

## **2. TINJAUAN PUSTAKA**

### **2.1 *E-Learning***

#### **2.1.1 Pengertian *E-Learning***

Menurut Rusman, Deni, & Cepi (2012: 265), WBL merupakan salah satu bentuk *e-learning* yang materi maupun cara penyampaianya melalui internet (*web*). Melalui pengertian tersebut akan tercipta sebuah pembelajaran berbasis web. Pembelajaran berbasis *web* tersebut melakukan kegiatan belajar mengajar dengan memanfaatkan jaringan internet. *E-learning* berbasis *web* ini juga memberikan kesempatan para penggunanya untuk berkomunikasi dan bertukar informasi dimana saja dan kapan saja.

Menurut Rusman (2012: 293) *e-learning* merupakan segala aktivitas belajar yang menggunakan bantuan teknologi elektronik. Melalui *e-learning*, pemahaman siswa tentang sebuah materi tidak tergantung pada guru/instruktur tetapi dapat diperoleh dari media elektronik. Teknologi elektronik yang banyak digunakan misalnya internet, intranet, tape video atau audio, penyiaran melalui satelit, televisi interaktif serta CD-ROM (Rusman, 2012: 291).

#### **2.1.2 Manfaat *E-Learning***

Menurut Made Wena (2009: 213-214) manfaat *e-learning* dapat dikategorikan berdasar 3 sudut pandang, yaitu :

##### **1) Sudut pandang siswa**

E-learning dapat membuat aktivitas belajar siswa menjadi lebih fleksibel, siswa dapat mengakses pembelajaran setiap saat dan berulang-ulang. Selain itu siswa juga dapat berinteraksi dengan guru setiap saat, jadi ketika ada pertanyaan ataupun merasa kurang jelas siswa dapat langsung bertanya pada gurunya. Hal ini sangat berbeda dengan pembelajaran konvensional, di mana proses kegiatan belajar mengajar dilakukan harus tepat waktu dan tempatnya. Selain itu

menurut Seok (Lantip Diat Prasajo & Riyanto, 2011: 231) ada beberapa kelebihan *e-learning* dibanding pembelajaran konvensional apabila fasilitas infrastruktur yang mendukung terpenuhi menjangkau daerah pedesaan, yaitu:

- Sekolah-sekolah kecil di pedesaan dapat mengakses atau mempelajari mata pelajaran yang tidak diajarkan di sekolahnya
- Bagi siswa yang mengikuti *home schoolers* (program pendidikan keluarga) dapat mengikuti pembelajaran yang tidak dapat diajarkan orangtuanya seperti bahasa asing atau komputer.
- *E-learning* dapat diakses oleh siswa yang fobi (phobia) terhadap sekolah, siswa yang di rawat di rumah sakit atau di rumah, yang putus sekolah tapi berminat melanjutkan sekolahnya, siswa yang berada di luar daerah maupun di luar negeri.
- Siswa yang tidak tertampung di sekolah konvensional dapat memanfaatkan *e-learning* untuk mendapatkan pendidikan.

## 2) Sudut pandang guru/pendidik

Dari sudut pandang seorang pendidik menurut Soekartawi (Made Wena, 2009: 213) manfaat *e-learning* antara lain:

- Lebih mudah dalam melakukan kemutakhiran bahan-bahan belajar sesuai dengan perkembangan ilmu yang ada.
- Dapat mengembangkan diri atau melakukan penelitian dengan waktu luang lebih banyak.
- Mengontrol kebiasaan belajar siswa, guru dapat mengetahui kapan siswanya belajar.
- Mengecek siswa dalam mengerjakan soal-soal latihan.
- Memeriksa jawaban siswa dan memberitahukan hasil penilaian kepada siswa.

