BAB II
TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Aplikasi

Menurut Jogiyanto (2012), 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 Elisa (2016), aplikasi sering juga disebut sebagai perangkat lunak, merupakan program komputer yang isi instruksinya dapat diubah dengan mudah.

Sedangkan menurut Solichin (2016:1), aplikasi atau perangkat lunak (*software)* merupakan bagian yang tidak terpisahkan dari suatu sistem komputer.

Dari pengetian 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 (2011), 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. Memaksimumkan pelayanan bagi konsumen.
2. Meminimumkan investasi pada persediaan.
3. Perencanaan kapasistas
4. Persediaan dan kapasitas.
5. Dan lain-lain.

Menurut Septian (2017), 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 Percetakan

Steele (2014), menyatakan percetakan offset penawaran kualitasa yang menarik bagi komunitas kreatif. Manfaat ini dikombinasikan dengan alur kerja yang matang masih akan bertahan untuk beberapa tahun kedepan. Pasar percetakan offset tidak mungkin diimbangi oleh percetakan digital dalam waktu dekat. Dari segi biaya per halaman dan *dutycyle*, percetakan offset masih belum bisa dikalahkan percetakan digital.

Fauzi et.al (2018), Percetakan merupakan sebuah teknologi yang memproduksi salinan dari sebuah dokumen atau foto dengan cepat, seperti kata-kata, gambar yang berada di atas sebuah media seperti kain, kertas, kayu dan sebagainnya.

Dari pengertian diatas dapat disimpulkan bahwa percetakakan merupakan teknologi yang mana sampai saat ini masih banyak digunakan oleh masyarakat sebagai salah satu media untuk melakukan pencetakan seperti undangan, kartu nama dan lain-lain.

2.1.4 Pengertian AW Digital Print

AW digital Print merupakan salah satu percetakan yang terletak di Palembang. Yang berdiri pada tahun 2015 dengan jumlah pegawai 4 orang. Percetakan AW Digital Print menyedikan berbagai jenis produk percetakan seperti, baliho, undangan, banner, kartu nama, stiker dan lain-lain.

2.1.5 Pengertian Aplikasi Pemesanan *Online* pada Percetakan AW Digital Print

Aplikasi pemesanan *online* pada percetakan AW Digital Print merupakan aplikasi yang dibuat untuk memermudah pelangggan dalam melakukan pesanan terhadap produk percetakan yang ada di AW Digital Print secara *online.*

2.2 Teori Khusus

2.2.1 *Object Oriented Program* (OOP)

Menurut Abdullah (2017), 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 dalamm 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 handpone dapat memiliki data warna, merk, ukuran layar, dan sebagainnya. Begitu juga dengan objek-objek yang ada didalamnya seperti layar memiliki data berupa lebar, tinggi dan sebagainnya.

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, restrart, memanggil, mengirim pesan dan sebagainnya. Handphone juga dapat diganti *casing* untuk mengubah warnanya (mengubah nilai property).

2.2.2 *Unified Modelling Language* (UML)

Menurut (Salahuddin, 2014) 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

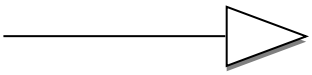
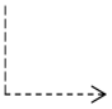
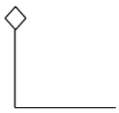
Menurut (Sukamto dan Salahuddin, 2018:141) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebuat atribut dan metode atau operasi. Berikut penjelasan atribut dan *method :*

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

Tabel 2.1 Simbol Class Diagram

No	Simbol	Deskripsi
1	Kelas ClassName + memberName + memberName	Kelas pada struktur sistem
2	Antarmuka / *interface*  name_interface	sama dengan konsep *interface* dalam pemograman berorintasi objek.
3	Asosiasi / *association* 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity.*

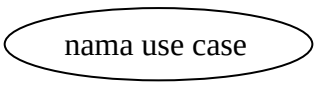
Lanjutan **Tabel 2.1** Simbol Class Diagram

No	Simbol	Deskripsi
4	Asosiasi berarah / *directed association* 	Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	Kebergantungan / *dependency* 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	Agregasi / *aggregation* 	Relasi antarkelas dengan makna semua-bagian (*whole-part)*

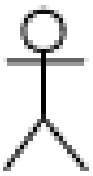
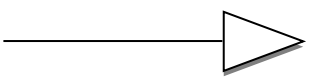
2.2.4 Use Case Diagram

Menurut (Sukamto dan Salahuddin, 2018:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakukan (*behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor denagan 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	*Use case* 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anta unit atau aktor; biasanya dinyatakan dengan menggunkana kata

Lanjutan Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
		kerja di awal fase nama *use case*
2	Aktor / *actor* 	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 menggunkanan kata benda di awal fase nama aktor
3	Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki interkasi dengan aktor
4	Ekstensi / *extend* < < extend> >	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *ingeritance* pada pemograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case*
5	Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / *include / uses*	Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang

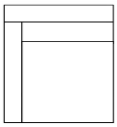
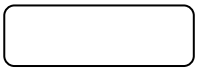
Lanjutan Tabel 2.2 Simbol Use Case

No	Simbol	Deskripsi
	< < include > > 	ditambahkan memerlukan *use case* ini.

2.2.5 Activity Diagram

Menurut (Sukamto dan Salahuddin, 2018:161), 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 (*Intial State)* 	Start atau intial state adalah state atau keadaan awal pada saat sistem mulai hidup.
2	End / status akhir (*final state)* 	End atau final state adalah state keadaan akhir dari daur hidup suatu sistem.
3	Swimlane 	Memisahkan organisasi bisnis yang bertanggunga jawab terhadap aktivitas yang terjadi.
4	Aktivitas 	Aktivitas yang dilakukan sistem aktivitas biasanya diawali dengan kata kerja.
5	Penggabungan/join	Asosiasi penggabungan dimana

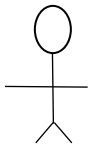
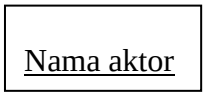
Lanjutan Tabel 2.3 Simbol Activity Diagram

No	Simbol	Deskripsi
		lebih dari satu aktivitas digabungkan menjadi satu
6	Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

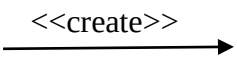
2.2.6 Sequence Diagram

Menurut (Sukamto dan Salahuddin 2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* 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..

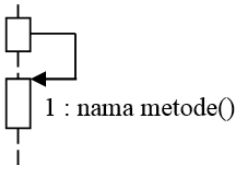
Tabel 2.4 Simbol Sequnce Diagram

No	Simbol	Deskripsi
1	Aktor  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri,

Lanjutan Tabel 2.4 Simbol Seunce Diagram

No	Simbol	Deskripsi
		jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2	Garis hidup/*lifeline*	Menyatakan kehidupan suatu objek
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubungdengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
5	Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Lanjutan Tabel 2.4 Simbol Sequnce Diagram

No	Simbol	Deskripsi
6	Pesan tipe *call* 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  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
7	Pesan tipe *send* 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe *return* 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan

Lanjutan Tabel 2.4 Simbol Sequnce Diagram

No	Simbol	Deskripsi
		suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.