

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Rancang Bangun**

##### **2.1.1. Rancang**

Menurut (Farhan, 2017) perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem kedalam bahasa pemograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplemntasikan.

##### **2.1.2. Bangun**

Menurut (Farhan, 2017) pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Jadi dapat disimpulkan bahwa rancang bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

#### **2.2. Sistem Informasi**

Menurut (Wahyu, 2019) Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi harian mendukung oprerasi, bersifat yang menggunakan komputer bisa disebut sistem informasi berbasis komputer (computer Based Information Systems atau CBIS), Dalam praktik dalam istilah sistem informasi lebih sering di pakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Yang dimaksud sistem informasi adalah sistem informasi yang berbasis komputer.

#### **2.3. Pemesanan**

Menurut (Simatupang, 2020)Pemesanan adalah suatu aktifitas yang dilakukan oleh pelanggan sebelum membeli.Untuk mewujudkan kepuasan pelanggan maka

perusahaan harus mempunyai sebuah sistem Pemesanan yang baik. Menurut Kamus Besar Bahasa Indonesia yang dimaksud Pemesanan adalah proses, perbuatan, cara memesan tempat, barang, dsb) kepada orang lain.

#### **2.4. Basis Data ( Data Base )**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. (Rosa dan Shalahuddin, 2015).

#### **2.5. Bahasa Pemrograman dan Perangkat Lunak Penunjang**

Bahasa pemrograman dan perangkat lunak pendukung yang digunakan dalam penulisan ini adalah antara lain Website, XAMPP, PHP, Laravel, HTML, CSS, dan JavaScript.

#### **2.6. Website**

Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Abdulloh, 2015).

#### **2.7. XAMPP**

Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lainnya. Fungsi dari Xampp adalah untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL, dan PhpMyAdmin. (MADCOMS, 2016).

#### **2.8. PHP**

PHP singkatan dari Hypertext Preprocessor yang merupakan server-side programming, yaitu bahasa pemrograman yang diproses di sisi server. Fungsi utama

PHP dalam membangun website adalah untuk melakukan pengolahan data pada database. Data website akan dimasukkan ke database, diedit, dihapus, dan ditampilkan pada website yang diatur oleh PHP.(Rohi Abdulloh, 2015).

## **2.9. HTML**

HTML singkatan dari Hypertext Markup Language, yaitu script yang berupa tag-tag untuk membuat dan mengatur struktur website. (Rohi Abdulloh,2015).

Beberapa tugas HTML dalam membangun website diantaranya, sebagai berikut:

- a) Menentukan layout website.
- b) Memformat text dasar seperti pengaturan paragraph, dan format font.
- c) Membuat list dan formulir.
- d) Membuat tabel, gambar, video, audio, dan link.

## **2.10. CSS**

Cascading Style Sheets, yaitu skrip yang digunakan untuk mengatur desain website. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih indah (Rohi Abdulloh, 2015).

## **2.11. JavaScript**

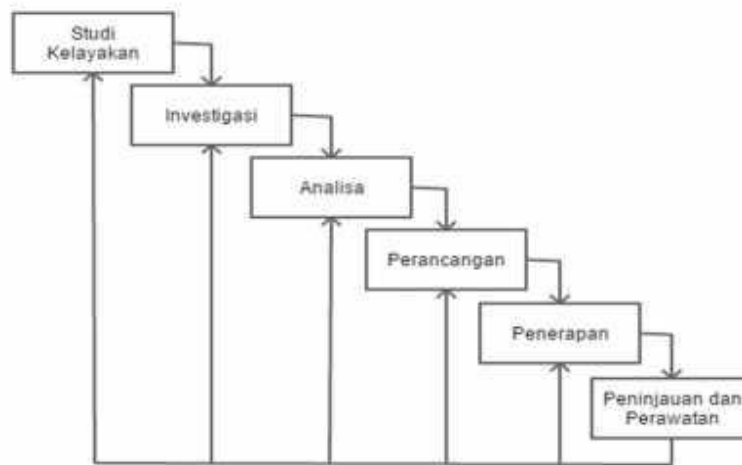
JavaScript ialah suatu bahasa scripting yang digunakan sebagai fungsionalitas dalam membuat suatu web. (Hidayatullah dan Kawistara, 2015).

