

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Absensi**

Absensi adalah suatu bentuk pendataan presensi atau kehadiran seseorang atau pegawai yang merupakan bagian pelaporan dari suatu institusi yang berisi data – data status kehadiran yang disusun dan diatur secara rapi dan mudah untuk dicari, dan digunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan (Erna Simonna, 2009).

Menurut Joko Supriyono (2013), terdapat beberapa jenis absensi yang dibedakan berdasarkan cara penggunaan dan tingkat daya gunanya. Secara umum, jenis – jenis absensi dibagi menjadi dua, yaitu :

1. Absensi secara manual, yaitu cara pengentrian kehadiran atau kedatangan dengan cara menggunakan pena melalui tanda tangan dan kertas.
2. Absensi secara non manual, yaitu cara pengentrian kehadiran atau kedatangan dengan menggunakan alat atau dengan menggunakan sistem terkomputerisasi seperti penggunaan kartu dengan kode batang (*barcode*) dan pengambilan sidik jari (*fingerprint*).

Dalam dunia perkuliahan terlebih pada bidang akademik, presensi merupakan salah satu data yang penting. Sebelum memulai atau sesudah mengikuti kegiatan perkuliahan, setiap mahasiswa diminta untuk mengisi daftar kehadiran atau presensi sebagai bukti bahwa mahasiswa telah hadir dan mengikuti kelas mata kuliah yang diambilnya. Selain itu, data absensi sangat diperlukan pihak administrasi akademik untuk memonitor atau memantau keaktifan mahasiswa dalam mengikuti setiap kegiatan dan aktifitas pendukung akademik.

Penyebab mahasiswa melakukan absen (tidak hadir) dipengaruhi beberapa hal seperti kondisi kesehatan yang dialami mahasiswa sehingga harus mengirimkan surat medis kepada pihak administrasi akademik yang ada di kampus, adanya ketidakcocokkan dengan mata kuliah yang sedang diambil atau dari keinginan

mahasiswa itu sendiri untuk hadir atau tidak hadir dalam kelas mata kuliah yang sedang dijalaninya.

## **2.2 Aplikasi**

### **2.2.1 Pengertian Aplikasi**

Menurut Kamus Besar Bahasa Indonesia (KBBI) (1998), aplikasi adalah penerapan dari rancang sistem untuk mengelolah data yang menggunakan aturan – aturan atau ketentuan – ketentuan bahasa pemograman tertentu. Selain itu, aplikasi adalah penggunaan suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses suatu *input* menjadi *output* yang sesuai dengan keinginan (Jogiyanto, 1999).

Berdasarkan pengertian yang disebutkan diatas, dapat disimpulkan bahwa aplikasi adalah suatu sistem yang terdiri dari perintah atau instruksi tertentu yang dibuat secara rapi dan kompleks yang dapat diterapkan pada suatu komputer untuk dapat memasukkan suatu masukan (*input*) dan dapat menghasilkan suatu keluaran (*output*) yang sesuai dengan masukan yang diinginkan sebelumnya.

### **2.2.2 Aplikasi Web (*Web Application*)**

#### **2.2.2.1 Pengertian Website**

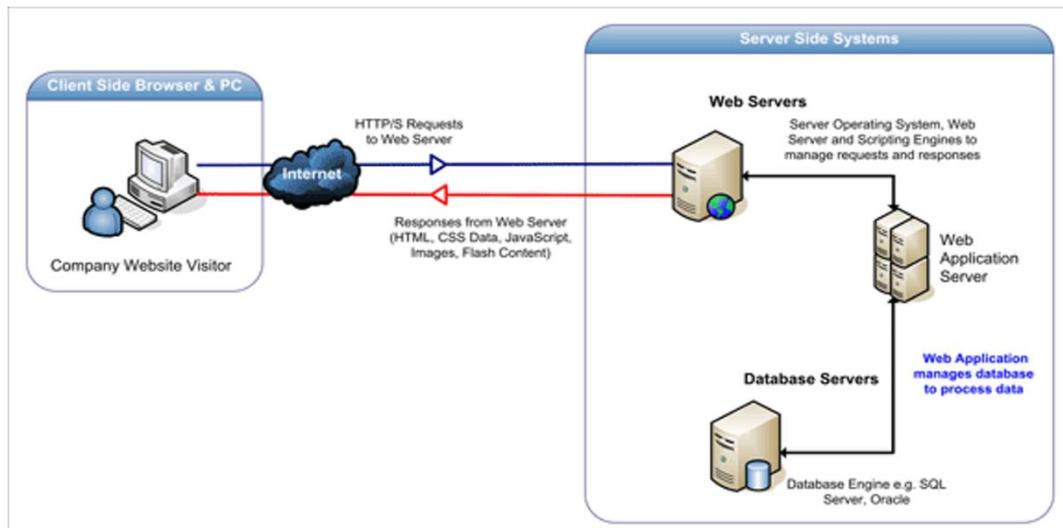
*Website* merupakan kumpulan halaman web yang saling terhubung dan file – filenya saling berkaitan. Web terdiri dari *page* atau halaman dan kumpulan dari halaman – halaman tersebut disebut dengan *homepage*. *Homepage* biasanya memiliki halaman yang saling berkaitan dengan halaman yang berada dibawahnya. Setiap halaman yang berada dibawahnya disebut dengan *child page* yang berisi *hyperlink* ke halaman lainnya yang ada diweb (Gregorius, 2000).