### 3) Sudut pandang sekolah

Bagi sekolah, *e-learning* memiliki beberapa manfaat antara lain:

- Akan tersedia bahan ajar yang telah divalidasi oleh ahli sesuai dengan bidangnya sehingga setiap guru dapat menggunakan dengan mudah, efektif, dan efisien.
- Pengembangan isi pembelajaran akan sesuai dengan berbagai pokok bahasan.
- Dapat menjadi pedoman praktis implementasi pembelajaran sesuai kondisi dan karakteristik pembelajaran.
- Menumbuhkan sikap kerjasama baik antara sesama guru maupun antara guru dengan siswa dalam memecahkan masalah pembelajaran.

## 2.2 Internet

Menurut Sibero(2011), internet merupakan Interconnected Network, yaitu sebuah jaringan komputer yang saling menghubungkan antar komputer secara global. Internet mampu bekerja karena adanya jaringan komputer, baik jaringan lokal maupun jaringan dengan jangkauan global yang lebih luas. Selain itu, internet juga memakai protokol komunikasi yang sama, yaitu IP atau TCP.

Menurut Berners Lee, internet yaitu sebuah jaringan yang memiliki beberapa jaringan di dalamnya. Konsep dari Berners Lee tersebut dapat diartikan bahwa adanya jaringan komputer lokal yang terhubung dengan jaringan lainnya.

## 2.3 World Wide Web (WWW)

### 2.3.1 Pengertian WWW

Menurut Gregorius(2000), *website* adalah kumpulan halaman *web* yang saling terhubung dan seluruh *file* saling terkait. *Web* terdiri dari *page* atau halaman dan

kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas dengan halaman-halaman terkait berada di bawahnya. Biasanya, setiap halaman di bawah *homepage* (*child page*) berisi *hyperlink* ke halaman lain dalam *web*.

*Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi (Sholechul Azis, 2013).

### 2.3.2 Jenis-jenis *Website*

Pernyataan dari sumber katadata.co.id, berdasarkan buku CMM Website Interaktif MCMS Joomla(CMS), jenis *website* dibagi berdasarkan sifat adalah:

- 1) *Website* dinamis, yaitu sebuah *website* yang menyediakan konten atau isi yang selalu berubah setiap saat. Contoh website dinamis adalah media berita daring.
- 2) *Website* statis, merupakan website yang kontennya sangat jarang diubah. Misalnya, profil organisasi dan sebagainya.

## 2.4 HTML 5

HTML atau Hyper-Text Markup Language merupakan bahasa *markup* standar dalam pembuatan website. Umumnya, penggunaan HTML adalah untuk menyusun bagian paragraf, heading, maupun link pada halaman web. Tapi, meskipun susunannya seperti coding, HTML bukanlah bahasa pemrograman.

HTML tidak dianggap sebagai bahasa pemrograman karena tidak bisa memberikan fungsi yang dinamis. Sekarang, bahasa markup ini dianggap sebagai standar web resmi, dikelola oleh World Wide Web Consortium (W3C), yang juga bertugas merilis pembaruan rutin HTML.

### 2.4.1 Kelebihan dan kekurangan HTML

Kelebihan yang ada pada bahasa *markup* HTML :

- Digunakan secara luas dan memiliki banyak sumber serta komunitas yang besar. Dijalankan secara alami di setiap *web browser*.
- Mudah dipelajari.
- *Open-source* dan sepenuhnya gratis.
- Rapi dan konsisten.
- Menjadi standar resmi *web*, dikelola oleh (W3C).
- Integrasi mudah dengan bahasa *backend*, seperti PHP dan Node.js

Kekurangan yang ada pada bahasa *markup* HTML :

- Umumnya digunakan untuk halaman *web* statis. Untuk fitur dinamis, perlu menggunakan JavaScript atau bahasa *backend*, seperti PHP.
- Tidak bisa menjalankan *logic*. Alhasil, semua halaman *web* harus dibuat terpisah meskipun menggunakan elemen yang sama, seperti *header* dan *footer*.
- Fitur baru tidak bisa digunakan dengan cepat di sebagian *browser*.
- Perilaku *browser* terkadang sulit diprediksi. Misalnya, *browser* lama tidak selalu bisa me-render *tag* yang lebih baru.

