

## **BAB II TINJAUAN PUSTAKA**

### **2.1. Sistem Informasi Manajemen**

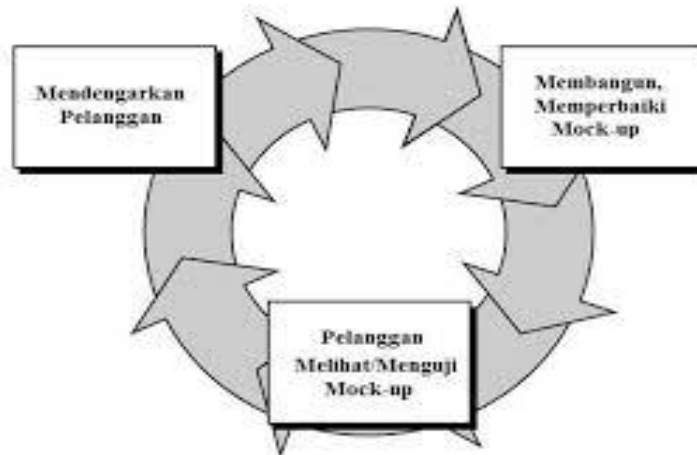
Sistem Informasi Manajemen (SIM) adalah suatu sistem yang digunakan untuk mengelola informasi yang berkaitan dengan kegiatan manajemen suatu organisasi atau bisnis. Tujuannya adalah untuk membantu manajemen dalam membuat keputusan yang tepat dan efektif. SIM meliputi berbagai aspek, seperti pengumpulan, penyimpanan, pengolahan, dan distribusi informasi yang relevan bagi manajemen (Yunitarini, 2018).

Sistem informasi manajemen adalah sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam transformasi yang teratur (Muliadi, Andriani and Irawan, 2020).

Sistem Informasi Manajemen merupakan suatu sistem berbasis komputer yang membuat informasi tersedia bagi para pengguna yang memiliki kebutuhan serupa. Sistem informasi manajemen organisasi adalah sebuah sistem manusia atau mesin yang terpadu (integrated) untuk menyajikan informasi guna mendukung fungsi operasi, manajemen, dan pengambilan keputusan dalam sebuah (Yusuf, 2018).

### **2.2. Pengembangan Sistem *Prototype***

Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa and Shalahudin, 2018).



**Gambar 2. 1 Ilustrasi model *prototype***  
**Sumber :** (Rosa and Shalahudin, 2018)

Menurut (Rosa and Shalahudin, 2018) terdapat tahapan dalam proses *prototipe* yaitu:

1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Merancang dan Membuat *Prototipe*

Pada tahap ini, dilakukan perancangan dan pembuatan prototipe sistem yang disesuaikan dengan kebutuhan pengguna.

3. Uji Coba

Pada tahap ini, dilakukan pengujian *prototipe* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan prototipe, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih belum sesuai kembali ke tahap pertama.

**2.3. UML (*Unified Modeling Language*)**

Menurut (Rosa and Shalahudin, 2018) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan


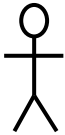


mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara lain sebagai berikut:

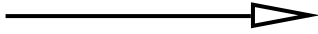

### 1. *Use Case Diagram*

Menurut (Rosa and Shalahudin, 2018) *use case* diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *use case*. Para aktor terhubung oleh garis ke *use case* yang mereka kerjakan.

*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini:

**Tabel 2. 1 Simbol diagram *use case***

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan



Simbol	Deskripsi
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini


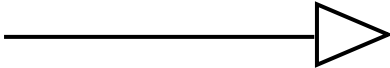

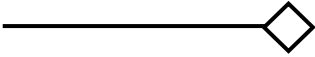
Sumber: (Rosa and Shalahudin, 2018)

## 2. Class Diagram

Menurut (Rosa and Shalahudin, 2018) Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas. Dan bagian bawah merupakan *operation* dari kelas tersebut. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2.

Tabel 2. 2 Simbol *Class Diagram*

Simbol	Deskripsi			
Kelas <table border="1" style="margin-left: 20px;"> <tr><td><b>nama_kelas</b></td></tr> <tr><td>+atribut</td></tr> <tr><td>+operasi()</td></tr> </table>	<b>nama_kelas</b>	+atribut	+operasi()	Kelas pada struktur sistem
<b>nama_kelas</b>				
+atribut				
+operasi()				
Antarmuka/ <i>Interface</i>  <b>nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek			
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			




Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )


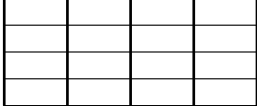


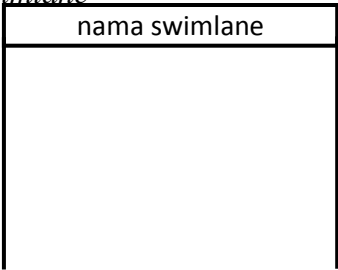
Sumber: (Rosa and Shalahudin, 2018)

### 3. Activity Diagram

Menurut (Rosa and Shalahudin, 2018) Sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan. Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 3 Simbol Activity Diagram

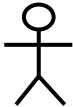
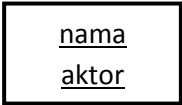
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu


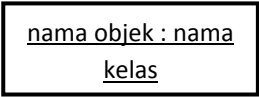


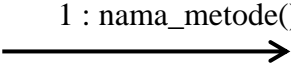
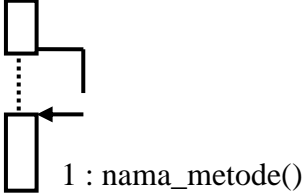
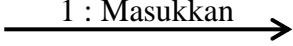
Simbol	Deskripsi
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

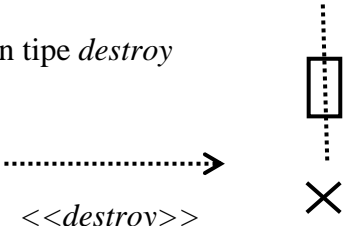
#### 4. *Squential Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case*. simbol-simbol yang ada pada *sequence* digram dapat dilihat pada tabel 2.4 di bawah ini :

**Tabel 2. 4 Simbol *Sequence Diagram***

Simbol	Deskripsi
Aktor  	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem

Simbol	Deskripsi
	informasi yang akan dibuat itu sendiri,
<p>Atau</p> <p><b>nama aktor</b> tanpa waktu aktif</p>	jadi walaupun simbol dari aktor gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda
<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p> 
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirim data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim

Simbol	Deskripsi
<p>Pesan tipe <i>destroy</i></p>  <p>.....&gt; &lt;&lt;destroy&gt;&gt;</p>	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

Sumber : (Rosa and Shalahudin, 2018).

## 2.4.Pengertian SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada *Relation DBMS (Database Manajement System)* (Rosa and Shalahudin, 2018).

Singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, program manajemen basis data atau pabrikan dan sering disebut Manajemen Database Sistem (DBMS). Ini adalah sifat *open source* DBMS. MySQL juga merupakan basis data yang diakses jaringan program, sehingga dapat digunakan untuk multiuser aplikasi (banyak pengguna) (Wadisman *et al.*, 2019).

SQL adalah perangkat lunak relation database *manajement system* (RDBMS) yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas (Kristanto, 2018)

Jadi *Structure Query Language* adalah perangkat lunak relation databasemanajement system (RDBMS) mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan kinerja basis data.



## **2.5. Mobile**

Mobile adalah sebuah sistem perangkat lunak yang memungkinkan setiap pemakai melakukan mobilitas dengan perlengkapan PDA-asisten digital perusahaan pada telepon genggam atau seluler. Android dan iOS merupakan sistem operasi mobile yang untuk saat ini merajai pasaran. Aplikasi mobile juga dikenal sebagai web app, online app, iPhone app atau smartphone app. (Yuhefizar, 2019).

## **2.6. Java**

Java merupakan bahasa pemrograman yang digunakan secara luas untuk pengodean aplikasi web. Bahasa ini telah menjadi pilihan populer di antara developer selama lebih dari dua dekade, dengan jutaan aplikasi Java yang digunakan saat ini. Java merupakan bahasa multiplatform yang berorientasi pada objek dan berpusat pada jaringan yang dapat digunakan sebagai platform di dalamnya. Java merupakan bahasa pemrograman yang cepat, aman, dan andal untuk mengodekan segala sesuatu mulai dari aplikasi seluler dan perangkat lunak korporasi hingga aplikasi *big data* dan teknologi sisi server. (Yuhefizar, 2019).

## **2.7. Pengujian *Black Box***

Menurut Rosa dan Shalahuddin (2018) mengatakan bahwa pengujian *Black-Box* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi – fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang mencoba semua fungsi dengan memakai perangkat lunak, apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black-box* dengan kasus benar dan kasus salah.

## **2.8. Hasil Penelitian Sebelumnya**

Berikut ini adalah beberapa literature yang digunakan dalam penelitian, dapat dilihat pada tabel berikut :

**Tabel 2. 5 Hasil Penelitian**

No	Judul Penelitian	Nama Tahun	Metode	Hasil
1	Perancangan Sistem Informasi Pemesanan Kamar Hotel Berbasis Online	(Purba, 2020)	<i>Waterfall</i>	Sistem Informasi Pemesanan Kamar Hotel dapat memudahkan receptionist untuk melakukan pengolahan data reservasi hotel yang meliputi kamar yang tersedia, biaya sewa kamar, dan semua fasilitas hotel yang dapat digunakan oleh tamu
2	Perancangan Sistem Informasi Pemesanan Kamar Hotel Berbasis Website (Web) Menggunakan Data Flow Diagram (Dfd)	(Muliadi, Andriani and Irawan, 2020)	SDLC	Sistem informasi pemesanan kamar hotel yang berbasis website (WEB) ini memudahkan pelanggan untuk mengetahui informasi hotel dan sistem pemesanan kamar pada Hotel Kartika. website (WEB) juga menampilkan data fasilitas hotel, data kamar hotel, data laporan hotel beserta data harga kamar hotel dan tipe kamar hotel secara jelas karena tersedianya database yang baik dalam bentuk file-file komputer.
3	Perancangan Sistem Informasi Manajemen Perhotelan Berbasis Web (Studi Kasus Hotel XYZ)	(Yanto and Darmawan, 2015)	<i>Waterfall</i>	Sistem Informasi Manajemen Perhotelan yang proses pemesanannya sudah berbasis online ini dapat mempermudah para pelanggan untuk melakukan proses pemesanan kamar