Selain itu, *website* diartikan sebagai salah satu aplikasi yang terdiri dari beragam dokumen multimedia seperti teks, gambar, animasi dan video yang menggunakan protocol HTTP (*Hypertext Transfer Protocol*) dan cara untuk mengaksesnya dengan menggunakan perangkat lunak yang disebut dengan *browser* seperti Mozilla Firefox, Internet Explorer, Opera, Google Chrome, Safari dan lainnya.

### 2.2.2.2 Aplikasi Web

Aplikasi web atau *Web Applications* adalah aplikasi yang dapat diakses oleh *user* atau pengguna melalui jaringan internet atau intranet (Apers Chris, 2012). Aplikasi web diartikan sebagai perangkat lunak yang dikodekan dengan bahasa pemrograman (PHP atau Javascript) dan dikombinasikan dengan *HyperText Markup Language* yang dapat terintegrasi dan dijalankan dengan *browser*.

Aplikasi web memiliki tiga (3) *layer* yaitu *layer* pertama berada disisi *client* yang memiliki sistem *browser* dasar, *layer* kedua terdapat *dynamic content generation tool* seperti JavaScript dan PHP dan *layer* ketiga terdapat penyimpanan data yang terdiri dari *back end* dari *database* seperti MySQL atau Oracle seperti pada penjelasan gambar 2.1.



Gambar 2.1 Cara Kerja *Web Application*

Sumber : Apper Chris, 2012. *Beginning Iphone and Ipad Web Apps Scripting with HTML5, CSS3 & JavaScript*. New York : Apress

### 2.2.3 Aplikasi *Mobile* (*Mobile Application*)

#### 2.2.3.1 Pengertian *Mobile*

Kata *Mobile* berasal dari bahasa Inggris yang memiliki arti berpindah. Dapat diartikan bahwa *mobile* adalah perpindahan dari suatu tempat ke tempat lain yang dapat digunakan kapanpun dan dimanapun dengan menggunakan perangkat yang

berbasis *mobile* seperti telepon pintar (*smartphone*), telepon seluler, *pager*, PDA (*Portable Digital Assistant*) dan lainnya.

### 2.2.3.2 Aplikasi *Mobile*

Menurut Pressman dan Bruce (2014) Aplikasi *Mobile* adalah sebuah aplikasi yang telah dirancang khusus untuk *platform mobile* seperti *iOS*, *android* atau *windows mobile*. Pada banyak kasus, aplikasi *mobile* memiliki antarmuka pengguna (*user interface*) unik yang telah disediakan oleh penyedia *platform mobile* dengan sumber daya berbasis web, menyediakan akses ke semua jenis informasi yang relevan dengan aplikasi dan kemampuan untuk memproses, mengumpulkan dan menganalisis format informasi dengan prosedur yang sesuai dengan *platform*. Selain itu, aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan *user* melakukan mobilitas dengan menggunakan perlengkapan seperti PDA (*Portable Digital Assistant*), telepon seluler atau *Handphone* (Raharja, 2009).

Aplikasi *mobile* menurut Brian Fling (2009) dibagi menjadi beberapa kelompok, yaitu :

a. *Short Message Service* (SMS)

*Short Message Service* (SMS) adalah aplikasi *mobile* sederhana yang dirancang untuk mengirimkan pesan (teks) dan berguna ketika terintegrasi dengan jenis aplikasi *mobile* lainnya.

b. *Mobile Websites* (Situs *Web Mobile*)

*Mobile Websites* (Situs *Web Mobile*) adalah situs *web* yang dirancang khusus untuk perangkat yang berbasis *mobile*. Situs *Web Mobile* biasanya memiliki desain sederhana dan bersifat memberikan informasi.

c. *Mobile Web Application* (Aplikasi *Web Mobile*)

*Mobile Web Application* (Aplikasi *Web Mobile*) adalah aplikasi *mobile* yang penggunaannya tidak perlu lagi diinstal atau dikompilasi pada perangkat *mobile*. Biasanya aplikasi *web mobile* menggunakan XHTML (*eXtensible HyperText Markup Language*), CSS (*Cascading Style Sheets*) dan JavaScript, aplikasi *web mobile* mampu memberikan pengalaman kepada *user* layaknya aplikasi *native* atau asli.

d. *Native Application* (Aplikasi Asli)

*Native Application* atau aplikasi *platform* adalah aplikasi mobile yang harus diinstal pada perangkat *mobile*, karena aplikasi ini dikembangkan dan disusun secara khusus untuk perangkat *mobile* oleh penyedia *platform mobile*.

### 2.3 Kode QR

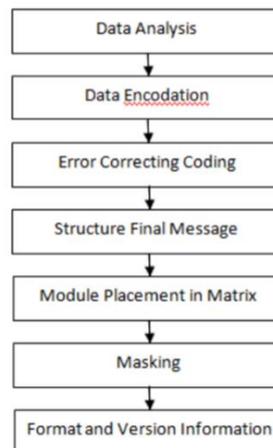
Kode QR atau *Quick Response Code* (*QR Code*) adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. Kode QR merupakan evolusi dari kode batang (*barcode*) (Nugraha, M. Pasca dan Rinaldi Munir, 2011). *Barcode* merupakan sebuah symbol penandaan objek nyata yang terbuat dari pola batang – batang berwarna hitam dan putih agar mudah untuk dikenali komputer (Rahayu, Yeni Dwi, dkk., 2006). Contoh sebuah Kode QR dapat dilihat pada gambar 2.2.