### 2.4.2 Tag HTML

Tag HTML yang lazim dipakai pada HTML 5, yaitu:

*Table 2.1 Tag HTML*

TAG	KETERANGAN
<!DOCTYPE>	Menentukan tipe dokumen

<html>	Membuat sebuah dokumen HTML
<head>	Membuat informasi pada dokumen
<meta>	Membuat metadata pada dokumen
<title>	Membuat judul dari halaman
<body>	Membuat isi konten pada halaman
<h1> to <h6>	Membuat heading
<p>	Membuat paragraf
 	Membuat baris baru pada halaman
<hr>	Membuat garis lurus panjang secara horizontal
<!--...-->	Membuat komentar
<b>	Membuat huruf cetak tebal
<form>	Membuat form pada HTML
<input>	Membuat kolom input pada form
<textarea>	Membuat input teks dengan banyak baris
<button>	Membuat tombol
<select>	Membuat input pilihan ( <i>dropdown</i> )
<option>	Membuat daftar pilihan dari tag <select>
<label>	Membuat label dari sebuah elemen <input>
<iframe>	Membuat jendela baru yang berisikan konten web lain
<img>	Menampilkan gambar
<a>	Membuat <i>hyperlink</i>
<ul>	Membuat urutan list dengan simbol atau karakter spesial pada HTML
<ol>	Membuat urutan list dengan nomor, abjad, dan angka romawi
<li>	Membuat sebuah item daftar
<table>	Membuat <i>table</i>
<th>	Membuat sel <i>header</i> pada <i>table</i>

<tr>	Membuat baris pada <i>table</i>
<td>	Membuat sel dalam <i>table</i>
<style>	Membuat style atau desain pada halaman web
<div>	Mengelompokkan tag atau elemen ( <i>block-line</i> )
<span>	Mengelompokkan tag atau elemen ( <i>inline</i> )
<script>	Membuat <i>script</i> program pada <i>client side</i>

## 2.5 PHP

PHP atau singkatan dari Hypertext Preprocessor, merupakan bahasa pemrograman yang berjalan pada *server side (backend)* atau sisi server. Bahasa pemrograman tersebut juga termasuk salah satu bahasa *scripting*. Bahasa *scripting* adalah bahasa yang mengotomatiskan eksekusi *task* (tugas) dalam *environment runtime* khusus. Tugas ini mencakup menginstruksikan halaman statis (dibuat dengan HTML dan CSS) untuk melakukan tindakan tertentu dengan aturan yang sudah ditetapkan.

Seperti, kita dapat menggunakan script untuk memvalidasi apakah semua kolom dalam sebuah *form* sudah diisi sebelum *form* dikirim kembali ke server. *Script* tersebut akan berjalan, lalu memeriksa semua kolom ketika user mengirimkan form. Jika ada kolom yang masih kosong, akan muncul teks peringatan untuk memberi tahu user tentangnya.

### 2.5.1 Kelebihan PHP

Kelebihan bahasa pemrograman PHP, yaitu :

- Mudah dipelajari – Dapat mempelajari PHP dengan mudah karena tersedia dokumentasi lengkap tentang fungsinya beserta contoh.
- Banyak digunakan – PHP umum digunakan untuk membuat berbagai jenis platform seperti *e-commerce*, blog, media sosial, dan lain-lain.

- Hemat biaya – PHP adalah bahasa pemrograman/skrip yang *open-source*, dengan kata lain bahasa tersebut dapat digunakan secara gratis.
- Ada banyak komunitasnya – Jika menjumpai masalah terkait PHP, banyak situs atau blog yang secara spesifik membahas PHP di internet.
- Terintegrasi dengan *database* – beberapa contoh program *database* yang terintegrasi dengan PHP adalah MySQL, Oracle, Sybase, DB2, dll.

### 2.5.2 PHP Native

PHP *Native* merupakan bahasa program PHP murni. Yang berarti program tersebut dibuat dari nol tanpa bantuan dari library eksternal maupun memiliki *styling* program sedari awal.

Walaupun dibuat dari nol, PHP *Native* atau PHP murni memiliki kelebihan tersendiri. Kelebihannya adalah program yang dapat dibaca karena program tersebut hanya bisa mudah dibaca oleh pembuat program tersebut.