## **2.12. MySQL**

MySQL adalah salah satu aplikasi DBMS yang sudah banyak oleh para pemogram aplikasi web. Contoh DBMS lainnya adalah : PostgreSQL (freeware), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dsb". (Hidayatullah dan Jauhari,2015).

### 2.13. Metode Pengembangan Sistem

Model *waterfall* merupakan model SDLC yang mirip seperti air terjun. Model ini mengusulkan sebuah pendekatan kepada pengembangan *software* yang sistematis dan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Model *waterfall* ditunjukkan pada Gambar 2.1 berikut ini.



Gambar 2.1 Model *Waterfall*

### 2.14. UML( Unified Modeling Language )

Menurut (Wido,2018) UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori objectoriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna. Tujuan pemodelan dalam kerangka pengembangan sistem adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar tim pengembang yang beranggotakan beberapa/banyak anggota.

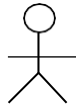



Beberapa diagram dalam UML yang akan digunakan dalam membantu pengembangan sistem yaitu :

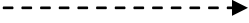
### 2.14.1. Use Case Diagram

Use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara asar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram use case :

Table 2 1 Simbol-simbol use case diagram

No	Simbol	Keterangan Fungsi
1.	Aktor 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	Use Case 	Use Case adalah deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya atau sebaliknya dari bawah ke atas.





5.	<i>Defendancy</i> 	<i>Defendancy</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (independen).
----	--------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------


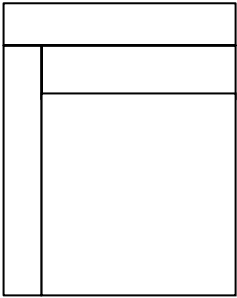
### 2.14.2. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Table 22 Simbol-simbol activity diagram.

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah statusakhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

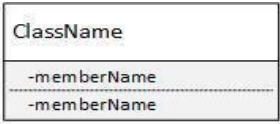

### 2.14.3. Class Diagram


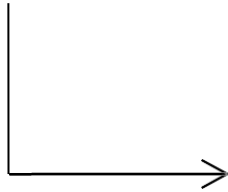
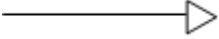

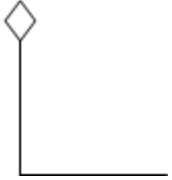
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan method atau operasi. Berikut penjelasan atribut dan method:

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau method adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3 Simbol-simbol Class Diagram

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>interface</i> 	Samadengankonsep <i>interface</i> dalampemrograman berorientasi objek

3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum- khusus)
6	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agrgasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

#### 2.14.4. Squend Diagram

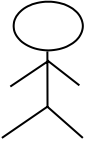
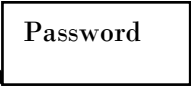

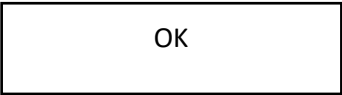

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

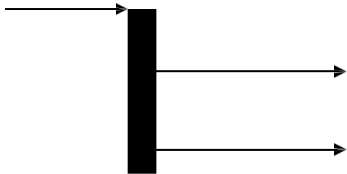


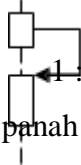




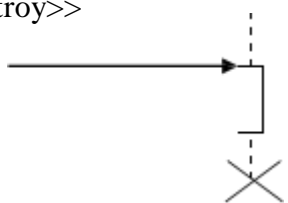
Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Table 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
1	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor</p>
2.	<p>Garis hidup //lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi peran</p>
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan</p>

		<p>di dalamnya, misalnya</p> <p>2: cek Status Login()</p> <p>1: login ()</p> <p>3 : open()</p> <p>Maka cek status login() dan open() dilakukan didalam metode login().</p> <p>Akor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe <i>create</i></p> <p>&lt;&lt;create&gt;&gt;</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/ metode, maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7.	<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Meyatakan bahwa suatu objek mengirimkan data/masukan/informasi/ ke objek lainnnya, arah panah mengarah pada objek yang menerima kembalian.</p>
8.	<p>Pesan tipe <i>return</i></p> <p>1: keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau</p>

		metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	<p>Pesan tipe <i>destroy</i></p> <p>&lt;&lt;destroy&gt;&gt;</p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i>