Gambar 2.2 Contoh Kode QR

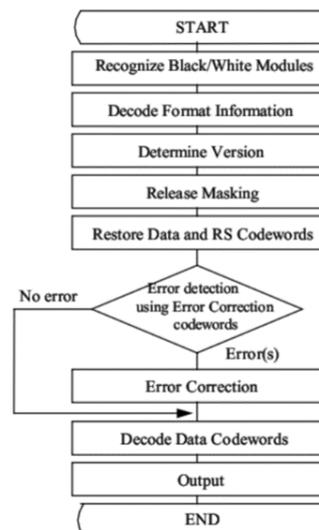
Kode QR dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. Kode QR dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula (Nugraha, M. Pasca dan Rinaldi Munir, 2011).

Prosedur pembangkitan Kode QR dari sebuah teks dapat dijelaskan melalui diagram alir pada gambar 2.3.



Gambar 2.3 Diagram prosedur pembangkitan Kode QR

Langkah – langkah untuk membaca Kode QR menjadi suatu teks merupakan kebalikan dari langkah – langkah pembangkitan Kode QR. Berikut prosedur pembacaan Kode QR pada gambar 2.4.



Gambar 2.4 Diagram prosedur pembacaan Kode QR

## 2.4 Android

Menurut Safaat (2015), *Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Selain itu, menurut Boedjiono, *et al* (2015), *Android* adalah *software stack* Google untuk perangkat *mobile*. Pengertian lainnya, *Android* adalah sebuah sistem operasi

untuk perangkat *mobile* yang menyertakan *middleware* (*virtual machine*) serta sejumlah aplikasi utama dan merupakan modifikasi dari kernel *linux* (Andry, 2011).

Tujuan dari pengembangan sistem operasi ini adalah untuk menyediakan *platform* yang bersifat *open source* atau terbuka yang membantu pengguna mudah mengakses internet menggunakan telepon seluler. *Android* dirancang untuk mempermudah pengembangan pembuatan aplikasi dengan Batasan yang minim sehingga kreatifitas pengembang aplikasi menjadi lebih berkembang (Andry, 2011).

Karena *android* merupakan sistem operasi yang berbasis *open source* dan bebas untuk dimodifikasi, *android* tidak memiliki ketentuan yang tetap dalam pengaturan perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut fitur – fitur dalam *android*, menurut Lee Wei Meng (2011) :

- a. *Storage* , menggunakan SQLite, *database* yang ringan untuk penyimpanan data.
- b. *Connectivity* , mendukung GSM atau EDGE, IDEN, CDMA, EV-DO dan UMTS.
- c. *Bluetooth*, *Wifi*, LTE dan WiMax sebagai fitur tambahan dalam mengirim berkas dan mengakses jaringan internet.
- d. *Messaging (Message)*, mendukung fitur mengirimkan *Short Message Service* (SMS) dan *Multimedia Messaging Service* (MMS).
- e. *Web Browser*, berdasarkan pada *open source* WebKit, bersama mesin.
- f. *Chrome's V8 JavaScript*.
- g. *Media support*, termasuk mendukung untuk beberapa media
  - H.263, H.264 (3GP atau MP4), MPEG-4 SP, AMR, AMRWB (3GP), ACC, HE-ACC (MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, GIF, dan BMP.
- h. *Hardware support*, mendukung sensor akselerasi, kamera, kompas digital, sensor kedekatan, dan GPS.
- i. *Multi-touch*, mendukung fitur layer sentuh (*multi-touch screens*).
- j. *Multi-tasking*, mendukung fitur aplikasi *multi-tasking*.

- k. *Flash-support*.
- l. *Tethering*, mendukung pembagian dari koneksi internet sebagai *wired* atau *wireless hotspot*.
- m. *Play store*, katalog aplikasi yang dapat *download* dan diinstal pada *smartphone* secara *online*, tanpa menggunakan PC (*Personal Computer*).

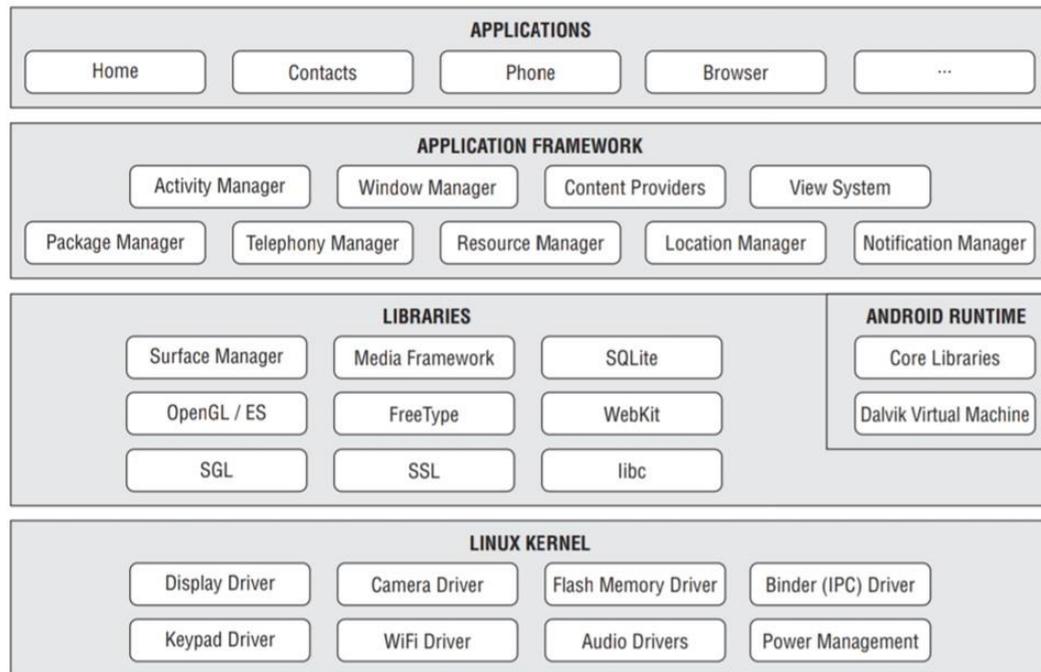
Pengembangan sistem operasi *android* menghadirkan banyak versi yang beragam, berikut tabel versi yang pernah dikembangkan oleh *android* :

