

BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

Dalam penulisan skripsi ini penulis menjelaskan tentang landasan teori terlebih dahulu sehingga penulis mendapatkan alur perancangan secara keseluruhan dari penelitian ini.

2.1.1 Perancangan Sistem

Prosedur untuk menyusun sebuah sistem guna memberikan kepuasan akan spesifikasi kebutuhan fungsionalitas, memenuhi kebutuhan secara tersirat atau jelas dari sisi performa maupun penggunaan sumber daya, memenuhi target, batasan kepuasan pada proses desain dari sisi pembiayaan, waktu, dan perangkat terhadap penggunaanya disebut juga dengan perancangan sistem.

Perancangan sistem merupakan sebuah penetapan proses data yang urgensi bagi sistem baru, dan fase-fase dalam merancang sistem, meliputi :

- a. Menyediakan rancangan sistem yang terdetail.
- b. Mengenali bermacam alternatif bentuk sistem.
- c. Mengevaluasi berbagai alternatif komposisi sistem.
- d. Mencocokkan komposisi terbaik.

(Rahmawati, 2019).

2.1.2 Sistem

Sistem yaitu gabungan-gabungan yang terdiri dari bagian-bagian seperti sumber daya manusia, jaringan kerja dari langkah-langkah yang saling berhubungan, data, teknologi seperti perangkat keras maupun perangkat lunak yang saling terintegrasi utuh guna tercapainya tujuan atau target tertentu yang sama (Julianti et al., 2019).

2.1.3 Informasi

Menurut Davis (Alpiandi, 2016), informasi merupakan data yang sudah diproses menjadi sebuah wujud yang bermanfaat untuk si penerima dalam pengambilan keputusan pada saat ini atau di masa mendatang.

Informasi memiliki sumber yaitu data. Data merupakan fakta yang mendeskripsikan suatu insiden-insiden dan integrasi bukti-bukti. Data merupakan bentuk yang belum diolah atau mentah sehingga perlu diproses lebih jelas lagi agar menjadi berfaedah bagi pemakainya. Kualitas informasi tergantung pada beberapa hal yaitu:

- a. *Accuracy*, berarti informasi harus terhalang dari keraguan.
- b. *On time*, berarti informasi yang diterima harus tepat waktu.
- c. *Relevant*, berarti informasi harus memiliki faedah bagi si penerima.

(Pamungkas, 2017).

2.1.4 Sistem Informasi

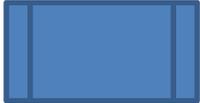
Menurut Ayu and Perdana (Wahyudi et al., 2019) Sistem informasi yaitu gabungan elemen-elemen yang berguna untuk mengumpulkan, memproses, dan meny informasi untuk mencapai hasil tujuan tertentu. Sistem informasi terdiri dari enam komponen sumber daya, yaitu *procedure*(prosedur), *database* (basis data) *network* (jaringan), *hardware* (perangkat keras), *software* (perangkat lunak) dan sumber daya manusia. Dengan kehadiran sistem informasi maka manusia sangat mudah untuk mendapatkan informasi di media elektronik khususnya *website*. Adapun contoh dari sistem informasi adalah seperti sistem informasi akademik yang memberi informasi kegiatan sekolah maupun perkuliahan, sistem manajemen yang berguna untuk monitoring kerja karyawan, sistem informasi akuntansi yang berguna untuk memberi informasi keuangan perusahaan dan sistem informasi lainnya.

2.1.5 Aliran Sistem Informasi Yang Baru

Disebut juga denah alir sistem merupakan denah yang mendeskripsikan aliran pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan rentetan-rentetan dari prosedur yang terdapat di sistem dan menunjukkan apa yang diatur oleh sistem (Jaya, 2016).

Berikut tabel dibawah ini simbol-simbol yang digunakan dalam system flow

Tabel 2. 1 Simbol Sistem flow

NO	SIMBOL	NAMA	FUNGSI
1		Terminator	Permulaan dan akhir program
2		Garis Alir	Garis penghubung antar terminator
3		Preparation	Proses
4		Proses	Proses mengolah data
5		Input/Output Data	Proses Masukan/Keluaran data parameter
6		Predefined Process (Sub Program)	Proses awal Subprogram/Proses menjalankan Subprogram
7		Decision	Proses yang memberi keputusan antara terminator
8		On Page Connector	Penghubung bagian-bagian Flowchart yang berada

			pada satu halaman
		Off page Connector	Penghubung bagian-bagian Flowchart yang berada pada halaman berbeda

