BAB II
TINJAUAN PUSTAKA

1.1 Penelitian Terdahulu

Penelitian terdahulu ini akan menjadi acuan penulis dalam membuat laporan akhir yang berguna untuk memperkaya teori yang digunakan dalam menkaji penelitian yang dilakukan. Berikut adalah penelitian terdahulu yang berasal dari beberapa jurnal yang terkait dengan judul laporan akhir penulis:

1. Sistem Informasi Absensi Foto *Webcam* Menggunakan Metode Togaf Pada SMK Media Insani Cendekia Cikarang

Berdasarkan penelitian yang dilakukan oleh (Suherman, Rizal Ananda & Irfan Afriantoro, 2022) telah dirancang suatu sistem informasi absensi foto *webcam* menggunakan metode Togaf. Pada aktifitas kegiatan absensi di SMK Media Insani Cendekia (MIC) Cikarang saat ini masih dilakukan dengan cara manual, yaitu absensi siswa dilakukan dengan dibuatkan formulir absensinya dan dilakukan dengan manual oleh guru mata pelajaran terkait. Kerugian yang mungkin muncul pada sistem absensi manual adalah rekapitulasi data yang masih memakan banyak waktu dan tenaga. Metode penelitian yang digunakan adalah metode The *Open Group Architecture Framework* (TOGAF). Metoda ini hanya terfokus pada fase F (*Migration Plaining*) dan fase G (*Implementation Governance*). Hasil penelitian ini berupa aplikasi yang memanfaatkan kamera HP sebagai komponen utama dalam membantu proses absensi setiap guru, staff dan siswa.

2. Rancang Bangun Aplikasi Sistem Informasi Absensi Karyawan Online

Berdasarkan penelitian yang dilakukan oleh (Rully Roosdianto, Ani Oktarini Sari, Arief Satriansyah, 2021) sistem yang berjalan dalam pencatatan data absensi pada CV Cahaya Toner masih manual, yaitu dengan cara melihat satu persatu data absensi karyawan, dicatat di buku absensi dan membubuhkan tanda tangan. Hal ini tidak efisien dan membutuhkan waktu yang lama, sehingga berpengaruh terhadap proses perhitungan penggajian karena jam masuk dan jam keluar kerja karyawan sulit untuk dikontrol. Metode penelitian yang digunakan pada aplikasi ini adalah pengembangan dengan model *waterfall*. Diharapkan dengan adanya aplikasi

absensi karyawan berbasis *web* ini dapat memberikan kemudahan dalam proses absensi, pencarian data dan perhitungan rekap absensi, serta mengurangi resiko kehilangan dan kesalahan pencatatan data absensi pada CV Cahaya Toner.

3. Aplikasi Absensi Dengan Metode Geolocation Berbasis Web (Studi Kasus: PT. Codepolitan Integrasi Indonesia)

Berdasarkan penelitian yang dilakukan oleh (Shandy Tresnawati & Alfian Pratama, 2021) saat ini di PT. Codepolitan Integrasi Indonesia masih melakukan pencatatan manual terhadap sistem absensinya, selain itu terdapat berbagai masalah yang terjadi seperti sulitnya melakukan laporan secara *realtime* kepada pimpinan, sulitnya *tracking* data absen/izin/cuti/sakit berdasarkan karyawan secara cepat, dan resiko kesalahan *input* data absensi, hal tersebut menyebabkan ketidakefektifan dalam melakukan absensi untuk pengelolaan data karyawan. Maka dari itu dibutuhkan teknologi bernama *geolocation*, oleh karena itu penulis membuat suatu aplikasi berbasis *web* dengan memanfaatkan *geolocation* untuk membantu proses absensi karyawan di PT. Codepolitan Integrasi Indonesia. Penulis membuat aplikasi absensi *online* dengan metode *geolocation*. Sistem ini terbukti dapat diterima dengan baik oleh karyawan PT. Codepolitan Integrasi Indonesia.

4. Pengembangan Sistem Informasi Absensi Kepegawaian Menggunakan Geolocation Di SMP Negeri 1 Bongomeme

Berdasarkan penelitian yang dilakukan oleh (Muhammad Yunan Tahir, Lillyan Hadjaratie & Moh.Ramdhan Arif Kaluku, 2022) dengan hasil observasi, ditemukan bahwa sistem absensi pegawai serta guru di sekolah ini masih belum mengikuti perkembangan yang ada, sehingga membuat sekolah ini masih agak tertinggal dibandingkan sekolah-sekolah lain. Oleh sebab itu maka dibutuhkan sebuah sistem informasi absensi yang bisa diakses lewat sebuah *web* tanpa harus melakukan absensi manual. *Geolocation* atau *Global Positioning System* (GPS) sendiri merupakan singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Berdasarkan beberapa masalah yang telah dijabarkan, maka penulis menjadikannya landasan untuk memberi solusi yang diharapkan mampu

menyelesaikan masalah. Dengan membuat sistem informasi yang terintegrasikan dengan *geolocation* guna mengetahui lokasi.