Tabel 2.1 Tabel Versi *Android*

<b>Versi <i>Android</i></b>	<b>Nama</b>
Android 1.0	Astro (Alpha)
Android 1.1	Bender (Beta)
Android 1.5	Cupcake
Android 1.6	Donut
Android 2.0 – Android 2.1	Éclair
Android 2.2	Froyo (Frozen Yoghurt)
Android 2.3	Gingerbread
Android 3.0 – Android 3.2	Honeycomb
Android 4.0	Ice Cream Sandwich
Android 4.1 – Android 4.3	Jelly Bean
Android 4.4	KitKat
Android 5.0 – Android 5.1	Lollipop
Android 6.0	Marshmallow
Android 7.0 – Android 7.1	Nougat
Android 8.0 – Android 8.1	Oreo
Android 9.0	Pie
Android 10.0	Q

Sumber : <http://www.nesabamedia.com/urutan-tingkatan-android/>

Untuk memahami bagaimana *android* berkerja, Lee Wei Meng (2011) berikut gambar dari beberapa *layer* yang membangun sistem operasi *android* :



Gambar 2.5 Arsitektur Android (Lee Wei Meng, 2011)

Sistem operasi *Android* dibagi menjadi lima (5) bagian dari empat (4) *layer*, yaitu :

a. *Linux kernel*

*Layer* ini berada dibagian bawah. *Layer* ini mengandung semua *driver* perangkat tingkat untuk berbagai komponen perangkat keras dari *android*. Dalam *layer* ini terdapat *Display Driver*, *Camera Driver*, *Flash Memory Driver*, *Binder (IPC) Driver*, *Keypad Driver*, *WiFi Driver*, *Audio Drivers*, dan *Power Management*.

b. *Libraries*

*Libraries* adalah kumpulan dari kode yang digunakan oleh *Android OS*. Kode ini ditulis dengan bahasa pemograman C atau C++. Pada *layer* ini terdapat *Surface Manager*, *Media Framework*, *SQLite*, *OpenGL/ES*, *FreeType*, *WebKit*, *SGL*, *SSL*, dan *libc*.

c. *Android Runtime*

*Android Runtime* atau sering disebut dengan *middleware* merupakan bagian *layer* dimana *android* berjalan. Aplikasi *android* menggunakan bahasa

pemrograman JavaScript. Dalam *layer* ini terdapat dua (2) bagian yaitu *Core Libraries* untuk membuat aplikasi *android* dengan bahasa Java dan *Dalvik Virtual Machine* sebagai tempat untuk menjalankan aplikasi *android*. *Dalvik Virtual Machine* merupakan mesin virtual yang didesain secara khusus dan dioptimasi untuk perangkat yang memiliki memori kecil, sumber tenaga terbatas, dan kemampuan prosesor yang kecil. *Android Runtime* berada di *layer* yang sama dengan *Libraries*.

d. *Application Framework*

*Layer* ini menampilkan berbagai kapabilitas dari *android* untuk pengembang aplikasi sehingga dapat memanfaatkannya dalam aplikasi. Dalam *layer* ini terdapat *Activity Manager*, *Window Manager*, *Content Providers*, *View System*, *Package Manager*, *Telephony Manager*, *Resource Manager*, *Location Manager*, dan *Notification Manager*.

e. *Applications*

*Layer* ini berada dibagian atas, yang didalamnya terdapat aplikasi – aplikasi yang dapat dijalankan dan dioperasikan oleh pengguna dengan menggunakan sistem operasi *android* ini.

## 2.5 *Hypertext Preprocessor (PHP)*

Menurut Sutarman (2007), *Hypertext Preprocessor* atau PHP adalah bahasa *server – side* yang didesain khusus untuk aplikasi web. PHP dapat disisipkan dalam bahasa HTML (*HyperText Markup Language*) dan bahasa PHP akan dieksekusi oleh server sehingga pada saat membuka *browser* bahasa PHP akan diubah kedalam bentuk HTML dan bahasa PHP tidak akan terlihat di *browser*. Selain itu, menurut Kasiman Peranginangin (2006), penggunaan bahasa PHP memungkinkan *website* atau Web dapat dibuat secara dinamis sehingga *maintenance website* atau situs web menjadi lebih mudah dan efisien.