Penggunaan PHP *Native* tersebut cukup diperuntukkan untuk *website* statis, yang berarti untuk pembuatan *website* yang jarang berubah dan tetap. Dan juga tidak cocok untuk penggunaan bisnis karena mudah sekali banyak melakukan perubahan *website*. Itulah kekurangan dari PHP *Native*.

## 2.6 Javascript

Javascript merupakan bahasa pemrograman yang berjalan pada *client-side* atau sisi klien dan juga termasuk bahasa *scripting*. Javascript digunakan oleh banyak programmer untuk pembuatan *website* yang sangat dinamis dan juga menampilkan interaksi kepada pengguna. Javascript juga merupakan bahasa pemrograman yang paling populer pada tahun 2022 bila dilihat berdasarkan *survey* pada *website* stackoverflow sebanyak 65,36% dari 71.547 responden.



Javascript adalah bahasa pemrograman dijalankan menggunakan *interpreter* dan dibantu dengan *web browser* seperti Google Chrome, Mozilla Firefox dan lain-lain. Pengkodingannya dibuat dengan menyisipkan *tag* `<script>` yang ada pada HTML atau bisa dipanggil secara eksternal dengan menaruh sumber direktori file Javascript tersebut.

### 2.6.1 Kelebihan Javascript

Kelebihan yang ada pada bahasa Javascript adalah:

- **Struktur yang sederhana.** Strukturnya yang sederhana membuat JavaScript lebih mudah dipelajari dan diterapkan, serta lebih cepat daripada sejumlah bahasa lain. Galat juga mudah diidentifikasi dan diperbaiki.
- **Eksekusi lebih cepat.** JavaScript mengeksekusi *script* langsung di *browser web* tanpa harus terhubung ke *server* atau menggunakan *compiler*. Selain itu, sebagian besar *browser* memungkinkan JavaScript meng-*compile* kode pada saat eksekusi program.
- **Fungsi yang serbaguna.** JavaScript kompatibel dengan bahasa pemrograman lain seperti PHP, Perl, dan Java. Bahasa ini juga menjadikan *data science* dan *machine learning* bisa diakses oleh developer.
- **Cukup populer dan banyak digunakan.** Ada berbagai sumber dan forum yang bisa membantu pemula mempelajari bahasa *scripting* ini.
- **Mengurangi beban server.** Karena berjalan di sisi klien, JavaScript bisa mengurangi permintaan yang dikirim ke *server*. Validasi data bisa dilakukan melalui *browser*, dan update hanya berlaku pada bagian halaman *web* tertentu.
- **Selalu diperbarui dan dikembangkan.** Tim *developer* JavaScript dan ECMA International terus memperbarui serta merancang *framework* dan *library* baru sehingga bisa terus relevan.

## 2.7 Bootstrap

### 2.7.1 CSS

CSS atau singkatan dari Cascading Style Sheet merupakan bahasa *markup* yang berfungsi untuk membuat desain tampilan pada website agar menjadi rapi dan menarik. Tidak hanya itu, CSS dapat membuat animasi sederhana dan tampilan gerak bila dipadukan dengan bahasa Javascript sebagai interaksinya.

Bahasa *markup* CSS tersebut dibuat oleh Hakon Wium Lie. Hakon lahir pada tahun 1965 di Norwegia. Dia juga pernah bekerja bersama Tim Berners-Lee di CERN kemudian mengusulkan CSS pada tahun 1999. Dan pada saat itu juga menjadi CTO Opera hingga kini.

### 2.7.3 Framework Bootstrap

*Framework* Bootstrap merupakan *framework* CSS yang berfungsi untuk mendesain *website responsive* dengan cepat dan mudah. *Framework* Bootstrap dibuat oleh Mark Otto dan Jacob Thornton dari Twitter pada tahun 2011. Framework ini berbasis *open-source* yang berarti dapat dipakai siapa saja secara gratis.