5. Sistem Informasi Kompensasi Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus : STMIK AKAKOM YOGYAKARTA)

Berdasarkan penelitian yang dilakukan oleh (Andi Purbo Setiawan, 2020) STMIK AKAKOM menetapkan kompensasi sebagai pengganti absensi kehadiran mahasiswa dalam perkuliahan. Kompensasi dilakukan jika mahasiswa tidak menghadiri perkuliahan dengan status alpha. Pada saat ini STMIK AKAKOM dalam pelaksanaan kompensasi masih dilakukan secara manual, sehingga aktivitas yang dilakukan membutuhkan waktu yang lama. Sehingga STMIK AKAKOM harus menggunakan sistem yang berbeda yaitu kompensasi secara *online* dengan metode observasi. Dengan adanya sistem kompensasi *online* ini, mahasiswa dapat mengurangi waktu yang dibutuhkan.

2.1 Absensi

Ramadhan, W & Surya Hendra P (2022:527) menyatakan bahwa "Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari aktifitas pelaporan yang ada di dalam sebuah institusi yang disusun dan diatur sehingga mudah untuk dicari dan dipergunakan ketika diperlukan oleh pihak yang berkepentingan". Dari hal tersebut absensi berisikan dokumen pencatatan kehadiran dan ketidakhadiran seseorang di suatu tempat seperti perusahaan atau instansi pendidikan yang juga digunakan untuk mengawasi produktivitas tiap-tiap kegiatan orang tersebut.

2.2 Validasi

Purnomo (2018) menyatakan bahwa "Validasi adalah suatu pengukuran atau tes untuk mengetahui ketepatan dan keakuratan dari suatu alat ukur atau suatu pengukuran". Validasi juga dapat didefinisikan sebagai tindakan pembuktian suatu kegiatan untuk mengukur sejauh mana seseorang dapat mencerminkan karakteristik antar orang sehingga dapat diukur. Pembuktian ini mengenai kebenaran data yang diteliti, apakah data tersebut sesuatu yang valid terhadap suatu *instrument*, yang

mana *instrument* ini berisikan kumpulan data untuk mengukur objek penelitian sehingga didapatkan data agar dapat di validasi.

2.3 Kompensasi

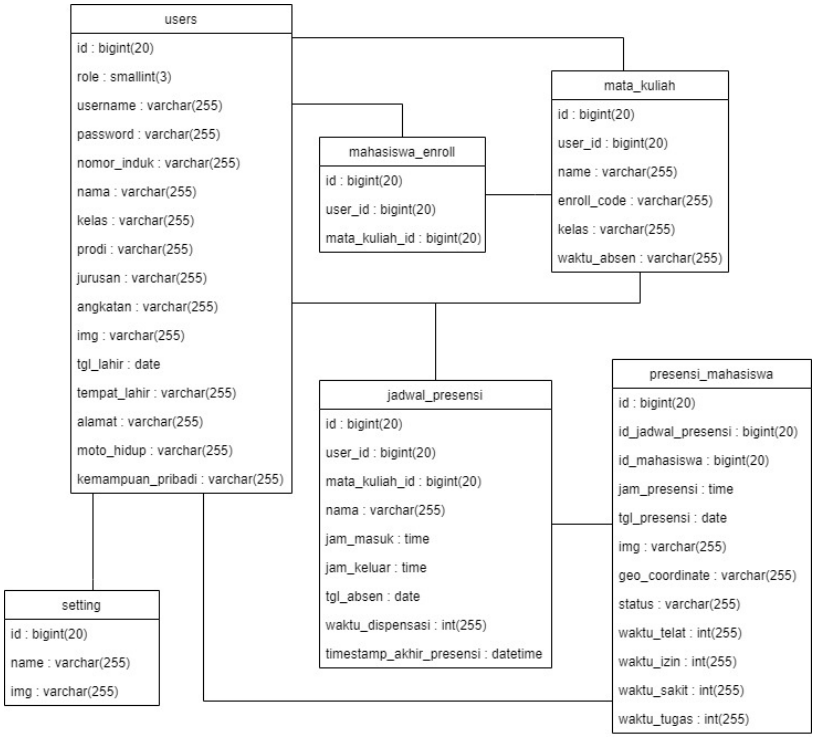
Aqsa (2020:371) menyatakan bahwa "Kompensasi merupakan segala sesuatu yang diterima dapat berupa fisik maupun *non* fisik dan harus dihitung dan diberikan kepada seseorang yang umumnya merupakan obyek yang dikecualikan dari pajak pendapatan". Kompensasi juga dapat berupa suatu balas jasa yang diberikan oleh suatu kelompok, organisasi atau instansi kepada para tenaga kerja yang disebabkan tenaga kerja tersebut telah memberikan suatu tenaga ataupun pikiran demi kemajuan organisasi yang berguna untuk mencapai tujuan yang ditetapkan, dan dari hal tersebut dapat diketahui bahwa kompensasi juga dipegaruhi oleh kinerja seseorang dalam melakukan suatu kegiatan.