Bahasa PHP memiliki banyak keunikan yang tidak dimiliki oleh bahasa pemrograman lainnya. Bahasa PHP difokuskan pada pembuatan *script* secara *server – side* yang dapat melakukan apa saja seperti *Computer – Generated Imagery (CGI)* seperti mengumpulkan data yang berasal dari *form*, menghasilkan halaman web yang dinamis dan mengirim serta menerima *cookies* (Kasmin Peranginangin, 2006).

Bahasa PHP tidak memiliki batasan pada hasil *output* HTML (*HyperText Markup Language*). Bahasa PHP memiliki kemampuan untuk mengelolah gambar, file PDF, video, dan *flash*. Bahasa PHP juga dapat menghasilkan teks seperti XHTML (*eXtensible HyperText Markup Language*) dan file XML lainnya (Kasmin Peranginangin, 2006).

## 2.6 MySQL

Menurut Abdul Kadir (2008), MySQL merupakan perangkat lunak (*software*) yang tergolong sebagai *Database Management System* (DBMS) yang memiliki sifat terbuka (*open source*) yang dilengkapi dengan kode – kode (*source code*) khusus yang digunakan untuk membuat MySQL.

Menurut Ahmad Solichin (2001) dalam MySQL terdapat fitur – fitur yang sangat melekat pada perangkat lunak ini, yaitu :

- a. *Relational Database System*.
- b. Arsitektur *Client – Server*.
- c. Mengenal perintah *Structured Query Language* (SQL) standar.
- d. Mendukung *Sub Select*.
- e. Mendukung *View*, *Stored Prosedured* (SP), *Triggers*, *Replication*, *Transaction*, dan *Foreign Key* (FK).
- f. Tersedia fungsi GIS, bebas diunduh (*dowloand*), stabil, tangguh, fleksibel, dan memiliki keamanan (*securiry*) yang baik.

## 2.7 Kajian Teori Tentang Aplikasi yang akan digunakan

### 2.7.1 Android Studio

Android Studio adalah lingkungan pengembangan terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Android Studio menawarkan banyak fitur yang meningkatkan produktivitas yaitu sistem *build* berbasis *Gradle* yang fleksibel, *emulator* yang cepat, pengkodean terintegrasi oleh *GitHub*, mendukung C++ dan NDK serta dukungan bawaan untuk *Google Cloud Platform* yang memudahkan integrase *Google Cloud Messaging* dan *App Engine* (Developers, 2019).

### 2.7.2 Android SDK

Android SDK (*Software Development Kit*) adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Saat ini, Android SDK menjadi sebagai alat bantu dan API untuk mengembangkan aplikasi *platform* android menggunakan bahasa pemrograman Java (Developers, 2014).

### 2.7.3 XAMPP

XAMPP merupakan singkatan dari Apache, MySQL, PHP, dan Perl sedangkan huruf “X” sebagai suatu perangkat lunak yang dapat dijalankan di 4 (empat) sistem operasi seperti Windows, Mac OS, Linux dan Solaris. Oleh karena itu, XAMPP sering disebut dengan *cross platform (software multi OS)* (Nesabamedia, 2019).

*Software* ini merupakan gabungan dari beberapa fungsi yang sama yaitu membantu pembuat web dengan adanya *web server* sendiri di PC (*personal computer*) atau Laptop. *Software* ini berlisensi GNU serta dapat *download* secara gratis (*free*) dan dapat membantu untuk pembuat web pemula (Nesabamedia, 2019).

XAMPP didirikan oleh suatu perusahaan bernama Apache Friends dengan tujuan untuk membuat *web server* pribadi sehingga dapat membantu membuat tampilan web yang dinamis, istilah ini biasa disebut dengan *localhost* (Nesabamedia, 2019).

### 2.7.4 Flutter

*Flutter* adalah *framework* pengembangan perangkat lunak bersifat *open source* yang dikembangkan oleh Google untuk membuat *mobile application* pada iOS dan Android dengan menggunakan basis kode tunggal (*single code basic*). Basis kode tunggal (*single code basic*) memungkinkan pengembang untuk mengembangkan aplikasi lebih cepat dan murah. *Flutter* juga mengkompilasi kode ke dalam program asli (*native program*) yang dapat dijalankan jauh lebih cepat di iOS dan Android. *Flutter* menggunakan bahasa pemrograman *Dart (.dart)* sebagai bahasa utama dalam pengkodean. Desain antarmuka (*design interface*) dan kode logika (*logic codes*) dapat digabungkan menjadi satu dengan file *Dart* dengan ekstensi *.dart*. (Ahmad Hanis Mohd Shabli, 2019).

### 2.7.5 Visual Studio Code

Visual Studio Code adalah editor pengkodean yang bersifat *open source* yang dikembangkan oleh Microsoft untuk Windows, Linux, dan OS X. Terdapat dukungan untuk *debugging*, *embedded git control*, *snippets*, dan *code refactor*. Visual Studio Code juga dapat disesuaikan, sehingga pengguna (*user*) dapat mengubah tema editor (*the theme's editor*), pintasan papan ketik (*keyboard shortcuts*), dan preferensi (*preference*) (Frederic Lardinois, 2015).

