



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Aplikasi

Menurut Jogiyanto (2015:10), aplikasi adalah sekelompok atribut yang terdiri dari beberapa form report yang disusun sedemikian rupa sehingga dapat mengakses data. Aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jogiyanto menambahkan aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Menurut Asropudin (2013:32), aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, dan Microsoft Exel.

Sedangkan menurut Hapsoro (2016:18), aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer.

Dari pengetahuan diatas dapat disimpulkan bahwa aplikasi merupakan suatu perangkat lunak, software yang bertugas untuk melakukan sekumpulan perintah atau tugas-tugas tertentu yang dijalankan oleh user yang saling berkaitan

2.1.2 Pengertian Pemesanan

Menurut Utara (2014:22), pemesanan adalah suatu aktivitas yang dijalankan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Tujuan pemesanan yaitu :

1. Memaximumkan pelayanan bagi konsumen.
2. Meminimumkan investasi pada persediaan.
3. Perencanaan kapasistas
4. Persediaan dan kapasitas.
5. Dan lain-lain.



Menurut Septian (2017:16), pemesanan dalam arti umum adalah perjanjian antara 2 pihak atau lebih, perjanjian tersebut berupa produk atau jasa.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa pemesanan ialah transaksi yang dilakuan antara produsen dan konsumen sebelum membeli suatu produk atau jasa yang telah mempunyai kesepakatan antara dua pihak.

2.1.3 Pengertian Pembayaran

Desita (2016:12), pembayaran adalah suatu proses memberikan uang untuk pengganti harga barang yang diterima dan melunasi utang. Sistem pembayaran dibagi menjadi dua yaitu sistem pembayaran tunai dan sistem pembayaran non tunai. Sistem pembayaran tunai biasanya menggunakan uang tunai seperti uang kertas atau logam. Sedangkan sistem pembayaran non tunai menggunakan kartu kredit, giro, debet dan pembayaran uang elektronik.

2.1.4 Pengertian *Grand Atyasa Convention Center*

Grand Atyasa Convention Center merupakan salah satu gedung serbaguna yang setara dengan hotel berbintang yang terletak di Palembang. Yang berdiri pada tahun 2013. *Grand Atyasa Convention Center* menyediakan berbagai jenis tipe room dan fasilitas yang ada di *Grand Atyasa Convention Center*.

2.1.5 Pengertian Aplikasi Pemesanan *Grand Atyasa Convention Center*

Aplikasi pemesanan *Grand Atyasa Convention Center* merupakan aplikasi yang dibuat untuk mempermudah pelanggan dalam melakukan pesanan terhadap gedung yang tersedia dan tipe room yang ada di *Grand Atyasa Convention Center* secara *online*.

2.2 Teori Khusus

2.2.1 *Object Oriented Program (OOP)*

Menurut Abdullah (2017:30), OOP (*Object Oriented Program*) merupakan teknik pemograman dengan menggunakan konsep objek. Tujuan dari OOP adalah untuk memudahkan programmer dalam pembuatan program dengan menggunakan



konsep objek yang ada dalam kehidupan sehari-hari. Jadi setiap bagian permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.

Sebuah objek pada OOP memiliki data atau disebut property yang menjelaskan tentang sifat-sifat objek tersebut. Seperti sebuah handphone dapat memiliki data warna, merk, ukuran layar, dan sebagainya. Begitu juga dengan objek-objek yang ada didalamnya seperti layar memiliki data berupa lebar, tinggi dan sebagainya.

Selain memiliki property, sebuah objek dalam OOP memiliki method berupa fungsi yang dapat dipanggil untuk melakukan tindakan atau merubah nilai dari property yang ada di dalamnya. Seperti handphone dapat melakukan tindakan merekam, restart, memanggil, mengirim pesan dan sebagainya. Handphone juga dapat diganti *casing* untuk mengubah warnanya (mengubah nilai property).

2.2.2 *Unified Modelling Language (UML)*

Menurut Salahuddin (2014:28) pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak.

Pada perkembangan perkembangan teknik pemograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemodelan berorientasi objek, yaitu *Unified Modeling Language (UML)*. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

2.2.3 **Class Diagram**

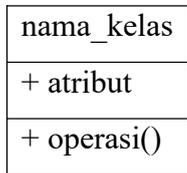
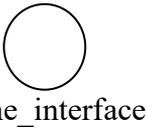
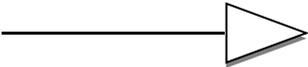
Menurut Salahuddin (2014:55) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan



dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.1 Simbol Class Diagram

No	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka / <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

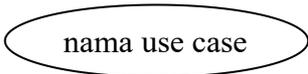


Sumber: Salahuddin (2014:55)

2.2.4 Use Case Diagram

Menurut Salahuddin (2014:48), *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 aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>
2		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem 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 fase nama aktor
3		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang



ditambahkan dapat berdiri sendiri

Lanjutan **Tabel 2.4** Simbol Use Case Diagram

No	Simbol	Deskripsi
		walaupun 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 yang sama dengan <i>use case</i>
5	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.
6	Menggunakan / <i>include</i> / <i>uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <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: Salahuddin (2014:48)

2.2.5 Activity Diagram

Menurut Salahuddin (2014:70), 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 diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

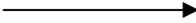
Tabel 2.3 Simbol Activity Diagram

No	Simbol	Deskripsi
1	Start / status awal (<i>Intial</i>)	Start atau intial state adalah state



	<i>State</i>)	atau keadaan awal pada saat sistem
--	----------------	------------------------------------

Lanjutan **Tabel 2.5** Simbol Activity Diagram

No	Simbol	Deskripsi
		mulai hidup
2	End / status akhir (<i>final state</i>) 	End atau final state adalah state keadaan akhir dari daur hidup suatu sistem.
3	Event 	Event adalah kegiatan yang menyebabkan berubahnya status mesin.
4	State 	Sistem pada waktu tertentu. State dapat berubah jika ada event tertentu yang memicu perubahan tersebut.

Sumber: Salahuddin (2014:70)

2.2.6 Sequence Diagram

Menurut Salahuddin (2014:66), 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 menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diintansiasi 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 pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel 2.4 Simbol Sequence Diagram

No	Simbol	Deskripsi
	<p>Aktor</p>  <p>atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_aktor</div>	<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 adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

Sumber: Salahuddin (2014:66)