



BAB II

TINJAUAN PUSTAKA

2.1. Teori Judul

2.1.1 Absensi

Absensi menurut Nugroho dalam Santoso dan Yulianto (2017:67) Absensi adalah sebuah pembuatan data untuk daftar kehadiran yang biasa digunakan bagi sebuah lembaga atau instansi yang sangat perlu membutuhkan sistem seperti ini. Absensi menuaikan sebuah sistem yang harus dipergunakan sebagai konsep sistem absensi, disaat sistem membutuhkan sebuah data maka sistem akan dijadikan sebagai aplikasi yang sanggup menjalankan dan membuat data absensi tersebut.

2.1.2 Android

Menurut Hermawan (2011 : 1), Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *I-Phone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

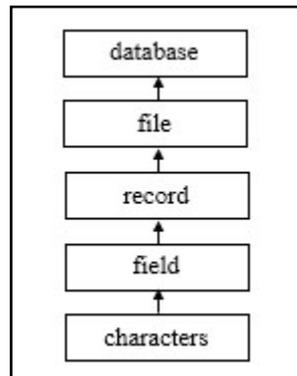
2.2. Teori Khusus

2.2.1. Data Base

Pengertian *database* menurut Bambang Hariyanto dalam jurnal Minarni dan Susanti (2014:105) adalah :”kumpulan data (elementer) yang secara logic berkaitan dalam mempresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam system tertentu”. Dari definisi diatas maka dapat disimpulkan bahwa database adalah kumpulan dari suatu data



yang saling berhubungan satu dengan yang lainnya, yang berdasarkan sebuah skema atau struktur tertentu, yang kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Suatu database terdiri dari *file*, *record*, *field data*, dan *characters*.



Gambar 2.1. Jenjang Data

a. *Characters*

Merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakterkarakter khusus (*special characters*) yang membentuk suatu item data/*field*.

b. *Field*

Memrepresentasikan suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.

- *field name* : harus diberi nama untuk membedakan *field* yang satu dengan lainnya.
- *field representation* : tipe *field* (karakter, teks, tanggal, angka, dsb), lebar *field* (ruang maksimum yang dapat diisi dengan karakterkarakter data).
- *field value* : isi dari *field* untuk masing-masing record.

c. *Record*

Kumpulan dari *field* membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu *file*. Misalnya *file* personalia, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.



d. *File*

File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* mata pelajaran berisi data tentang semua mata pelajaran yang ada.

e. *Database*

Kumpulan dari *file* / *table* membentuk suatu *database*.

2.2.2. Android Package (APK)

Menurut jurnal Harison, Busran, Putra (2016 : 198) *Android Package* umumnya digunakan menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat Android. APK pada dasarnya seperti *zip file*, karena berisi dari kumpulan *file*, dapat diperoleh melalui berbagai metode, seperti menginstal sebuah aplikasi melalui *Market*, *download* dari sebuah situs web, atau membuat sendiri dengan bahasa Java

2.2.3. Data Flow Diagram

Sukamto et. al (2014:70), "*Data Flow Diagram (DFD)* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

a. Proses (*Process*)

Aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

b. Aliran Data (*Data Flow*)

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

c. Penyimpanan Data (*Data Store*)

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam *data store*. Aliran data di-*update* atau ditambahkan ke *data store*.

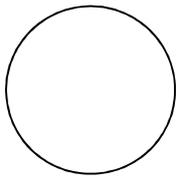
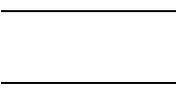


d. Entitas (*Entity*)

Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.

Notasi yang digunakan dalam membuat DFD ada empat buah, yaitu sebagai berikut:

Tabel 2.1. Simbol-simbol Data Flow Diagram

No.	Notasi	Keterangan
1		<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan:</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>).</p>

Lanjutan **Tabel 2.1.** Simbol-simbol Data Flow Diagram

		<p>Catatan:</p> <p>Nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda</p>
3		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan:</p> <p>Nama yang diberikan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
4		<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>Catatan:</p> <p>Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>

(Sumber: Sukamto et. al, 2014:71)



Menurut Sukanto et. al (2014:72), berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

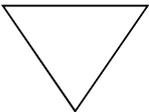
DFD Level 3, 4, 5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.2.4. Block Chart

Kristanto (2011:68) menjelaskan bahwa *block chart* berfungsi untuk memodelkan masukan, keluaran, proses, dan transaksi dengan menggunakan simbol-simbol tertentu yang pembuatannya harus memudahkan bagi pemakai dalam memahami alur dari sistem.