### 2.7.6 Notepad++

Notepad++ merupakan suatu aplikasi *code editor* bersifat *free software* yang dapat dijalankan diperangkat yang memiliki *Operating System* (OS) *Windows*. Notepad++ menggunakan komponen Scintilla untuk bisa menampilkan dan menyunting sebuah *source code* atau *coding* dari berbagai bahasa pemrograman. Selain itu, bahasa pemrograman yang didukung oleh Notepad++ adalah bahasa C++.

Keunggulan dari Notepad++ yang dapat dirasakan oleh pengguna terutama dalam *web programming* yaitu sederhana, cepat dan ringan. Selain itu, Notepad++ dapat membantu programming dalam melengkapi *code* yang bersifat percabangan (*for*) dan perulangan (*while*). Adanya *block program* yang dapat membantu programmer yang sedang menulis program yang panjang sehingga lupa untuk menutup *block program* pada suatu *syntax*. Kemudian adanya *Quick color picker* dimana sangat berguna pada saat menulis kode warna pada HTML ataupun CSS tetapi tidak harus menuliskan kode dahulu, jika kita lupa dengan kode warna yang kita inginkan biasanya akan muncul kotak dialog yang menampilkan banyak warna yang kita inginkan. (Sadewa, Universitas Brawijaya)

### 2.7.7 Bootstrap

Bootstrap adalah sebuah *framework* atau *library* atau kumpulan dari beberapa fungsi yang digunakan untuk membangun sebuah CSS dan dibuat secara khusus untuk pengembangan *front-end website*. Bootstrap termasuk dalam salah satu dari *framework* HTML, Javascript dan CSS. Bootstrap kini sudah banyak digunakan oleh para pengembang website (*Web Developer*) karena lebih mudah dan cepat dalam membuat suatu *grid*, tombol, navigasi, desain *interface* pada suatu *website*

tanpa harus membuang banyak waktu dalam mendesain halaman website (Nesabamedia, 2019).

Kelebihan dari Bootstrap adalah bisa digunakan untuk membantu mempercepat *front-end* pada website, memiliki tampilan yang modern sehingga bisa membuat website yang menarik dan kreatif sesuai dengan keinginan pengembang web. Selain itu, Bootstrap memiliki kelengkapan yang baik seperti sudah mencakup CSS, HTML, dan juga Javascript serta sifat aplikasi *Open Source* sehingga dapat digunakan tanpa harus membayar dan harus mempunyai lisensi (Nesabamedia, 2019).

Kekurangan dari Bootstrap adalah tidak semua browser dapat menampilkan tampilan website yang sama dan hanya dapat ditampilkan sesuai dengan layar perangkat yang digunakan. Kemudian, Bootstrap mendukung CSS3 yang dimana CSS3 minim atau kurang terhadap gambar (Nesabamedia, 2019).

### **2.7.8 Google Formulir atau Google Form**

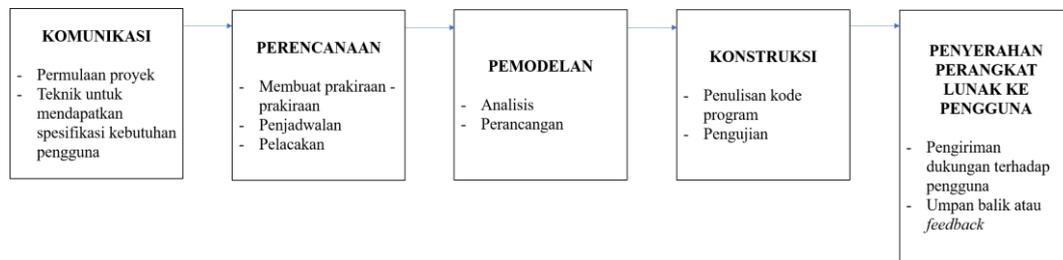
*Google* Formulir atau *Google Form* merupakan bagian dari inovasi dari *Google Docs* yang digunakan untuk membantu membuat pertanyaan dalam bentuk formulir atau kuisisioner (Silvia, 2019). *Google* Formulir atau *Google Form* juga merupakan sistem yang berupa *template* formulir yang dapat digunakan untuk mendapatkan informasi dari pengguna. Selain itu, penyimpanan data dari *Google* Formulir disimpan dalam penyimpanan *Google Drive* (Untung Rahardja, dkk. 2018).

## **2.8 Metode Pengembangan Perangkat Lunak**

### **2.8.1 Metodologi Air Terjun (*Waterfall*) atau Metode *System Development Life Cycle* (SDLC)**

Metodologi Air Terjun (*Waterfall*) atau Metode SDLC adalah metode pengembangan perangkat lunak yang menggunakan pendekatan yang sistematis dan berurutan atau sekuensial. Dalam metode ini, pengembangan dimulai dari spesifikasi kebutuhan *user* atau analisis kebutuhan (*requirement analysis*), melakukan tahap perencanaan atau *planning*, tahap pemodelan (*modelling*), tahap konstruksi (*construction*) serta penyerahan perangkat lunak kepada pengguna atau *user* (*deployment*) dan terakhir pengembangan berkelanjutan (*maintenance*) pada

perangkat lunak (Roger, 2010). Berikut penjelasan pengembangan metode air terjun atau *waterfall* pada gambar 2.6.