(Sumber: Data Penelitian, 2020)

2.1.6 SDLC (*System Development Life Cycle*)

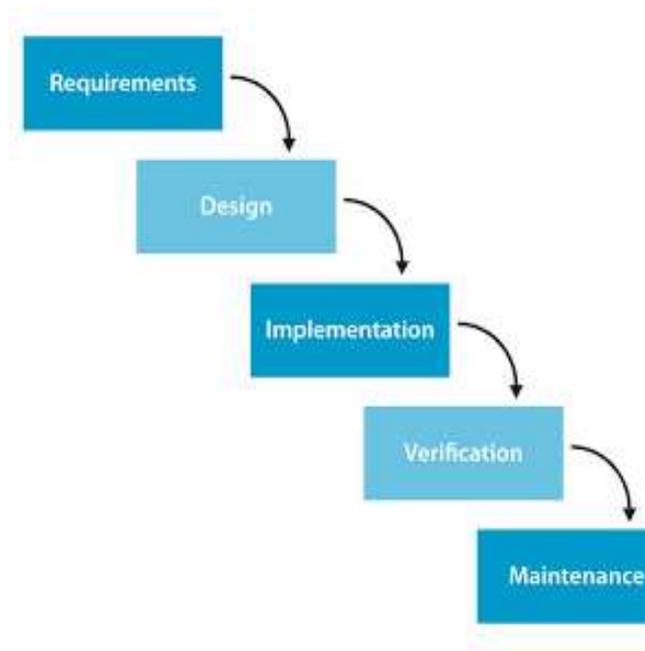
Software Development Life Cycle (SDLC) merupakan proses mengembangkan dan mengubah suatu sistem perangkat lunak dengan menggunakan tahapan-tahapan dan metodologi yang dipakai oleh pengembang untuk mengembangkan sistem-sistem perangkat lunak terdahulu. Dalam proses pengembangan sistem perangkat lunak, SDLC memiliki berbagai tahapan antara lain tahapan Prototype, tahapan Waterfall, tahapan Iteratif, tahapan RAD dan tahapan Spiral (Jinteks et al., 2019).

2.1.7 Model Waterfall

Model *Waterfall* merupakan model linear yang memiliki langkah-langkah pengembangan system yang terstruktur yaitu *Requirements*, *Design*, *Implementation*, *Verification*, dan *Maintenance*. Model ini juga sering digunakan

dalam pembangunan suatu sistem atau perangkat lunak yang bersifat umum (Masykur, 2014).

Gambar dibawah merupakan tahapan dari model *Waterfall*



Gambar 2. 1 Model Waterfall

(Sumber: Data Penelitian, 2014)

2.1.8 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) merupakan salah satu standar bahasa yang selalu digunakan di dunia sistem informasi untuk menentukan hasil gambaran, rancangan, mendokumentasikan artifak (bagian dari informasi yang dipakai dalam suatu proses perakitan *software*) yang diinginkan, menggambarkan

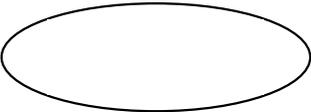
arsitektur didalam pemrograman berorientasi objek serta membuat analisis & desain (Nugraha et al., 2018).

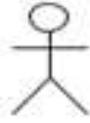
UML terbagi dari banyak komponen-komponen grafis yang disatukan ke bentuk-bentuk diagram. Tujuan representasi komponen-komponen grafis ke dalam diagram yaitu untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Gabungan dari beragam perspektif inilah yang disebut dengan sebuah model. Adapun diagram-diagram yang terdapat pada UML yaitu: *Use Case diagram*, *Activity Diagram*, *Sequence diagram*, dan *Class Diagram*. Adapun penjelasan dari diagram-diagram tersebut yaitu:

1. *Use Case*

Use case mendeskripsikan tugas-tugas sistem dari perspektif pengguna eksternal dan dalam sebuah metode yang gampang dimengerti. *Use Case* diagram adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal pengguna. Berikut contoh gambar *Use Case*:

Tabel 2. 2 Simbol *Use Case*

Lambang	Nama	Keterangan
<p data-bbox="336 1720 443 1749"><i>Use Case</i></p> 	<p data-bbox="805 1709 874 1816"><i>Use Case</i></p>	<p data-bbox="938 1675 1337 1854">Menggambarkan elemen-elemen yang saling bertukar pesan antar aktor</p>
<p data-bbox="336 1962 475 1991"><i>Aktor /Actor</i></p>	<p data-bbox="805 1921 884 1951">Aktor</p>	<p data-bbox="938 1906 1337 2007">Orang, atau proses sistem lain yang berinteraksi dengan</p>

		sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri
<p>Relasi Asosiasi/<i>Association</i></p> 	Relasi Asosiasi	Relasi yang menghubungkan antara <i>Use Case</i> dan aktor
<p>Relasi Ekstensi /<i>extend</i></p> <p><<<i>extend</i>>></p> 	Relasi Ekstensi	Suatu <i>Use Case</i> secara optimal menggunakan fungsionalitas yang disediakan oleh <i>Use Case</i> lainnya

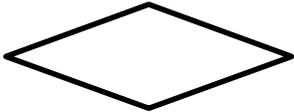
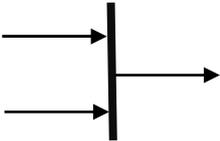
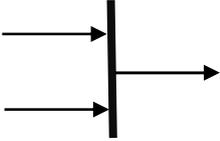
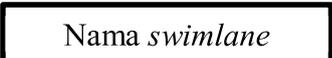
(Sumber: Data Penelitian, 2018)

2. Activity Diagram

Activity diagram menggambarkan aliran kerja atau kegiatan dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram

aktivitas menggambarkan aktivitas sistem bukan apa yang dikerjakan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut contoh gambar *Activity diagram*:

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Deskripsi
status awal 	Permulaan dari aktifitas sistem
aktivitas 	Aktivitas yang dikerjakan oleh aktor
percabangan / <i>decision</i> 	Opsi untuk mengambilkan keputusan
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana jika ada pilihan aktivitas digabungkan menjadi satu
Percabangan / <i>fork</i> 	Digunakan untuk menunjukkan kegiatan secara paralel.
Status akhir 	Status akhir diagram aktivitas
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang

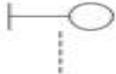
	terjadi
--	---------

(Sumber: Data Penelitian, 2018)

3. *Sequence diagram*

Sequence diagram merupakan suatu diagram yang digunakan untuk menggambarkan perilaku pada sebuah skenario dan mendeskripsikan bagaimana wujud dan sistem saling terhubung. Berikut adalah simbol gambar dari *Sequence diagram*:

Tabel 2. 4 Simbol Sequence Diagram

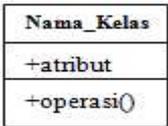
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Boundary Lifeline</i>	Menggambarkan hubungan suatu elemen yang berbeda, secara khas merupakan penghubung <i>actor</i> dengan layar.
2		<i>Entity Lifeline</i>	Menggambarkan suatu tempat atau mekanisme yang menangkap pengetahuan atau informasi dalam suatu sistem
3		<i>Control Lifeline</i>	Menggambarkan suatu pengendalian yang mengorganisir dan menjadwalkan aktivitas elemen-elemen.
4		<i>Message</i>	Perilaku sistem yang menandai adanya suatu alur informasi atau transisi kendali antar elemen.
5		<i>Actor</i>	Menunjukkan seorang pemakai sistem yang memulai alur peristiwa/kejadian.
6		<i>Activation bar</i>	Menggambarkan lamanya suatu pesan diproses.
7		<i>Note</i>	Menunjukkan catatan untuk komentar dari suatu pesan antar elemen.

(Sumber: Data Penelitian, 2018)

4. *Class Diagram*

Class Diagram merupakan kombinasi objek-objek yang mempunyai struktur umum, behavior umum, relasi umum, dan semantic/kata yang umum. *Class-class* ditentukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah class digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. Berikut adalah simbol-simbol dari *class diagram*:

Tabel 2. 5 Simbol Class Diagram

Gambar	Nama	Keterangan
<p>Kelas</p> 	Kelas/ <i>Class</i>	Kelas pada struktur sistem
<p>Antarmuka/<i>Interface</i></p>  <p>nama_interface</p>	Antarmuka	Serupa <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/<i>Association</i></p> 	Relasi Asosiasi	Relasi antar kelas, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p><i>Association Directed</i></p> 	Asosiasi Berarah/ <i>Directed</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