Adapun simbol-simbol yang sering digunakan dalam *block chart* adalah sebagai berikut:

**Tabel 2.2.** Simbol-simbol Block Chart

No.	Simbol	Arti
1		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2		Proses yang dilakukan oleh komputer.
3		Menandakan dokumen yang diarsipkan (arsip manual).
4		Data penyimpanan (data storage).

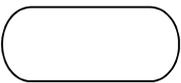
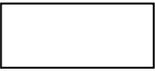
(Sumber: Kristanto, 2011:68)

2.2.5. Flowchart

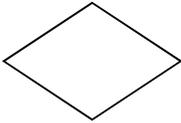
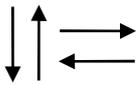
Sitorus (2015:14), "Flowchart (diagram alir) menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga flowchart merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu."

Simbol-simbol yang digunakan untuk menggambarkan flowchart dapat dilihat pada tabel berikut:

Tabel 2.3. Simbol-simbol Flowchart

No.	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2		<i>Input/ Output</i>	Meyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
3		<i>Process</i>	Menyatakan suatu tindakan (proses yang dilakukan oleh komputer).

Lanjutan **Tabel 2.3.** Simbol-simbol Flowchart

4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
5		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer)
6		<i>Flow</i>	Menyatakan jalannya arus suatu proses.

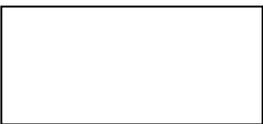
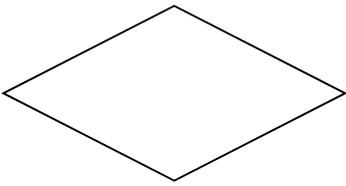
(Sumber: Sitorus, 2015:15)

2.2.6. Entity Relationship Diagram (ERD)

Sukanto et. al (2014:50), “ERD adalah permodelan awal basis data relasional yang dikembangkan berdasarkan teori himpunan dalam”.

Simbol-simbol yang digunakan untuk menggambarkan ERD dapat dilihat pada tabel berikut ini:

Tabel 2.4. Simbol-simbol Entity Relationship Diagram (ERD)

No.	Simbol/Gambar	Keterangan
1	<i>Entitas</i> 	Orang, tempat, atau benda; memiliki nama tunggal; ditulis dengan huruf besar; dan berisi lebih dari 1 <i>instance</i> .
2	<i>Attribute</i> 	Properti dari entitas, harus digunakan oleh minimal 1 proses bisnis, dan dipecah dalam <i>detail</i> .
3	<i>Relationship</i> 	Menunjukkan hubungan antar 2 entitas, dideskripsikan dengan kata kerja, memiliki modalitas (<i>null/not null</i>) memiliki kardinalitas (1:1, 1:N, atau M:N)

(Sumber: Fatta, 2007:124)



2.2.7. Kamus Data (KD) atau Data Dictionary (DD)

Sukamto et. al (2014:73), “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).”

Simbol-simbol yang digunakan dalam kamus data, yaitu:

Tabel 2.5. Simbol-simbol dalam Kamus Data

No.	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	Dan
3	[]	baik...atau...
4	{ } ⁿ	n kali diulang atau bernilai banyak
5	()	data opsional
6	*...*	batas komentar

(Sumber: Sukamto et. al, 2014:74)

2.3 Teori Program

2.3.1. HTML

Suryana et. al (2014:29), “*HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman web.”

Sibero (2013:19), “*HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk pertukaran dokumen web.”

Dari pengertian diatas, dapat disimpulkan bahwa HTML (*HyperText Markup Language*) merupakan bahasa yang sering digunakan untuk membuat halaman web.