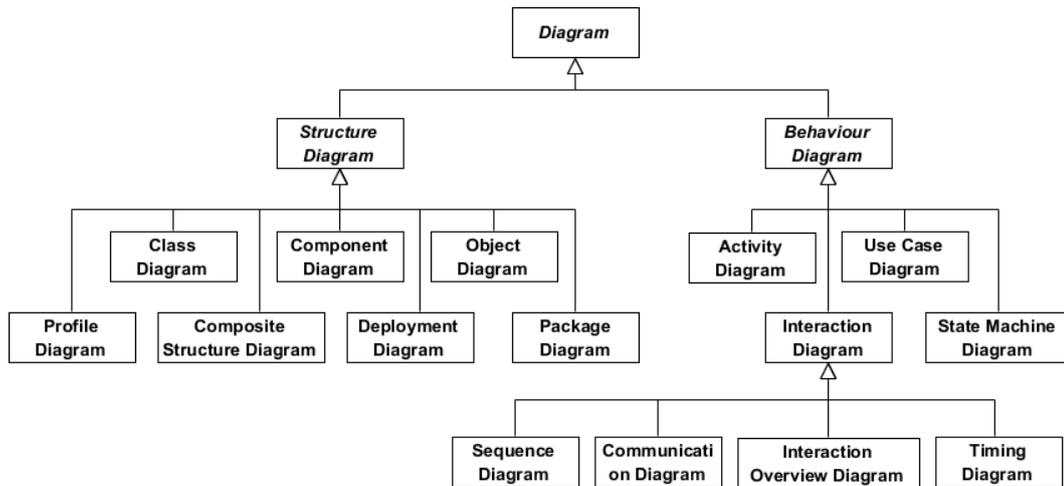
Gambar 2.6 Pengembangan Metode Air Terjun atau *Waterfall*

## 2.9 Rancangan Model *Unified Modelling Language* (UML)

*Unified Modelling Language* atau UML adalah suatu rancangan kebutuhan yang bermanfaat untuk mengelola kompleksitas sistem untuk meningkatkan produktifitas perangkat lunak pada suatu perusahaan atau industri. Dengan menggunakan UML, perusahaan dapat mengetahui apa saja yang menjadi kebutuhan serta meningkatkan kualitas serta mengurangi biaya dalam hal membuat suatu arsitektur kebutuhan. UML dirancang untuk menanggapi kebutuhan (*Visual Paradigm*, 2020).

Dalam *Unified Modelling Language* atau UML terdapat 14 model diagram yang dibagi menjadi 2 struktur yaitu *Structure Diagram* dan *Behavior Diagram*. Pada *Structure Diagram* terdapat 7 diagram yaitu *Class Diagram*, *Component Diagram*, *Deployment Diagram*, *Object Diagram*, *Package Diagram*, *Composite Structure Diagram*, dan *Profile Diagram*. Sedangkan pada *Behavior Diagram* terdapat 7 diagram yaitu *Use Case Diagram*, *Activity Diagram*, *State Machine Diagram*, *Sequence Diagram*, *Communication Diagram*, *Interaction Overview Diagram* dan *Timing Diagram* (*Visual Paradigm*, 2020).

Berikut alur diagram dari *Unified Modelling Language* atau UML pada gambar 2.7.



Gambar 2.7 Alur Diagram UML

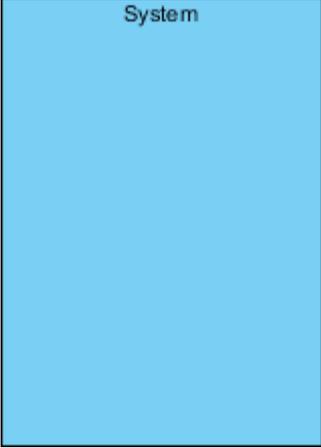
### 2.9.1 Rancangan *Use Case Diagram*

*Use Case Diagram* digunakan dalam tahap awal pengembangan perangkat lunak. Dimana diagram tersebut akan menjelaskan secara keseluruhan tentang aktifitas utama yang terjadi dalam perangkat lunak (*Visual Paradigm, 2020*). Tujuan dari rancangan *use case diagram* sebagai berikut :

- 1) Menentukan konteks dari suatu sistem
- 2) Mendapatkan persyaratan sistem
- 3) Validasi arsitektur sistem
- 4) Implementasi dan hasil dari kasus
- 5) Pengembangan dilakukan dengan cara analisis bersama dengan pakar

Tabel 2.2 Notasi dalam *Use Case Diagram*

Gambar	Nama Notasi	Keterangan
	Aktor atau <i>Actor</i>	Yang berinteraksi dalam <i>use case</i> . Dalam bentuk kata benda. Memiliki tanggung jawab dalam <i>input</i> dan <i>output</i> sistem

	<i>Use Case</i>	Proses yang terjadi dalam <i>use case</i>
	<i>Communication Link</i>	Garis yang menghubungkan partisipasi aktor dalam <i>use case</i>
	<i>Boundary of System</i>	Suatu sistem yang akan dijelaskan <i>use case diagram</i>

Sumber : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram>

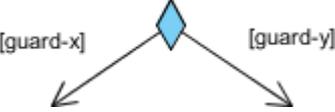
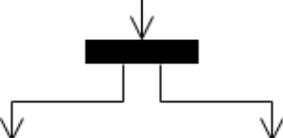
### 2.9.2 Rancangan *Activity Diagram*

*Activity Diagram* adalah suatu representasi grafis alur kerja kegiatan dan tindakan yang bersifat bertahap yang digunakan untuk menentukan pilihan, iterasi dan konkurensi. Diagram ini, menggambarkan aliran kontrol dalam suatu sistem. Selain itu, *Activity Diagram* dimaksudkan untuk proses komputasi dan organisasi (*Visual Paradigm*, 2020).