### 2.7.4 Kelebihan Framework Bootstrap

Kelebihan yang ada pada *Framework* Bootstrap, yaitu :

- **Ramah untuk pemula.** *Framework* ini dibuat dengan mengakses elemen dan class yang sudah siap pakai. Dan juga menyediakan *component design* yang lengkap lalu tinggal melakukan *copy paste* pada pembuatan program *web*.
- **Grid style yang canggih.** Semua elemen di dalam website bisa dibuat menyesuaikan dengan perangkat yang digunakan pengunjung. Artinya, tampilan Bootstrap dapat dimuat baik PC maupun *mobile*.

- **Kompatibilitas dengan *web browser* terbaru.** Bootstrap dapat mendukung semua *web browser* versi terbaru seperti Firefox, Google Chrome, dan Safari.
- **Bersifat *open-source*.** Bootstrap merupakan *framework* gratis yang memerlukan biaya apapun. Jadi, Bootstrap dapat diakses maupun digunakan oleh siapa saja.
- **Kebebasan kustomisasi.** Artinya *framework* ini dapat diubah sesuai selera pengguna.
- **Rutin diperbaharui.**
- **Memiliki dokumentasi yang lengkap.** Bootstrap memiliki panduan yang lengkap mengenai pengunduhan, pemasangan serta penggunaan *framework* tersebut.
- **Memiliki komunitas yang besar.** Bila memiliki kendala akan penggunaan *framework* tersebut, dapat ditanyakan melalui forum stackoverflow.

## 2.8 Codemirror

Codemirror adalah teks *editor* serbaguna yang diimplementasikan dalam bahasa Javascript. Diperuntukkan untuk mengedit *code*, dan dilengkapi dengan sejumlah *mode* bahasa dan fitur tambahan yang mengimplementasikan fungsi pengeditan yang canggih. Codemirror merupakan komponen *code editor* yang disematkan pada halaman *web*.

## 2.9 Database

### 2.9.1 Pengertian Database

Menurut C.J. Date, *database* merupakan koleksi data operasional yang sengaja disimpan dan juga dipakai oleh sistem aplikasi dari suatu organisasi. Terdapat 3 jenis data yang disimpan dalam *database* yaitu data *input*, *output* dan operasional.

- Data *input* yakni data yang masuk dari luar sistem.
- Data *output* yakni data yang dihasilkan oleh sistem.
- Data operasional yakni data yang tersimpan pada sistem.

Menurut Anhar (2010:45), *database* atau basis data adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau *kolom*. Struktur *file* yang menyusun sebuah *database* adalah *Data Record* dan *Field*.

### 2.9.2 Perintah MySQL

Perintah dalam MySQL dibagi menjadi 3 bagian :

- **DDL (*Data Definition Language*)**. Digunakan untuk menentukan skema dari *database* dan memodifikasi struktur objek dari *database* di dalam *database* yang ada. Perintah yang termasuk pada DDL adalah : *Create*, *Drop*, *Alter* dan lain-lain.
- **DML (*Data Manipulation Language*)**. Digunakan untuk mengambil dan memanipulasi data dalam *database*. Perintah yang termasuk pada DML adalah : *Insert*, *Update*, *Delete*.
- **DCL (*Data Control Language*)**. Digunakan untuk pengontrolan data yang berhubungan dengan izin, hak dan control lainnya pada sistem *database*. Perintah yang termasuk pada DCL adalah : *Grant* dan *Revoke*.

### 2.9.3 MySQL

Menurut Arief (2011d:152) “MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya”.

### 2.10 Visual Studio Code

VS Code merupakan *text editor* buatan Microsoft. *Text editor* ini bersifat *open source*, dengan kata lain dapat digunakan oleh siapapun dan gratis.

VS Code mempunyai fitur yang lengkap dalam penulisan *code*. Bila dirasa kurang memuaskan, fitur tambahan dapat dilakukan dengan mengunduh dan memasang pada menu Extension.

## 2.11 XAMPP

XAMPP merupakan *software web server* yang bersifat *open-source*. Kegunaan XAMPP yaitu menjalankan *web server* itu sendiri secara *local* dan *offline*.