Sibero (2013:19) menjelaskan bahwa struktur dokumen HTML sebagai berikut:



```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

2.3.2 PHP (PHP Hypertext Preprocessor)



Gambar 2.2. Logo PHP

Sibero (2013:49), “PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*.”

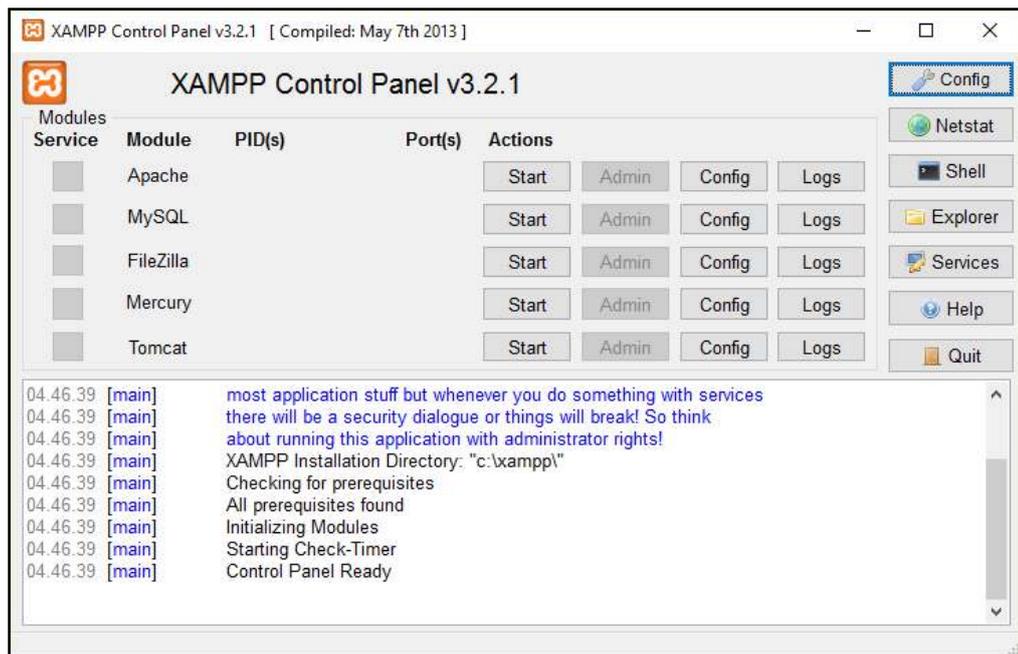
Wahana Komputer (2014:33), “PHP merupakan bahasa berbentuk *script* yang ditempatkan di dalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirimkan kepada *web browser* klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis.”

Dari kedua pengertian di atas, dapat disimpulkan bahwa PHP adalah bahasa pemrograman web yang mampu menerjemahkan kode dan memproses sehingga menghasilkan tampilan website.



2.3.3 Pengertian XAMPP

Wahana Komputer (2014:72), “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket.”



Gambar 2.3. Tampilan XAMPP Control Panel

2.3.4 MySQL

Sibero (2013:97) menjelaskan bahwa MySQL atau dibaca “*My Sekuel*” adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. Berikut merupakan tabel beberapa pernyataan yang ada pada MySQL:

Tabel 2.6. Pernyataan MySQL

No.	Pernyataan	Kegunaan	Contoh
1	INSERT	Untuk memasukkan data ke dalam tabel.	INSERT INTO nama_tabel (kolom_1, kolom_2, ... kolom_n)

Lanjutan **Tabel 2.6.** Pernyataan MySQL

			VALUES ('nilai_kolom_1', 'nilai_kolom_2', ..., 'nilai_kolom_n');
2	SELECT	Untuk melihat isi tabel.	SELECT nama_kolom FROM nama_tabel;
3	UPDATE	Untuk mengubah data dalam tabel.	UPDATE nama_tabel SET kolom_1 = nilai_baru_1, kolom_2 = nilai_baru_2, ... kolom_n = nilai_baru_n WHERE kondisi;
4	DELETE	Untuk menghapus data dalam tabel.	DELETE FROM nama_tabel WHERE kondisi;

Sibero (2013:71) menjelaskan fungsi-fungsi PHP MySQL adalah sebagai berikut:

Tabel 2.7. Fungsi-fungsi PHP MySQL

No.	Fungsi MySQL	Penggunaan
1	<i>mysql_connect</i>	Membuka koneksi <i>database</i> .
2	<i>mysql_error</i>	Mengambil pesan kesalahan.
3	<i>mysql_fetch_array</i>	Mengambil hasil record dan mendefinisikan dalam bentuk array.
4	<i>mysql_num_rows</i>	Mengambil jumlah baris data hasil query.
5	<i>mysql_query</i>	Mengirimkan perintah query pada server.
6	<i>mysql_select_db</i>	Memilih dan mengaktifkan <i>database</i> .



2.3.5 Sublime Text



Gambar 2.4. Logo Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages. Sublime Text bukanlah aplikasi open source dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

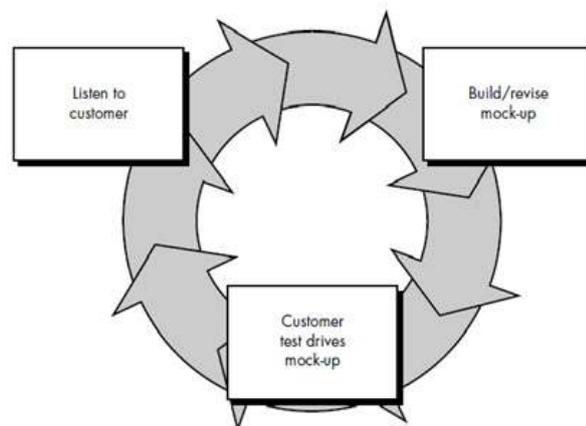
Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile dan XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user.



2.4 Metode Prototipe

Rosa & Shalahuddin (2013:31) menjelaskan metode model prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.

Metode model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*.



Gambar 2.5. Ilustrasi Model Prototipe

Mock-Up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak.

Seiring dengan mengembangkan prototipe maka sistem perangkat lunak yang sebenarnya dikembangkan juga sehingga sesuai dengan kebutuhan pelanggan