2.4 Website

Jimi Asmara (2019:3) menyatakan bahwa "*Website* adalah keseluruhan halaman-halaman *web* yang terdapat dalam sebuah *domain* yang mengandung informasi". Informasi tersebut dibangun atas banyak halaman *web* yang saling terhubung yang dapat diakses dengan mudah melalui *internet*. *Website* sendiri terdiri dari teks, gambar, suara, animasi sehingga menjadi media informasi menarik untuk dikunjungi oleh orang lain.

2.5 Database

Menurut Andaru (2018) "*Database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi". *Database* atau yang juga dikenal dengan basis data yang terdiri atas 2 kata yaitu "Basis" dan "Data". Untuk istilah dari "Basis" dapat diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan "Data" adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, barang yang nantinya direkam dalam bentuk angka, huruf, simbol, teks, bunyi, gambar, dan beberapa kombinasi lainnya.



Gambar 2.1 Relasi Antar Tabel

2.6 HTML

Adnyana (2019) menyatakan bahwa "HTML adalah singkatan dari *Hyper Text Markup Language* yang merupakan bahasa pemrograman dasar dalam pemembuat *website*, HTML terdiri dari *head, body* dan di dalamnya terdapat *TAG* dan *attibute,* walaupun dikatakan sebagai bahasa pemrograman, tetapi HTML belum dapat dikatakan sebagai bahasa pemrograman karena HTML tidak miliki logika, HTML hanya memberikan *output*, maka dari itu HTML diibaratkan sebagai pondasi atau struktur dari *web* dan yang menjadi bahasa pemrograman nya yaitu PHP dan Javascript". Dari hal tersebut HTML bersifat statis yang berarti tidak dapat berinteraksi aktif dengan *user*, sehingga HTML dapat dikombinasikan dengan bahasa pemrograman *web* lainnya.

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

Gambar 2.2 Syntax dasar HTML

2.7 PHP

Endra dkk., (2022:48) menyatakan bahwa "*Website* dapat dikembangkan dengan bahasa pemrograman dinamis, salah satunya adalah bahasa pemrograman PHP (*Hypertext Preprocessor*) yang merupakan bahasa pemrograman *open-source server side*". PHP adalah bahasa pemrograman yang digunakan untuk mengembangkan situs *web* statis, dinamis, ataupun aplikasi *web.* PHP merupakan bahasa pemrograman yang bersifat *scripting server-side* yaitu pengolahan data yang dilakukan oleh pengunjung atau penggunanya sehingga baik pengunjung maupun penggunanya dapat saling berinteraksi. PHP juga merupakan bahasa pemrograman yang bersifat *open source* dimana pengguna dapat dengan bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan.

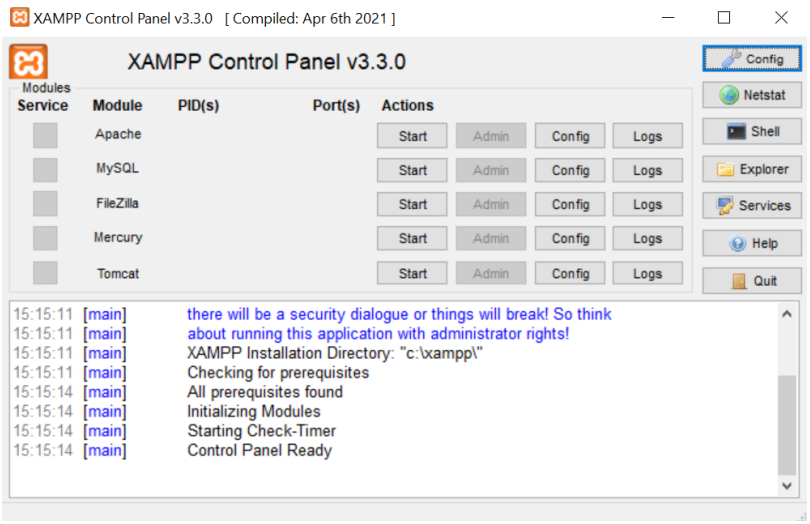
```
6  <body>
7      <?php
8          // Tulis Kode PHP
9      ?>
10 </body>
```

Gambar 2.3 *Syntax* Bahasa Pemrograman PHP

2.8 XAMPP

Menurut Iqbal (2019:15) "XAMPP merupakan sebuah *software web server* Apache yang di dalamnya sudah tersedia *database server* MySQL dan *support* PHP *programming*". XAMPP ini juga dikenal sebagai *Cpanel server* yang mampu melakukan *preview website* tanpa harus tersambung dengan jaringan *internet* ataupun *online*. Fungsi XAMPP sendiri adalah sebuah *server* yang berdiri sendiri (*localhost*) yang berisi oleh beberapa perangkat lunak yaitu Apache, MySQL, PHP, dan Perl. XAMPP sendiri merupakan sebuah singkatan dari:

1. X, yaitu *crossplatform* karena XAMPP dapat dijalankan di *Windows*, *Linux, Mac* dsb.
2. A, yaitu Apache yang berarti sebagai *web server.*
3. M, yaitu MySQL sebagai *Database Management System* (DBMS).
4. PP, merupakan singkatan dari PHP dan Perl sebagai bahasa pemrograman yang didukung.