Fungsi yang ada pada XAMPP, ialah:

- Memiliki penerjemah bahasa pemrograman yaitu PHP dan Perl
- Memiliki DBMS (*Database Management System*) yaitu MySQL dan PHPMyAdmin
- Fungsi tersedia lainnya yaitu Tomcat, Filezilla dan lain-lain

## 2.12 *Software Development Life Cycle*

Ian Sommerville (2011) menjelaskan bahwa ada lima tahapan pada model *waterfall*, yaitu *Requirements Analysis and Definition*, *Sytem and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operationa and Maintenance*. Tahapan pada SDLC tersebut, yaitu :

1. ***Requirement Analysis***. Pada tahap ini mengumpulkan informasi yang dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

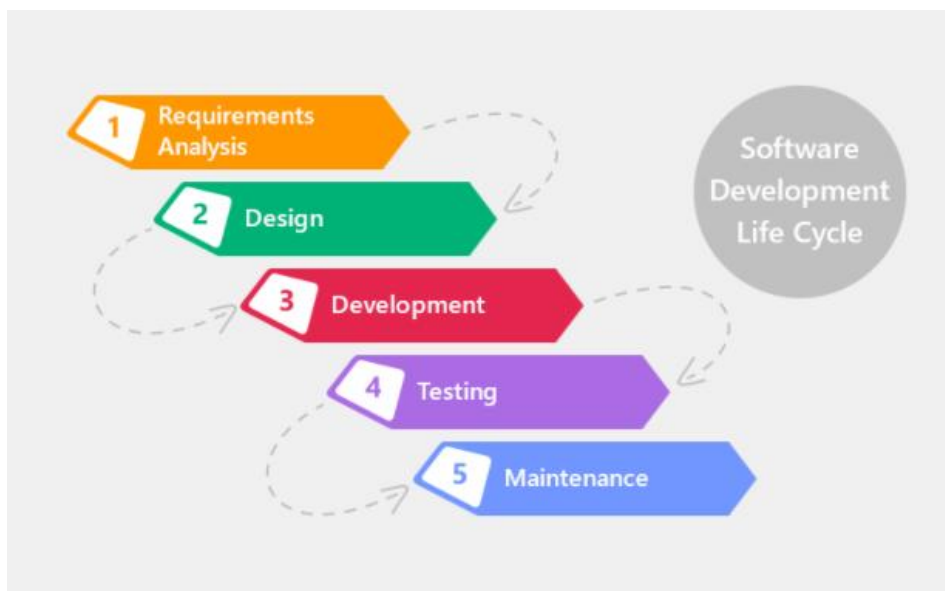
2. ***System and Software Design.*** Meneruskan dari tahapan pertama, pada tahap ini menyusun sebuah perancangan desain untuk dapat memberikan gambaran lengkap mengenai apa yang harus dikerjakan. Tahapan ini juga akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.
3. ***Implementation and Unit Testing.*** Tahapan SDLC untuk membuat model cara kerja aplikasi. Ada beberapa aspek yang perlu diperhatikan pada tahap ini, yaitu:
4. ***Integration and System Testing.*** Setelah seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.
5. ***Operation and Maintenance.*** Pada tahap terakhir dalam metode *waterfall*, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan 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.

### **2.12.1 Model Waterfall**

Model *waterfall* merupakan metode pengembangan perangkat lunak tertua sebab sifatnya yang natural. Metode Waterfall merupakan pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak. Urutan dalam model

*waterfall* bersifat serial yang dimulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem.

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Langkah-langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh sebab itu disebut *waterfall* (air terjun).



Gambar 2.1 Model Waterfall

## 2.13 Unified Modeling Language

### 2.13.1 Pengertian UML

Menurut Yadanur (2012), UML merupakan bahasa standar yang digunakan untuk membangun dan memvisualisasi sistem *software*.

Menurut Rosa Dan Shalahuddin (2015), UML adalah salah satu bahasa dasar yang banyak digunakan dalam dunia industri guna mendefinisikan kebutuhan, membuat desain dan analisis, dan menggambarkan arsitektur dalam pemrograman yang berorientasikan pada objek.