<i>Generalisation</i> 	Generalisasi/ <i>Generalisation</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi
<i>Dependency</i> 	Kebergantungan/ <i>Dependency</i>	Ketergantungan antarkelas
<i>Aggregation</i> 	Agregasi/ <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian

(Sumber: Data Penelitian, 2018)

2.2 Teori Khusus

Teori ini berusaha menjelaskan kenyataan-kenyataan itu dalam keterkaitannya antara yang satu dengan yang lainnya, hanya dapat di aplikasikan suatu macam masalah. Berikut ini adalah teori khusus yang mendukung dari penelitian ini:

2.2.1 PPDB (Penerimaan Peserta Didik Baru)

Penerimaan Peserta Didik Baru (PPDB) merupakan tahap awal yang dilakukan pelaksana lembaga pendidikan yang dimulai dari tingkat dasar sampai tingkat tinggi. Berdasarkan Peraturan Menteri Pendidikan dan Kebudayaan nomor 44 Tahun 2019, BAB I Pasal 2. Tentang Penerimaan Peserta Didik Baru pada

Taman Kanak-Kanak, Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas dan Sekolah Menengah Kejuruan, PPDB dilakukan berdasarkan: a. nondiskriminatif; b. objektif; c. transparan; d. akuntabel; dan e. berkeadilan. (2) Nondiskriminatif sebagaimana dimaksud pada ayat (1) dikecualikan bagi Sekolah yang secara khusus melayani peserta didik dari kelompok gender atau agama tertentu (Hidayat et al., 2020).

2.2.2 Website

Website atau akrab disebut sebagai situs *Web* merupakan peramban yang berisi sumber informasi saling terhubung, teks, gambar, gambar bergerak, video, atau macam-macam *file* lainnya. Sebuah situs *Web* biasanya diletakkan pada sebuah *server Web* yang dapat diakses melalui jaringan seperti media internet, ataupun jaringan wilayah lokal (LAN) melalui alamat internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai *World Wide Web* atau akrab dikenal dengan singkatan *WWW*. Untuk mengakses situs atau halaman *website* biasanya kita memerlukan *Web Browser* atau peramban (Harman et al., 2015).

2.2.3 PHPMyAdmin

PHPMyAdmin merupakan *software* yang ditulis secara bebas oleh pengguna dalam bahasa pemrograman PHP yang digunakan untuk mengelola administrasi MySQL melalui World Wide Web. PHPMyAdmin banyak men-

support berbagai operasi MySQL, diantaranya (pengolahan basis data, tabel-tabel, bidang (*fields*), hubungan antar relasi (*relation*), indeks, pengguna (*users*), kesepakatan (*deals*), dan lain-lain (Susilo, 2018).

2.2.4 PHP

PHP merupakan akronim dari *Hypertext Preprocessor* merupakan bahasa pemrograman yang biasanya digunakan khusus membuat atau mengembangkan suatu website dinamis. PHP juga terintegrasi dengan kode HTML, maksudnya adalah beda kondisi, HTML bertugas sebagai pondasi atau pembangun dari kerangka layout web, sedangkan PHP digunakan sebagai prosesnya, sehingga dengan adanya PHP tersebut, sebuah Web akan sangat mudah dikembangkan dan begitu juga dengan pemeliharannya (Hayat et al., 2019).



Gambar 2. 2 Logo PHP

(Sumber: <https://modernweb.com/wp-content/uploads/2015/02/PHP-logo.png>)

2.2.5 MySQL

MySQL merupakan program *software* atau perangkat lunak yang berfungsi sebagai penyimpanan data atau *Database Server*. Sedangkan *SQL* adalah bahasa *query* (permintaan) dalam *database server* termasuk dalam *MySQL* dan bahasa pemrogramannya itu sendiri. *SQL* juga dapat digunakan dalam *database server* lain seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya. *MySQL* Merupakan *database server* yang paling sering digunakan dalam pemrograman *PHP*. *MySQL* digunakan untuk proses transformasi data-data yang diperlukan berupa menambah, mengubah, dan menghapus data dan menyimpan data didalam *database*. Jadi dengan kesimpulan bahwa *MySQL* merupakan sebuah *software* atau program yang digunakan untuk memanipulasi data dan membuat *database* (Hayat et al., 2019).