Gambar 2.4 XAMPP *Control Panel*

2.9 Apache

Syafitri (2019) menyatakan bahwa *"*Apache merupakan perangkat lunak atau *software web server* yang bersifat *open source*, itu berarti Apache didukung oleh para *developer* di seluruh dunia. Apache menjadi lebih terawat dan terus diperbarui secara teratur dengan *fitur* dan fungsionalitas baru untuk meningkatkan kualitas dalam pengiriman layanan HTTP". Apache juga mudah dikelola karena memiliki struktur berbasis modul. Modul memungkinkan *administrator server* untuk mengaktifkan dan menonaktifkan fungsi atau fitur tambahan.

2.10 MySQL

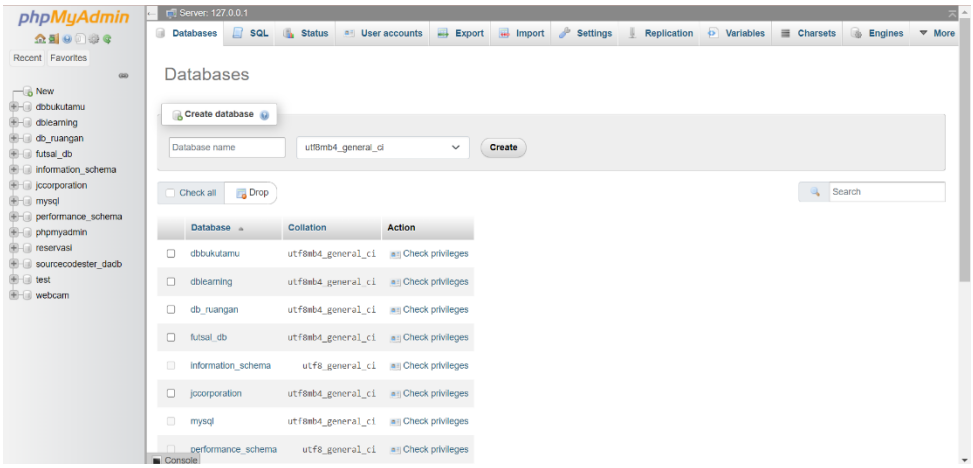
My Structure Query Language atau MySQL adalah *database management system* yang menggunakan bahasa *Structured Query Language* (SQL) sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server.* Kegunaan dari MySQL adalah sebagai pengumpulan data dari berbagai sumber, untuk *e-commerce*, maupun aplikasi *logging*. Selain itu juga MySQL merupakan sebuah *software database*, yang memiliki tipe data relasional, artinya MySQL menyimpan data dalam berbagai bentuk tabel-tabel yang saling berhubungan (Sitohang, 2018:8).

```
<!-- Format Select -->
SELECT * FROM 'nama_tabel' WHERE id
<!-- Format Insert -->
INSERT INTO 'nama_tabel' (isi field)
        VALUES (variabel field)
<!-- Format Update -->
UPDATE (nama tabel)
        SET
            username='$username',
        WHERE id = '$id'
<!-- Format Detele -->
DELETE FROM 'nama tabel yang ingin dihapus' WHERE id = '$id'(kondisi)
```

Gambar 2.5 Format Dasar Pemrograman MySql

2.11 PHPMyAdmin

PHPMyAdmin adalah sebuah program berbasis *web open source* yang berfungsi untuk mengelola *database* MySQL atau yang juga biasa disebut *tool database* yang juga digunakan untuk *administrator* dari *server* MySQL. Dengan adanya *software* berbasis *web* ini tentunya akan memudahkan dalam melakukan manipulasi *database* MySQL tanpa harus mengetikkan perintah pada *command line.* Tidak hanya itu *fitur* yang ditampilkan juga lengkap dan mudah dipahami bagi pengguna. PhpMyAdmin mendukung berbagai variasi MySQL, di antaranya adalah dalam mengelola *database*, *table*, *fields*, relasi, *indeks*, pengguna, perijinan, dan lain-lain (Rasebn, Setemen & Wayan, 2019).



Gambar 2.6 Tampilan dashboard PHPMyAdmin

2.12 Sublime Text

Supono (2018:14) menyatakan bahwa "*Sublime Text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau mengedit suatu aplikasi. *Sublime Text* memiliki *plugin* tambahan yang memudahkan *programmer*. Selain

itu *Sublime Text* juga memiliki desain yang *simple* dan keren sehingga terlihat elegan untuk sebuah *syntax editor*". Dari hal tersebut juga diketahui bahwa *Sublime Text* adalah salah satu *text-editor* yang dapat digunakan oleh berbagai macam *platform* OS (*Operating System*). *Sublime Text* mendukung dalam 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, ASP, HTML, dan lain sebagainya.

Gambar 2.7 Logo *Text-Editor Sublime Text*

2.13 CSS

CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang berfungsi untuk mengatur gaya tampilan/*layout* halaman *web* agar dapat ditampilkan lebih menarik. CSS digunakan bersama bahasa *markup* seperti HTML dalam membangun sebuah *website* dan tentunya CSS berfungsi untuk memisahkan konten dari tampilan visualnya di *situs* agar *website* yang dijalankan lebih menarik dan memiliki fungsi yang berjalan dengan baik. CSS dapat merubah *text,* warna, *background* dan posisi dari suatu *tag* (Marlina, 2021).