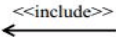
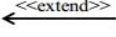
Menurut Joomla, UML merupakan *standard modeling language* yang terdiri dari kumpulan diagram-diagram, dikembangkan untuk membantu para pengembang sistem dan perangkat lunak agar bisa menyelesaikan tugas-tugas tertentu, seperti spesifikasi, visualisasi, desain arsitektur, konstruksi, simulasi, *testing*, dan dokumentasi.

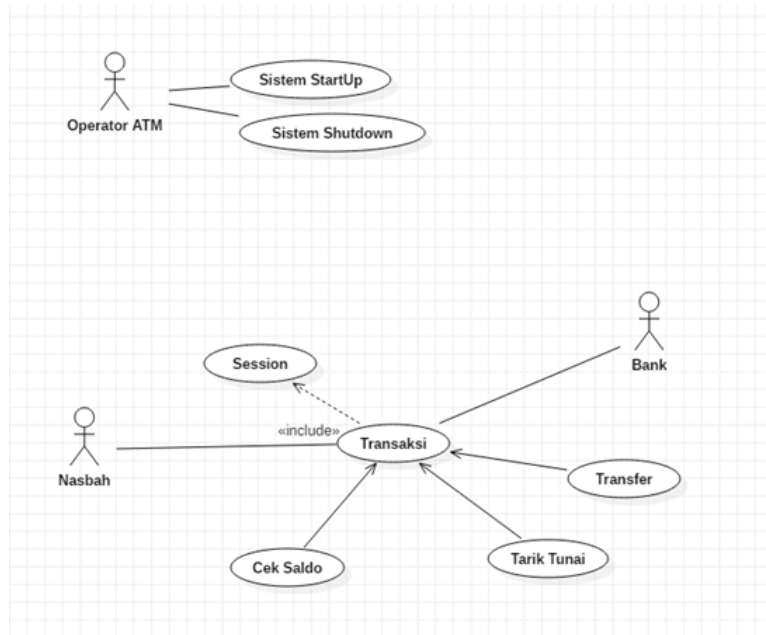
### **2.13.2 Use Case Diagram**

*Use case* adalah komponen gambaran fungsional dalam sebuah sistem. Sehingga konsumen maupun pembuat saling mengenal dan mengerti mengenai alur sistem yang akan dibuat.



Table 2.2 Simbol Use Case

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	Use case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi









Gambar 2.2 Use Case pada ATM

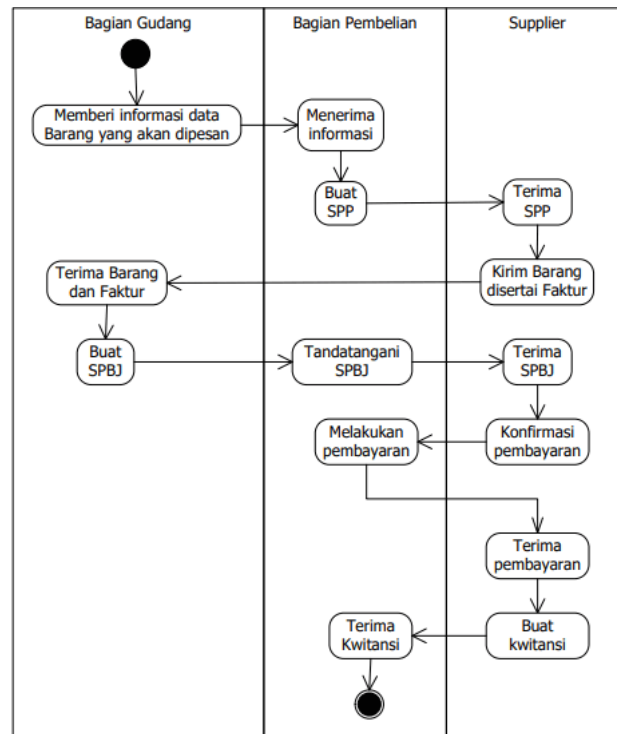
### 2.13.3 Activity Diagram

*Activity diagram*, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari Use Case yang memiliki alur aktivitas.

