



## BAB II TINJAUAN PUSTAKA

### 2.1 Teori Judul

#### 2.1.1 Pengertian Aplikasi

Menurut Asropudin (2013:6), “Aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, dan Microsoft Exel.”

Menurut Elisa (2016), “Aplikasi sering juga disebut sebagai perangkat lunak, merupakan program komputer yang isi instruksinya dapat diubah dengan mudah.”

Sedangkan menurut Chan (2017:4), “Aplikasi adalah koleksi window dan objek-objek yang menyediakan fungsi untuk aktivitas user, seperti pemasukan data, proses dan pelaporan.”

Dari pengetahuan-pengertian diatas, penulis menyimpulkan 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 Pengendalian

Menurut Zamzami, et.al (2018:93), “Pengendalian adalah serangkaian rencana dan prosedur yang dilakukan untuk menjamin bahwa tujuan organisasi dapat tercapai.”

Menurut Foster dan Sidharta (2019:242), “Pengendalian adalah fungsi manajemen yang melibatkan kegiatan pemantauan untuk memastikan bahwa mereka diselesaikan sebagaimana direncanakan dan mengoreksi setiap penyimpangan yang signifikan.”

Berdasarkan pengertian-pengertian diatas, penulis menyimpulkan bahwa pengendalian adalah suatu rencana atau kegiatan dari fungsi manajemen untuk memastikan apa yang direncanakan dapat tercapai

#### 2.1.3 Pengertian Persediaan

---

Menurut Haming dan Mahfudnurnajamuddin (2014:4) dikutip oleh Haming, et.al (2017:376) menyatakan, “Persediaan merupakan sumber daya ekonomi yang perlu diadakan dan di simpan untuk menunjang penyelesaian pengerjaan suatu produk. Sumber daya ekonomi tersebut dapat berupa kapasitas produksi, tenaga kerja, tenaga ahli, modal kerja, waktu yang tersedia, bahan baku, dan bahan penolong.”

Hidayat (2013:76) menjelaskan, “Persediaan adalah salah satu komponen yang memiliki peran penting dalam suatu aktivitas bisnis. Kelancaran proses produksi dan memenuhi permintaan (penjualan) akan sangat dipengaruhi oleh cara mengelola komponen dengan baik.”

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa persediaan adalah point penting dalam perusahaan untuk kelancaran proses bisnis, kelancaran produksi suatu produk dan memenuhi kebutuhan konsumen.

#### **2.1.4 Pengertian Semen**

Menurut KBBI, Semen adalah adukan kapur dan sebagainya untuk merekatkan batu bata, tembok dan sebagainya.

#### **2.1.5 Pengertian *Economic Order Quantity* (EOQ)**

Menurut Wijayanto (2012:223), “EOQ merupakan metode pengontrolan penyimpanan dengan mempertahankan basis kuantitas permintaan yang bersifat tetap pada saat level penyimpanan berada pada titik penyimpanan kritis yang telah ditetapkan.”

Menurut Haming, et.al (2017:378) “EOQ adalah jumlah unit bahan atau komponen yang dipesan oleh perusahaan dari pemasok (*supplier*) pada biaya persediaan yang minimum.”

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa EOQ merupakan metode yang mengendalikan penyimpanan agar biaya persediaan menjadi minimum.

#### **2.1.6 Pengertian PT. Satria Jaya Prima**

PT. Satria Jaya Prima adalah perusahaan yang bergerak dalam bidang pendistribusian semen. Yang berdiri pada bulan maret 2017. Yang beralamat di Lorong Sebatok, Duku, Ilir timur II, Kota Palembang, Sumatera Selatan, 30163.



### **2.1.7 Pengertian Aplikasi Pengendalian Persediaan Semen dengan Metode *Economic Order Quantity* (EOQ) pada PT. Satria Jaya Prima**

Aplikasi Pengendalian Persediaan Semen dengan Metode *Economic Order Quantity* (EOQ) pada PT. Satria Jaya Prima merupakan aplikasi yang di gunakan perusahaan untuk membantu perusahaan dalam pengendalian persediaan semen.

## **2.2 Teori Khusus**

### **2.2.1 *Object Oriented Program* (OOP)**

Menurut Abdullah (2017) OOP (*Object Oriented Program*) merupakan teknik pemrograman 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) 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 pemrograman 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,

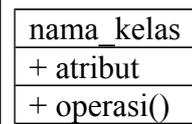
menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

### 2.2.3 Class Diagram

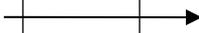
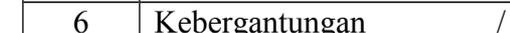
Menurut Salahuddin (2014) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

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

Lanjutan **Tabel 2.1** Simbol Class Diagram

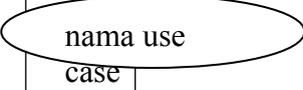
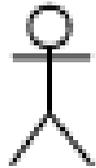
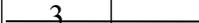
No	Simbol	Deskripsi
2	Antarmuka / <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</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 / 	Relasi antarkelas dengan makna

	<i>dependency</i> 	kebergantungan antar kelas.
7	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

### 2.2.4 Use Case Diagram

Menurut Salahuddin (2014) *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
	<i>Use case</i> 	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	Aktor / <i>actor</i> 	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	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
4	Ekstensi / <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah

	<< extend >> 	<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 pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i>
--	---	--

Lanjutan **Tabel 2.2** Simbol Use Case Diagram

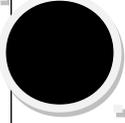
No	Simbol	Deskripsi
5	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>include</i> / <i>uses</i>  << include >> 	Relasi <i>use case</i> tambahan 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

### 2.2.5 Activity Diagram

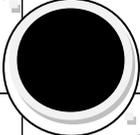
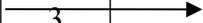
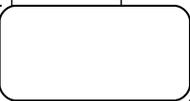
Menurut Salahuddin (2014) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan 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 State</i> ) 	Start atau intial state adalah state atau keadaan awal pada saat sistem mulai hidup
---	--	---

Lanjutan Tabel 2.5 Simbol Activity Diagram

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

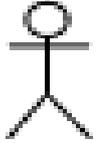
### 2.2.6 Sequence Diagram

Menurut Salahuddin (2014) 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 diinstantiasi 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 interaksinya 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
	Aktor  atau <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama_aktor</div>	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.

## 2.3 Teori Program

### 2.3.1 Pengertian PHP

Menurut Supono dan Viridiandry (2018:3) PHP adalah suatu bahasa pemrograman yang di gunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh computer yang bersifat *server-side* yang dapat ditambahkan ke HTML. Sedangkan menurut Sidik (2017:4) PHP merupakan secara umum dikenal sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

### 2.3.2 Pengertian MySQL

Menurut Sidik (2017:301) MySQL merupakan database yang termasuk paling populer di Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bias di katakana paling cepat, dan jarang bermasalah. MySQL telah terseddia juga di lingkungan windows. Sedangkan Anhar (2010:21) berpendapat bahwa MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (Database Management System) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL dan lain-lain.