Gambar 2.8 Logo CSS

2.14 JavaScript

Abdulloh (2018:193) menyatakan bahwa "JavaScript merupakan bahasa pemrograman *web* yang pemrosesnya dilakukan di sisi *client*". Itu berarti JavaScript juga akan membuat program agar dokumen yang ditampilkan dalam *web browser* menjadi lebih interaktif. JavaScript yang digunakan pada *website* juga bertujuan agar *website* dapat berjalan secara dinamis, seperti dapat menampilkan dan menghilangkan objek-objek pada *website* yang kemudian nantinya juga dapat memanggil kembali objek yang dihilangkan tersebut. JavaScript juga disebut sebagai bahasa *script* karena program yang dijalankan JavaScript dapat bekerja langsung tanpa perlu di*compile*.

Gambar 2.9 Logo JavaScript

2.15 JQuery

Abdulloh (2018:233) menyatakan bahwa "JQuery merupakan salah satu dari sekian banyak JavaScript *library*, yaitu kumpulan fungsi JavaScript yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode JavaScript". Fungsi JavaScript yang sudah dibentuk sebagai objek dan kode siap pakai tentunya dibuat untuk memudahkan pembuatan *website* dengan HTML yang berjalan di sisi *client*. JQuery dapat menyederhanakan kode JavaScript dengan cara memanggil fungsi yang telah disediakan. JQuery memiliki ciri khas pada penggunaan perintahnya, awalan untuk jQuery dengan tanda $ kemudian dilanjutkan dengan fungsi atau perintah.

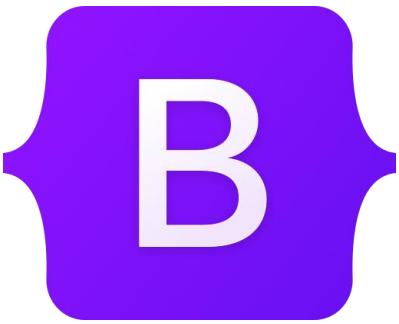
2.16 Framework

Menurut Setyawan, M.Yusril Helmi, dan Dinda Ayu dalam buku yang berjudul "*Membuat Sistem Informasi Gadai Online Menggunakan Codeigniter*

Serta Kelola Proses Pemberitahuannya" (2019:56) menyatakan bahwa "*Framework* merupakan kumpulan fungsi-fungsi dasar atau perintah yang biasa digunakan dalam mengembangkan suatu *software*, dengan harapan agar *software* yang dibangun menjadi lebih cepat dan terstruktur." Dengan adanya *framework* dapat mempermudah dalam membuat aplikasi yang akan dibuat. Saat menggunakan *framework* kita dapat fokus kepada proses yang dihadapi tanpa harus berfikir banyak mengenai masalah struktur aplikasi, standar *coding*,dll.

2.17 Bootstrap

Iin Mulyadin (2018) berpendapat bahwa "*Bootstrap* adalah sebuah alat bantu untuk membuat sebuah tampilan *website* yang dapat mempercepat pekerjaan seorang pengembang *website* atau pendesain halaman *website*". *Bootstrap* dapat membantu aplikasi menjadi *responsive*, yang berarti tampilan *web* yang dibuat oleh *bootstrap* akan menyesuaikan ukuran *layer* dan *browser* yang kita gunakan baik di *desktop, tablet* ataupun *mobile device. Bootstrap* biasanya digunakan dalam pemrograman berbasis HTML dan CSS serta sudah tersedia dalam bentuk *template* sehingga akan mempermudah dalam menjalankan kode program.



Gambar 2.10 Logo *Bootstrap*

2.18 Webcam.js

Istilah *webcam* merupakan suatu kombinasi dari kata "*web*" dan "*cam*" atau kamera video yang bertujuan untuk menyiarkan suatu video di *website*. *Webcam* pada dasarnya adalah suatu alat berupa perangkat keras yang memiliki bentuk kamera yang sering terpasang di komputer atau laptop, namun lain halnya dengan *webcam.js*. *Webcam.js* merupakan sebuah *software library* JavaScript yang menyediakan fungsi untuk mengambil gambar dari *webcam*. *Webcam.js* adalah API berbasis *browser* yang memiliki arti bahwa seorang pengembang dapat

mengintegrasikan *library* dengan halaman *web* dan dapat juga menggunakannya dalam aplikasi *webnya.* Dengan *webcam* kita dapat mengirimkan gambar-gambar secara *live* dari manapun ia berada ke seluruh penjuru dunia dengan bantuan *internet* (Hermawan, 2019).

