

BAB II LANDASAN TEORI

2.1. Sistem

Sistem adalah Kumpulan dari elemen elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2014).

2.2. Informasi

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).

2.3. Sistem Informasi

Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Kristanto, 2018).

2.4. Pengertian Pemesanan

Pemesanan yaitu fasilitas yang diantaranya akomodasi, meal, seat pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, night club, *discoutegue* dan sebagainya (Suartana 2017)

2.4.1. Jenis Pemesanan

Pemesanan dibagi menjadi dua jenis yaitu pemesanan online dan pemesanan offline :

1) Pemesanan *Online*.

Kemajuan teknologi saat ini mengakibatkan sistem pemesanan juga mengalami perkembangan ke arah sistem pemesanan online. Pemesanan online bisa diakses oleh siapapun dan dimanapun mereka berada yang memiliki akses internet.

2) Pemesanan *Offline*

Sistem pemesanan yang menggunakan pengiriman pemesanan langsung ke tempat dengan media pemesanan seperti telepon, *fax*, *e-mail*, dan *walk in*.

2.5. Aplikasi Mobile

Aplikasi Mobile adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer dan Handphone. Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi mobile merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA (Yuhefizar, 2014).

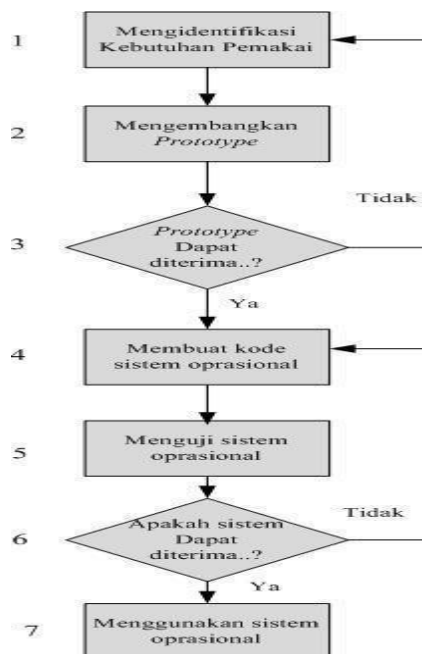
Adapun karakteristik perangkat mobile yaitu:

1. Ukuran yang kecil: Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas: Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk).

3. Daya proses yang terbatas: Sistem mobile tidaklah setangguh rekan mereka yaitu desktop.
4. Mengonsumsi daya yang rendah: Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop
5. Kuat dan dapat diandalkan: Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

2.6. Metode Pengembangan Sistem *Prototype*

Menurut McLoad (2017) mendefinisikan *Prototype* adalah suatu versi sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna yang memberikan gambaran bagaimana sistem akan berfungsi dalam bentuk yang telah selesai akan berfungsi bila telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu prototipe disebut *Prototyping*”.



Gambar 2. 1 Metode Prototyping Evolutionary

2.7. Perancangan Sistem UML (*Unified Modeling Language*)




UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa and Shalahudin, 2018).




1. *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Rosa and Shalahudin, 2018).

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, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
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

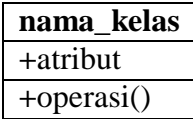


Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan
<pre><<extend>></pre> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</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 menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini
<pre><<include>></pre> 	





Sumber: (Rosa and Shalahudin, 2018)

2. *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 metode atau operasi. simbol-simbol yang ada pada diagram kelas pada tabel class diagram 2.2 di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
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>)



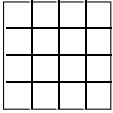


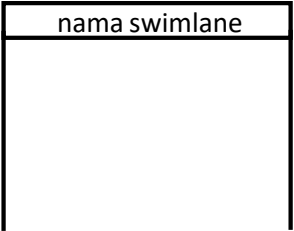
Sumber: (Rosa and Shalahudin, 2018)

3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 1 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.

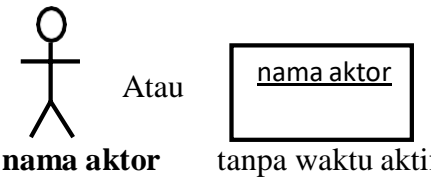

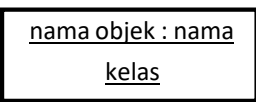

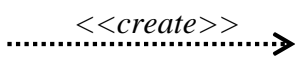
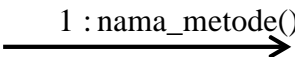
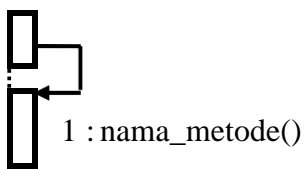
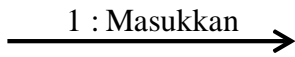
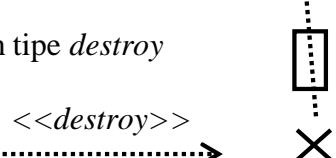
Percabungan/ <i>decision</i> 	Asosiasi percabungan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	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

Sumber: Rosa, A.S (2018)

4. *Sequence 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. 2 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat</p>
<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> 	<p>Menyatakan bahwa suatu objek mengirim data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>destroy</i></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)

2.8. XAMPP

Menurut Nugroho (2017), XAMPP adalah paket program *web* lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL.

Kelebihan dari XAMPP adalah performanya yang tinggi, stabil, memiliki banyak fitur, mudah dikonfigurasi, menggunakan hanya sedikit sumberdaya pada server, tidak bergantung kepada thread untuk melayani klien.

Kekurangan dari XAMPP adalah belum *support* IPV6, *update/patchnya* versi terbarunya lama keluar, *Fast-CGI* nya tidak berfungsi maksimal, pemakainya tidak sebanyak program *web* yang lain.

2.9. Android

Android adalah sistem operasi untuk telepon seluler berbasis Linux sebagai kernelnya. Saat ini Android bisa disebut raja dari *smartphone*, hal ini dikarenakan Android menyediakan *platform* terbuka (*open source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Awalnya, perusahaan *search engine* terbesar yaitu Google Inc. membeli Android, Inc. didirikan oleh Andy Rubin, Rich Milner, Nick Sears dan Chris White pada tahun 2003. Sedangkan pada Agustus 2005 Google membeli Android Inc. Kemudian untuk mengembangkan Android dibentuklah *open handset alliance* konsorsium dari 34 perusahaan *hardware*, *software* dan telekomunikasi, termasuk Google, HTC, IntelMotorola, Qualcomm, T-Mobile dan Nvidia.

2.10. Android Studio

Android Studio adalah *software platform* yang bersifat *open source* untuk *mobile device*. Android berisi sistem operasi, *middleware* dan aplikasi-aplikasi dasar. Android menyediakan *platform* terbuka bagi para pengembang untuk

menciptakan aplikasi. Basis OS Android adalah kernel linux 2.6 yang telah dimodifikasi untuk *mobile device* (Afrianto, 2018).

2.11. Pengujian *Black – Box*

Menurut Dondeti (2012),

“Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan pereayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program”.

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir. Karena hanya fungsi dari modul perangkat lunak yang menjadi perhatian, pengujian Black-Box juga mengacu pada uji fungsional, metode pengujian menekankan pada menjalankan fungsi dan pemeriksaan inputan dan data output (Howden, 2017)

Pengujian black-box berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang,
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