(*customer*) atau user. Metode model prototipe kurang cocok untuk aplikasi dengan skala besar karena membuat prototipe untuk aplikasi skala besar akan sangat memakan waktu dan tenaga.

2.5 Penelitian Terdahulu

Menurut penelitian Ihsan Verdian pada tahun 2015 dalam jurnal berjudul Aplikasi Sistem Absensi Mahasiswa Menggunakan Sidik Jari Pada Universitas Putra Indonesia “YPTK” Padang dengan Metode *System Development Life Cycle* (SDLC) yang bertujuan untuk mengurangi bahkan menghilangkan adanya manipulasi data kehadiran oleh mahasiswa. Hal ini dikarenakan sidik jari merupakan salah satu bagian unik dari manusia, sidik jari mahasiswa yang satu akan berbeda dengan sidik jari mahasiswa yang lain. Aplikasi pencatatan kehadiran mahasiswa menggunakan sidik jari dapat meningkatkan keakuratan data kehadiran karena sidik jari mahasiswa tidak mungkin dititipkan pada mahasiswa lain.

Menurut jurnal penelitian Heroe Santoso dan Ahmad Wilda Yulianto 2017 yang berjudul Analisa dan Perancangan Sistem Absensi Siswa Berbasis Web dan SMS *Gateway* dengan menggunakan Metode Prototipe bertujuan memberikan kontribusi dalam dunia pendidikan berupa mengelola absensi siswa secara elektronik, dalam upaya peningkatan efisiensi dan efektivitas pengolahan data serta meminimalkan terjadi kesalahan yang dilakukan oleh pengguna sistem.

Menurut penelitian Warkim, Hafiz Novanda Ichwan dan Husnul Kamar Z 2015 dengan judul jurnal Analisa dan Desain Sistem Kehadiran Pegawai Pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia dengan Metode *System Development Life Cycle* (SDLC) bertujuan untuk memberikan keuntungan dan manfaat bagi para pemangku kepentingan yaitu karyawan, staf dan Manajemen Sub-bagian. karyawan PAPPIPTEK LIPI dapat mengakses sistem informasi kehadiran, dan aplikasi resmi untuk permohonan izin secara online, personil sub bagian dapat melakukan perekapan kehadiran karyawan dan keterlambatan untuk perhitungan manfaat kinerja dan uang makan dengan mudah dan efisien. Serta Manajemen meng-approve karyawan yang



meminta izin, serta pemantauan kehadiran karyawan. Ini semua dapat dilakukan secara online.

Menurut Jurnal Al Husain, Abdul Haqy Aji Prastian dan Andre Ramadhan 2017 dengan jurnal berjudul Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi dengan metode Prototipe bertujuan untuk mempercepat absen hanya melalui smartphone android yang dimiliki oleh masing-masing karyawan. Hal ini membutuhkan jaringan lokal yang berjarak hanya didalam lingkungan perusahaan sehingga karyawan tidak dapat absen diluar perusahaan.

Menurut jurnal Abdi Darmawan, dkk. 2016 dengan jurnal berjudul Sistem Absensi dan Pelaporan Berbasis *Fingerprint* dan SMS Gateway dengan menggunakan metode *Waterfall* dengan perancangan sistem absensi dibagi menjadi dua environment, pertama absensi berbasis *Fingerprint* dapat memberikan kemudahan bagi pihak sekolah untuk data kehadiran siswa. Sedangkan yang kedua dengan menggunakan SMS gateway sebagai media informasi, yang dapat memberikan laporan dari pihak sekolah kepada orang tua siswa atas kehadiran dan tidak kehadiran siswa. Dengan perancangan sistem absensi ini diharapkan dapat membantu pihak sekolah dan orang tua dalam anak mereka.