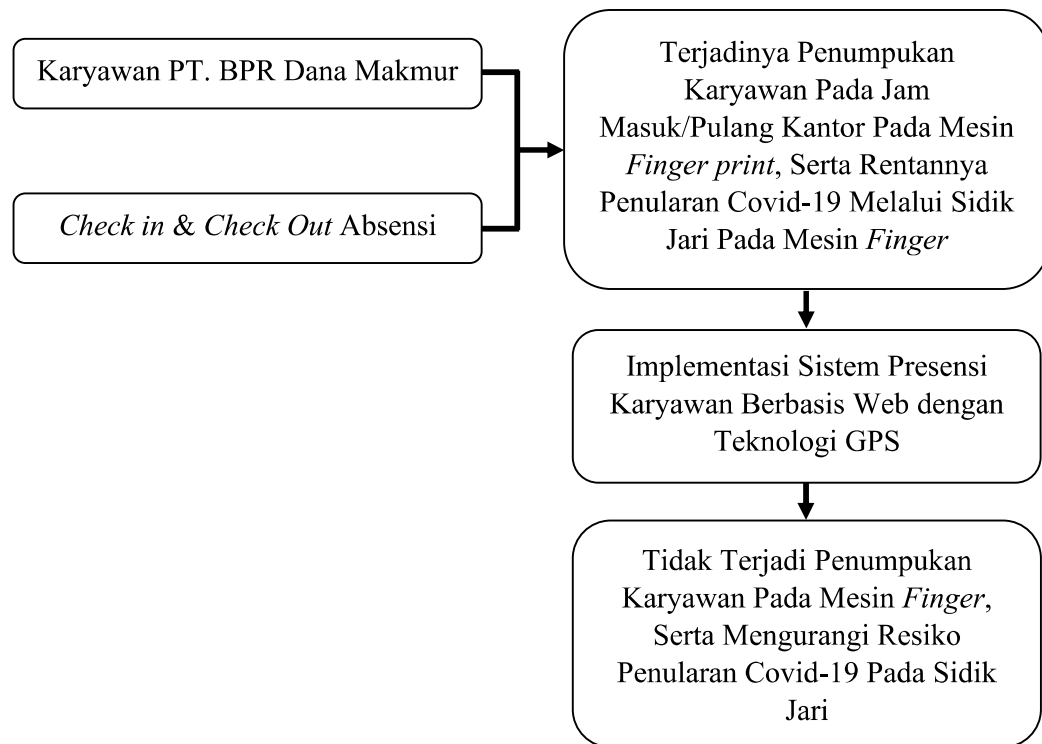
April 2019, P-ISSN : 2252-4983 / E-ISSN : 2549-3108. Penelitian tersebut membahas mengenai betapa pentingnya rekap hadir mahasiswa. Peneliti bertujuan untuk membuat suatu aplikasi absensi berbasis android. Hal ini didasari karena peneliti menganggap bahwa pemanfaatan teknologi saat ini sudah sangat pesat. Dalam perancangan sistem ini, peneliti menggunakan teknologi *QR Code (Quick Response Code)* yang nantinya akan digunakan dalam proses presensi. Dengan dikembangkannya sistem tersebut, peneliti berharap sistem tersebut dapat menjadi terobosan baru dalam metode absensi.

2.4 Kerangka Pemikiran

Pada umumnya, sistem absensi pada kantor menggunakan mesin *fingerprnt*. Namun, kelemahan dari mesin tersebut adalah tidak dapat melakukan absen untuk banyak orang pada waktu yang sama, sehingga karyawan terpaksa harus mengantri jika ingin melakukan absensi. Terlebih lagi, pada awal tahun 2020, dunia digemparkan dengan hadirnya *virus Corona (Covid-19)* yang di claim mampu ditularkan melalui sidik jari.

Oleh karena itu, dibutuhkanlah sebuah media absensi baru agar para karyawan PT BPR Dana Makmur dapat melakukan absensi secara bersamaan, dalam satu waktu secara *realtime*, dan menghindari penumpukan karyawan pada area mesin *finger* dalam rangka mendukung *social distance*, serta mengurangi resiko tertularnya *Covid-19* melalui sidik jari yang terdapat pada mesin *finger*. Dengan adanya sistem presensi dengan teknologi GPS berbasis web, absensi pun

dapat dilakukan secara mudah dan *realtime*, tanpa melalui mesin *finger*. Berdasarkan uraian diatas, maka kerangka berpikir dalam penelitian ini dapat digambarkan sebagai berikut :



Gambar 2.9 Kerangka Pemikiran