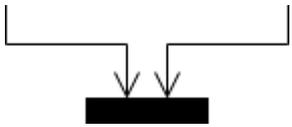
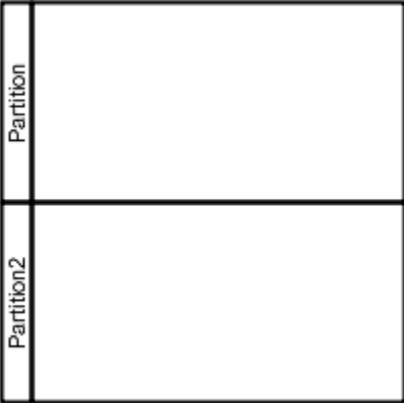
Tabel 2.3 Notasi dalam *Activity Diagram*

Gambar	Nama Notasi	Keterangan
	<i>Activity</i> atau Aktifitas	Digunakan untuk merepresentasikan suatu aksi

Tabel 2.3 Notasi dalam *Activity Diagram* (Lanjutan)

	<i>Action</i>	Proses atau aktifitas yang harus dilakukan
	<i>Control Flow</i> atau <i>Object Flow</i>	Garis yang memperlihatkan urutan eksekusi serta menampilkan aliran aktifitas dari satu aktifitas ke aktifitas lain
	<i>Initial Node</i>	Menggambarkan awal simbol dari suatu aktifitas dalam <i>activity diagram</i>
	<i>Activity Final Node</i>	Menghentikan seluruh aktifitas dalam <i>activity diagram</i>
	<i>Object Node</i>	Merepresentasikan sebuah objek yang terhubung dengan <i>Object Flows</i>
	<i>Decision Node</i>	Notasi yang mewakili kondisi pengujian yang memastikan bahwa aliran control atau aliran objek hanya berjalan satu arah
	<i>Merge Node</i>	Notasi yang menyatukan jalur keputusan yang berbeda menggunakan simpul keputusan
	<i>Fork Node</i>	Notasi yang membagi <i>behavior</i> menjadi serangkaian kegiatan paralel atau bersama

Tabel 2.3 Notasi dalam *Activity Diagram* (Lanjutan)

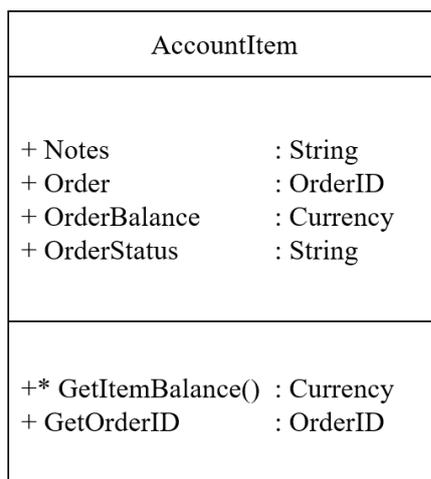
	<i>Join Node</i>	Notasi yang menyatukan kembali serangkaian aktifitas yang paralel atau bersamaan
	<i>Swimlane and Partition</i>	Notasi yang mengelompokkan aktivitas yang akan dilakukan oleh aktor pada suatu <i>activity diagram</i>

Sumber : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram>

### 2.9.3 Rancangan *Class Diagram*

*Class Diagram* adalah suatu diagram yang memetakan struktur sistem dengan memodelkan kelas, atribut, operasi serta hubungan antar objek. Selain itu, diagram ini menggambarkan serta mendeskripsikan tentang *class*, atribut, dan objek. *Class Diagram* bersifat statis karena *class diagram* tidak menggambarkan apa yang terjadi melainkan hubungan yang terjadi (Webstudi, 2019).

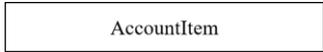
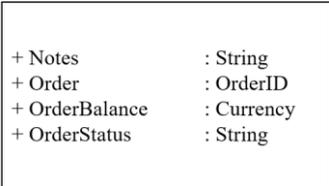
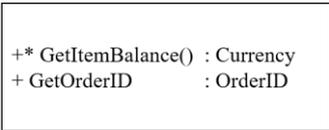
Dengan *Class Diagram* dapat memberikan informasi yang lebih mengenai sebuah sistem yang akan dibangun dengan menjelaskan kelas serta hubungan – hubungannya. Berikut bentuk dari *class diagram* yang dijelaskan pada gambar 2.8.



Gambar 2.8 Contoh *Class Diagram* dalam *AccountItem*

Berikut tabel yang menjelaskan notasi yang digunakan *Class Diagram* yang akan menjelaskan suatu kelas dalam sebuah sistem :

Tabel 2.4 Notasi dalam *Class Diagram*

Gambar	Nama Notasi	Keterangan
	<i>Name Class</i> atau Nama Kelas	Notasi ini berfungsi untuk membedakan dengan <i>class</i> yang lain
	<i>Attribute</i> atau Atribut	Notasi ini menjelaskan atribut – atribut yang ada didalam suatu <i>class diagram</i>
	<i>Method</i> atau Metode (Operasi Kelas)	Operasi ini menggambarkan suatu kelas yang akan berinteraksi dengan data pada atribut

Sumber : <http://www.webstudi.site/2019/02/Class-Diagram.html>