2.19 Geolocation

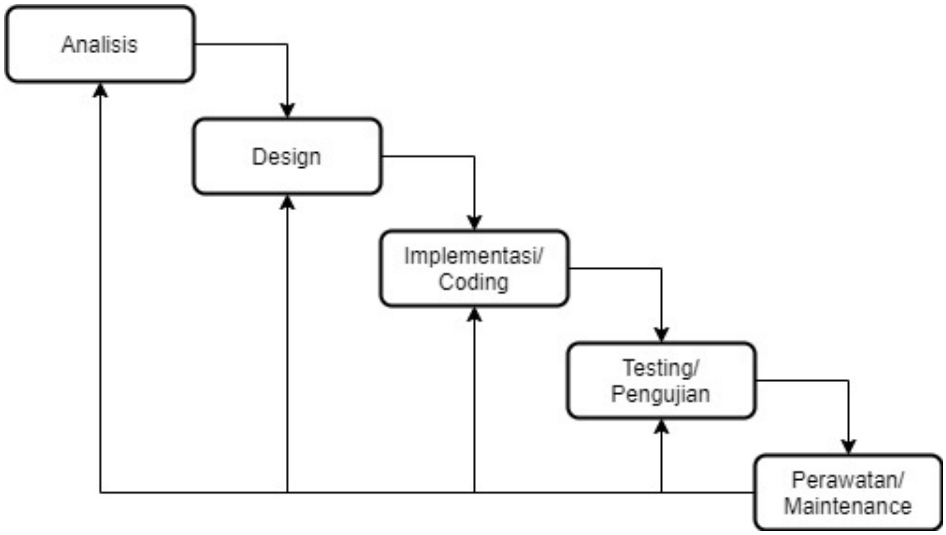
Noor Azizah & Nur Aeni W (2018) menyatakan bahwa "*Geolocation* merupakan proses mendeteksi lokasi keberadaan kita menggunakan koneksi *internet* berdasarkan titik koordinat *longitude* dan *latitude*". *Geolocation* merupakan singkatan dari *Global Positioning System* atau biasa disingkat GPS. Pada aplikasi ini menggunakan sebuah API (*Application Programming Interface*) dari HTML5 yang diperuntukkan untuk Layanan Berbasis Lokasi (LBS). Dengan fitur geolokasi HTML5 dapat memungkinkan untuk mengetahui koordinat geografis (angka lintang dan bujur) dari lokasi pengunjung situs *web* anda saat ini. Dengan penerapan geolokasi ini dapat berfungsi untuk mengetahui letak dimana mahasiswa melakukan absensi. Karena perangkat digunakan oleh individu, maka geolokasi menggunakan sistem penentuan posisi untuk melacak keberadaan individu menurut garis lintang dan garis bujur.

2.20 Metodologi Pengembangan Sistem

1. Waterfall

System Development Life Cycle (SDLC) merupakan metodologi umum yang digunakan untuk mengembangkan sistem informasi. Model *waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *waterfall* antara lain *requirement, design, implementation, verification,* dan *maintenance* (Wahid, 2020). Dari hal tersebut SDLC sendiri terdiri dari beberapa fase yang dimulai dari fase perencanaan, analisis, *design*, implementasi, pengujian hingga pemeliharaan sistem. Kelebihan menggunakan metode *waterfall* dalam pengembangan sistem informasi adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang

diperlukan juga mahal. Dengan adanya urutan yang pasti, dapat dilihat pula progress untuk setiap tahap secara pasti.



Gambar 2.11 Metodologi *Waterfall*

a. Analisis

Menurut Azwar (2019) "Analisis adalah memecahkan atau menguraikan sesuatu unit menjadi unit terkecil". Analisis sering dilakukan untuk memperoleh kesimpulan mengenai pelaksanaan kegiatan tersebut. Analisa atau pengumpulan data - data yang berkaitan dengan sistem yang akan dibuat. Metode pengumpulan data dapat diperoleh melalui diskusi, observasi, survei, dan wawancara. Dari pendapat diatas dapat ditarik kesimpulan bahwa analisis merupakan suatu kegiatan berfikir untuk menguraikan atau memecahkan suatu permasalahan dari unit menjadi unit terkecil.

b. Design

Perancangan/*design* dilakukan dengan tujuan untuk memberikan gambaran mengenai apa saja yang harus dikerjakan. *Design*/perancangan adalah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta *detail* komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya (Adiguna dkk., 2018).

c. Implementasi

Implementasi adalah berupa *coding*/kode dimana dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum. Tujuan implementasi kebijakan adalah untuk menetapkan arah agar tujuan kebijakan dapat teralisasi (Elih Yulia, 2020).

d. Testing

Menurut Cholifah, dkk (2018) ”Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan”. Proses selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan dengan menggunakan *Black Box* untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

e. Maintenance

Pemeliharaan/*maintenance* memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan. *Maintenance* dapat diartikan sebagai kegiatan untuk memelihara, menjaga fasilitas peralatan pabrik dan mengadakan perbaikan, penyesuaian atau penggantian yang diperlukan agar terdapat suatu keadaan operasi produksi yang memuaskan sesuai dengan apa yang direncanakan (Pasaribu, 2021).