Alur atau aktivitas berupa bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. Dalam buku Rekayasa Perangkat Lunak karangan Rosa A.S mengatakan, “Diagram aktivitas tidak menjelaskan kelakuan aktor. Dapat diartikan bahwa dalam pembuatan *activity diagram* hanya dapat dipakai untuk menggambarkan alur kerja atau aktivitas sistem saja.”

Table 2.3 Simbol Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi



Gambar 2.3 Activity Diagram Penjualan

## 2.14 Struktur Navigasi

Alur dari sebuah program termasuk terpenting dalam pembuatan aplikasi halaman *web* dan gambaranya harus ada pada tahap perencanaan. Menentukan struktur navigasi merupakan halaman yang sebaiknya dilakukan sebelum membuat *website*.

Beberapa dasar struktur pembuatan halaman *web* yaitu linier, non linier, hirarki dan campuran.

### 2.14.1 Navigasi Linier

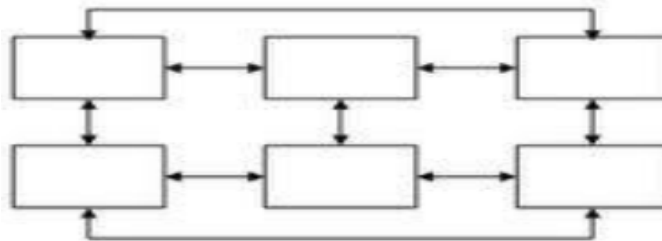
Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu rangkaian cerita yang berurut menurut urutannya.



*Gambar 2.4 Navigasi Linier*

### 2.14.2 Navigasi Non-Linier

Struktur navigasi Non-linier merupakan pengembangan dari struktur navigasi linier. Percabangan pada struktur non linier ini berbeda dengan percabangan pada struktur hirarki. Pada percabangan ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.

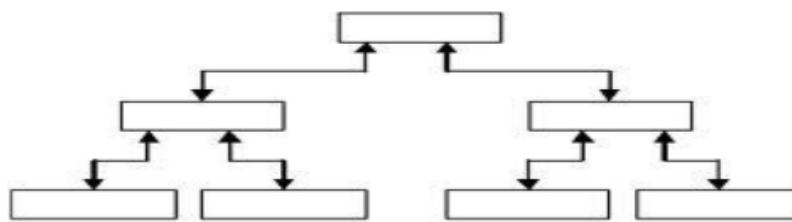


*Gambar 2.5 Navigasi Non-Linier*

### 2.14.3 Navigasi Hirarki

Struktur navigasi hirarki biasanya disebut struktur bercabang, merupakan struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu.

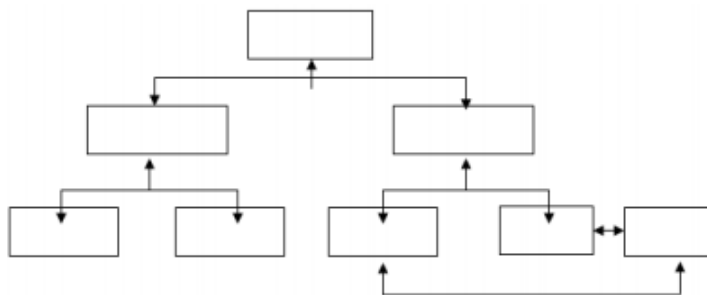
Tampilan pada menu satu akan disebut sebagai *Master Page* (halaman utama pertama), halaman ini mempunyai halaman percabangan yang disebut *slave page* (halaman pendukung).



Gambar 2.6 Navigasi Hirarki

### 2.14.4 Navigasi Campuran

Struktur navigasi *composite* (campuran) disebut juga struktur navigasi bebas yang merupakan gabungan dari ketiga struktur yang ada. Struktur navigasi ini biasa digunakan dalam pembuatan *multi media* karena dapat memberikan keinteraksian yang lebih tinggi.



Gambar 2.7 Navigasi Campuran