

BAB II

TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem adalah sekumpulan unsur atau elemen dan komponen-komponen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Fernanda dan Fernando, 2017)

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Wati, Rahmanto and Fernando, 2019)

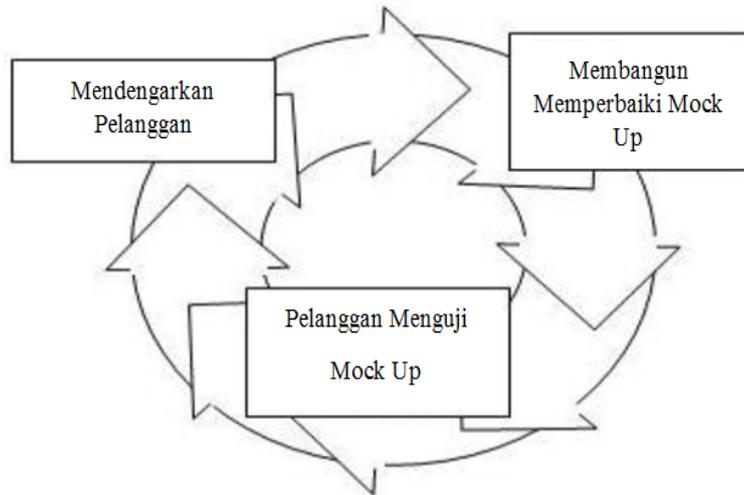
Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018).

Sistem informasi ialah suatu sistem yang menyajikan informasi yang bertujuan untuk mengambil suatu keputusan dan menjalankan kegiatan operasional suatu organisasi dan memberikan suatu keunggulan kompetitif (Awaludin, Bahri and Muslih, 2019)

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Adrian and Pramono, 2017)

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 *prototype* 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*)

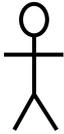
Menurut (Rosa and Shalahudin, 2018) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara 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 actor
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
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> 	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

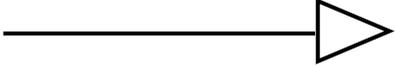
Simbol	Deskripsi
	menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

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*

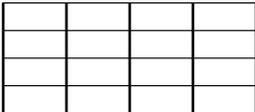
Simbol	Deskripsi			
Kelas <table border="1" style="margin-left: 20px;"> <tr><td>nama_kelas</td></tr> <tr><td>+atribut</td></tr> <tr><td>+operasi()</td></tr> </table>	nama_kelas	+atribut	+operasi()	Kelas pada struktur sistem
nama_kelas				
+atribut				
+operasi()				
Antarmuka/ <i>Interface</i>  nama_interface	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>dependecy</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 (Berliani, 2020) 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
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

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 management system* (RDBMS) yang didesain untuk melakukan proses manipulasi *database* berukuran besar dengan berbagai fasilitas (Kristanto, 2018)

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

2.4. Pengujian *Black Box*

Menurut (Herdiansah, Handayani and Yunita, 2020) 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.

2.5. Hasil Penelitian Sebelumnya

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

Tabel 2. 4 Hasil Penelitian

No	Judul Penelitian	Nama Tahun	Metode	Hasil
1	Sistem Informasi Reservasi Tour Agency Di Jawa Tengah Berbasis Web	(Hardadi, 2020)	<i>Waterfall</i>	Sistem Informasi Reservasi Tour Agency Di Jawa Tengah Berbasis Web adalah Untuk membuka peluang usaha pariwisata dengan cara memasarkan tempat wisata yang ada di Indonesia khususnya Jawa Tengah dengan menggunakan sistem berbasis aplikasi website dan memberikan kemudahan bagi para pelanggan yang akan memilih destinasi tujuan tempat wisata yang di pilih beserta fasilitas yang tersedia didalamnya seperti penginapan yang telah disediakan dalam pilihan paket paket perjalanan wisata dalam “tour agency”.

No	Judul Penelitian	Nama Tahun	Metode	Hasil
2	Sistem Informasi Pelayanan Jasa Tour Dan Travel Berbasis Web (Studi Kasus Smart Tour)	(Muliadi, Andriani and Irawan, 2020) (Riskiono, 2018)	Extreme programming	Sistem informasi pelayanan jasa tour dan travel berbasis web (Studi kasus Smart Tour) dapat mampu mengurangi resiko kesalahan pada pengelolaan data pada admin dan Sistem informasi pelayanan jasa tour dan travel berbasis web (Studi kasus Smart Tour) telah mampu mengurangi waktu yang terbuang pada saat pelanggan memesan tiket tour dengan menghasilkan kriteria baik yaitu persentase tanggapan responden sebesar 100%
3	Sistem Informasi Reservasi Paket Wisata Berbasis Web	(Oktapiah and Hasti, 2020)	Waterfall	Sistem informasi ini di dalam proses pembuatan laporan tidak mengalami kesulitan karena data sudah tersimpan secara otomatis dan terintegrasi dengan database
4	Sistem Informasi Lowongan Mencari Kerja Berbasis Web	(Zaini, Nursiyanto and Pratama, 2022)	RAD	Sistem informasi lowongan mencari kerja (SILMENKER) dibangun untuk memudahkan perusahaan membuka lowongan pekerjaan sehingga pencari kerja mudah untuk mendaftarkan dirinya melalui website yang telah dibangun.