2.21 Unified Modelling Language (UML)

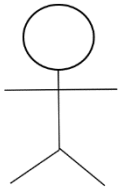
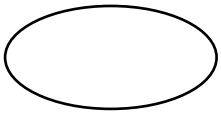
Rachmat Destriana, dkk (2021:1) berpendapat bahwa “UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifacts* tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem *non* perangkat lunak lainnya. Terdapat beberapa diagram yang didefinisikan

dalam UML, beberapanya adalah *use case* diagram, *activity* diagram, *class* diagram, dan *sequence* diagram.

1. Use Case Diagram

Use case diagram adalah salah satu jenis diagram yang ada pada UML yang menggambarkan interaksi antara sistem dan aktor. *Use case* diagram juga dapat mendeskripsikan tipe interaksi antara pengguna sistem dengan sistemnya. *Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang menggambarkan sebuah interaksi antara aktor terhadap sistem (Khadaffi, 2020). Berikut simbol-simbol yang digunakan dalam *use case* diagram disertai dengan keterangan fungsinya sebagaimana dijelaskan pada tabel 2.2.

Tabel 2.1 Simbol *Use Case* Diagram

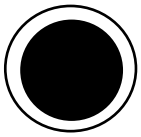
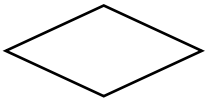
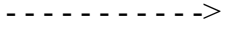
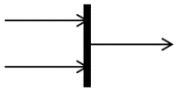
NO	SIMBOL	KETERANGAN
1.		Aktor: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*.
2.		*Use Case*: Abstraksi dan interaksi antara sistem dan aktor.
3.		*Association*: Abstraksi dari penghubung antara aktor dengan *use case*.
4.	- - - - - - - - - - ->	Generalisasi: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan *use case.*
5.	<<include>>	Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
6.	<<extend>>	Menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi.

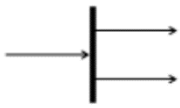
2. Activity Diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. *Activity* diagram adalah diagram yang menggambarkan alur kerja dari berbagai aktivitas *user* atau sistem, orang yang melakukan aktivitas, dan aliran berurutan dari aktivitas ini (Rachmat Destriana, dkk, 2021:5).

Berikut simbol-simbol yang digunakan dalam *activity* diagram disertai dengan keterangan fungsinya sebagaimana dijelaskan pada tabel 2.3.

Tabel 2.2 Simbol *Activity* Diagram

NO	SIMBOL	KETERANGAN
1.		*Start point*: Untuk memperlihatkan dimana aliran kerja berawal.
2.		*End point*: Untuk memperlihatkan dimana aliran kerja berakhir.
3.		*Activities*: Untuk menggambarkan langkah-langkah dalam aliran kerja.
4.		*Decision*: Pilihan untuk mengambil keputusan.
5.		*Control flow*: Menunjukkan urutan eksekusi.
6.		*Object flow*: Menunjukan aliran objek dari sebuah *action* atau *activity* ke *action.*
7.		*Join*/penggabungan: Menyatakan untuk menggabungkan kembali *activity* atau *action* yang *parallel.*

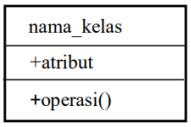
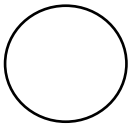
NO	SIMBOL	KETERANGAN
8.		*Fork*/percabangan: Menyatakan untuk memecah *behavior* menjadi *activity* atau *action* yang *parallel.*

3. Class Diagram

Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. *Class* diagram adalah sebuah diagram yang menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan menjelaskan bagaimana caranya agar mereka saling berkolaborasi (Rachmat Destriana, dkk, 2021:5).

Berikut simbol-simbol yang digunakan dalam *class* diagram disertai dengan keterangan fungsinya sebagaimana dijelaskan pada tabel 2.4.

Tabel 2.3 Simbol *Class* Diagram

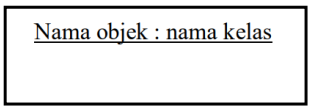
NO	SIMBOL	KETERANGAN
1.		*Class*: Kelas dalam struktur sistem.
2.		*Interface*/antarmuka: Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
3.		Asosiasi/*association*: Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
4.		Asosiasi berarah/*directed association*: Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*

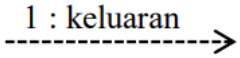
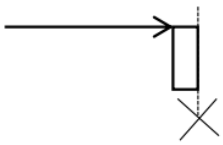
NO	SIMBOL	KETERANGAN
5.		Generalisasi: Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).
6.		Agregasi/*aggregetion*: Relasi antar kelas dengan makna semua bagian (*wholepart*).

4. Sequence Diagram

Rachmat Destriana, dkk (2021:5) menyatakan bahwa "*Sequence* diagram adalah diagram yang menggambarkan interaksi *user* dengan sistem secara sekuensial(berurutan)". Dalam hal ini pada *sequence* diagram menggambarkan sejumlah aktivitas atau kolaborasi antar *object*. Fungsi dari diagram ini yaitu untuk menunjukkan interaksi atau pesan yang disampaikan antar setiap *object*-nya.
Berikut simbol-simbol yang digunakan dalam *sequence* diagram disertai dengan keterangan fungsinya sebagaimana dijelaskan pada tabel 2.5.