Gambar 2. 3 Simbol MySQL

(Sumber: <https://www.MySQL.com/about/legal/logos.html>)

2.2.6 CSS (*Cascading Style Sheets*)

CSS atau akronim dari *Cascading Style Sheets* merupakan satuan atau gabungan kode yang yang tersinkron dalam *HTML* dan bertujuan untuk menggantikan gaya atau tampilan dari sebuah halaman *website*. CSS bisa juga digunakan untuk mengatur warna bagian tubuh pada ukuran border, teks, warna *hyperlink*, warna border, spasi antar paragraf, warna *mouse over*, ukuran gambar,

warna tabel spasi antar teks, margin kanan, margin kiri, margin bawah, margin atas dan bagian tubuh lainnya (Afisina et al., 2020).



Gambar 2. 4 Simbol CSS

(Sumber: <http://w3widgets.com/responsive-slider/img/css3.png>)

2.2.7 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) merupakan bahasa standar yang dipakai untuk menampilkan konten-konten seperti gambar, suara, video dan lainnya pada halaman *website*. Fungsi-fungsi yang dapat dilakukan dengan bahasa programan HTML adalah:

1. Mengatur serta mendesain tampilan isi halaman website,
2. Membuat tabel pada halaman website,
3. Menunjukkan halaman website secara daring,
4. Membuat rancangan formulir yang dapat menjadi masukkan serta menangani registrasi dan transaksi via website,
5. Menampilkan area gambar pada *browser*

(Mariko, 2019).



GAMBAR 2. 5 Logo Html

(Sumber: <https://www.w3.org/html/logo/>)

2.2.8 XAMPP

XAMPP atau kepanjangan dari *X Apache MySQL PHP Perl* merupakan *software* server yang berupa *local host* yang berdiri sendiri yang terintegrasi dengan program MySQL Database, Apache HTTP server, dan pengolah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan

webserver yang mudah dipakai dan dapat melayani tampilan halaman *web* yang dinamis (Soer & Wahyudi, 2015).



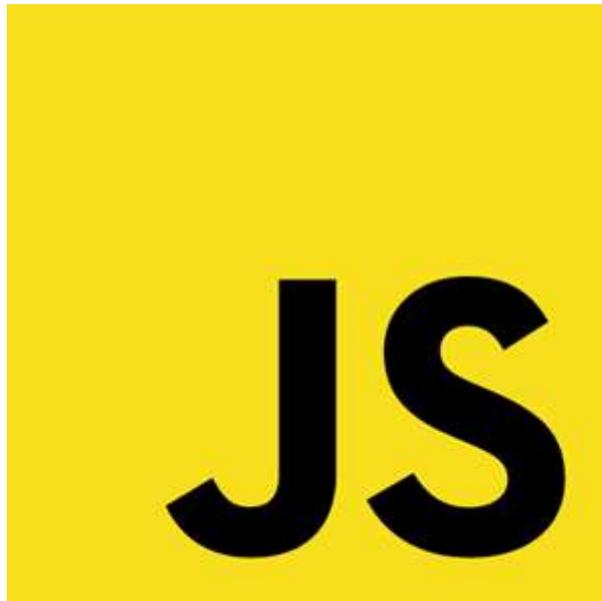
Gambar 2. 6 Logo XAMPP

(Sumber: <https://webhostmu.com/cara-instal-xampp/>)

5.4 Javascript

Menurut Lavarino & Yustanti (Mariko, 2019) Javascript merupakan bahasa pemrograman berisi kumpulan-kumpulan *script* yang berjalan pada suatu dokumen. Adapun karakteristik dari bahasa pemrograman Javascript adalah: (1) Bahasa pemrograman berjenis pemrograman level atas; (2) pengolahan data yang

ada di pemrogramannya bisa lakukan oleh pengguna atau disebut juga dengan *Client-Side*; (3) Berorientasi pada objek, dan (4) Bersifat *loosely typed*. Aplikasi atau *tools* standar yang digunakan dalam menjalankan Javascript adalah: (1) *Software text-editor*: seperti Notepad++, Adobe Dreamweaver dan NetBeans; (2) *Web browser*, seperti: Mozilla Firefox, Google Chrome, Internet Explorer, OperaMini, Safari, dan lain sebagainya; (3) HTML. Bahasa pemrograman JavaScript ini memiliki kelemahan sendiri yaitu JavaScript tidak mendukung browser versi lama.



GAMBAR 2.7 Logo Javascript

(Sumber: <https://seeklogo.com/vector-logo/273557/javascript-js>)