Tabel 2.4 Simbol *Sequence* Diagram

NO	SIMBOL	KETERANGAN
1.		*Object*: Menyatakan objek yang berinteraksi pesan.
2.		Garis hidup/*lifeline*: Menyatakan kehidupan suatu objek.
3.		Waktu Aktif: Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.

NO	SIMBOL	KETERANGAN
4.		Pesan tipe *create*: Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. Untuk pesan masukan dan tipe *call* memiliki simbol yang sama.
5.		*Message return type*: Menunjukan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya.
6		Pesan tipe *destroy*: Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy.*

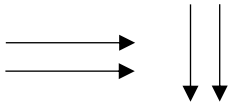
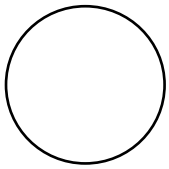
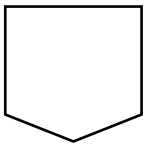
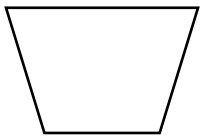
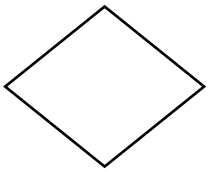
2.22 Flowchart

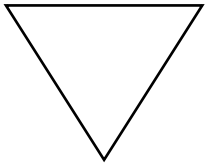
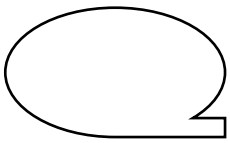
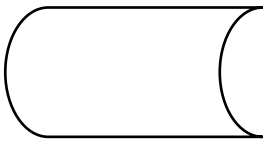
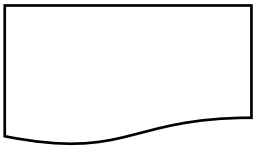
Menurut Maarif, V., Nur, H. M., & Rahayu (2018) "*Flowchart*, merupakan langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu". Diagram alir atau *flowchart* ini sendiri memiliki dua istilah yaitu *chart*(bagan) dan *flow*(hasil) yang memiliki prosedur sistem secara logika. Diagram alir ini selain dibutuhkan sebagai alat bantu komunikasi, juga diperlukan sebagai dokumentasi.

Flowchart memiliki 3 kelompok simbol yang dipakai:

1. *Flow Direction Symbol*, yang juga disebut *connecting line* yaitu simbol yang menampilkan sebuah arah arus untuk menghubungkan simbol satu dengan yang lain.
2. *Processing Symbol*, yaitu simbol-simbol yang menampilkan penjelasan dari sebuah sistem.
3. *Input/Output Symbol*, simbol yang digunakan untuk menampilkan jenis peralatan yang digunakan.

Tabel 2.5 Simbol Diagram *Flowchart*

NO	SIMBOL	KETERANGAN
1.		Arus/*flow Symbol*, berfungsi untuk menghubungkan satu simbol dengan simbol lainnya dan juga menjelaskan jalannya arus suatu proses.
2.		*Connector Symbol*, berfungsi untuk menyatakan sambungan dari proses yang satu ke proses lainnya dalam halaman yang sama.
3.		*Offline connector Symbol,* berfungsi untuk menyatakan sambungan dari proses yang satu ke proses lainnya dalam halaman yang berbeda.
4.		*Processing Symbol*, berfungsi untuk menyatakan suatu tindakan atau proses yang dilakukan oleh komputer.
5.		*Manual Symbol*, berfungsi untuk menyatakan suatu tindakan atau proses yang tidak dilakukan oleh komputer.
6.		*Decision Symbol*, berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya atau tidak.
7.		*Terminal Symbol*, berfungsi untuk menyatakan permulaan atau akhir suatu program.
8.		*Predefined Process Symbol*, berfungsi untuk menyatakan penyediaan tempat

NO	SIMBOL	KETERANGAN
		penyimpanan suatu pengolahan untuk memberi harga awal.
9.		*Keying Operation Symbol*, berfungsi untuk menyatakan semua jenis operasi yang diproses dengan menggunakan suatu mesin yang memiliki *keyboard.*
10.		*Offline-storage Symbol*, berfungsi untuk menunjukkan bahwa data dalam simbol akan disimpan ke suatu media tertentu.
11.		*Manual Input Symbol,* berfungsi untuk memasukkan data secara *manual* dengan menggunakan *online keyboard.*
12.		*Input/Output Symbol*, berfungsi untuk menyatakan proses *input* atau *output* tanpa tergantung jenis peralatan.
13.		*Magnetic Tape Symbol*, berfungsi untuk menyatakan *input* berasal dari pita magnetis atau *output* disimpan ke pita magnetis.
14.		*Disk Storage Symbol*, berfungsi untuk menyatakan *input* yang berasal dari *disk* atau *output* disimpan ke *disk.*
15		*Document Symbol*, berfungsi untuk menyatakan *input* berasal dari dokumen dalam bentuk kertas dan mencetak *output* dalam bentuk dokumen (melalui *printer*). *Punched Card Symbol*, berfungsi untuk

NO	SIMBOL	KETERANGAN
16.		menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu.