



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Aplikasi

Menurut Elisa (2016:10), “Aplikasi sering juga disebut sebagai perangkat lunak, merupakan program komputer yang isi instruksinya dapat diubah dengan muda.”

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 pengertian-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 Persediaan

Menurut Martani et.al (2016:245) “persediaan merupakan salah satu aset yang sangat penting bagi suatu entitas baik bagi perusahaan ritel, manufaktur, jasa, maupun entitas lainnya.”

Yuliana, Topowijono, dan Sudjana (2016:72) dalam penelitian yang berjudul “Penerapan Model EOQ (*Economic Order Quantity*) dalam Rangka Meminimumkan Biaya Persediaan Bahan Baku (Studi pada UD. Sumber Rejo Kandangan – Kediri)” menyimpulkan bahwa “persediaan merupakan kekayaan perusahaan yang berupa bahan baku, bahan setengah jadi, dan bahan jadi.”

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.

2.1.3 Pengertian Buffer Stock

Menurut Wijayanto (2016:223), “Buffer stock atau yang biasa disebut juga dengan safety stock (stok pengaman) merupakan persediaan yang biasanya disiapkan oleh sebuah perusahaan dengan cara melebihi jumlah stok suatu



barang dalam gudang dengan tujuan untuk mengantisipasi adanya stock out (kekurangan bahan baku atau barang).”

Menurut Assauri (2018:56), “Buffer stock adalah persediaan tambahan yang diadakan untuk melindungi atau menjaga kemungkinan terjadi kekurangan bahan (stock out).”

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa *Buffer Stock* merupakan metode yang mengendalikan penyimpanan agar Persediaan Selalu sedia.

2.1.4 Pengertian CV

CV adalah suatu bentuk badan usaha persekutuan yang didirikan oleh seorang atau beberapa orang yang mempercayakan uang atau barang kepada seorang atau beberapa orang yang menjalankan perusahaan dan bertindak sebagai pemimpin untuk mencapai tujuan bersama dengan tingkat keterlibatan yang berbeda-beda di antara anggotanya.

2.1.5 Pengertian Aplikasi Stok Barang

Aplikasi Pengendalian Persediaan Stok barang dengan Metode *Buffer Stock* pada CV. Rantau Bayur merupakan aplikasi yang di gunakan untuk membantu CV. Rantau Bayur dalam pengendalian persediaan stok barang.

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 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)

Yuni Sugiarti (2018:60) mengemukakan “ UML adalah bahasa grafis yang kompleks dan kaya dengan fitur.”

Menurut Salahuddin (2016:12) 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.

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa UML adalah 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 Safaat (2015:33), “*Class Diagram* merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) *class diagram* menggambarkan struktur dan deskripsi class, *package*, dan objek yang berhubungan satu sama lain seperti *containment*, asosiasi, dan lain-lain”.

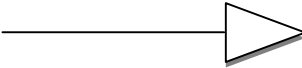
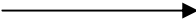
Menurut Salahuddin (2016:42) “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa, *Class Diagram* merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) *class diagram* menggambarkan struktur dan deskripsi class.

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 nama_kelas + atribut + operasi()	Kelas pada struktur sistem
2	Antarmuka / <i>interface</i>  name_interface	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>)

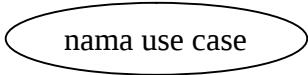
2.2.4 Use Case Diagram

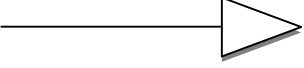
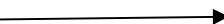
Menurut Salahuddin (2016:53), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Menurut Manalu (2015:67), “*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan”.

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa, *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.

Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
1	<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> < < extend > > 	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 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> < < 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 (2016:23), “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.

Menurut Rosa (2018:68), “Activity Diagram menggambarkan work flow (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 dapat dilakukan oleh sistem.”

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

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
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.

2.2.5 Sequence Diagram

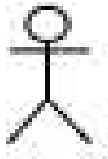
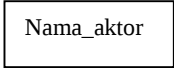
Menurut Salahuddin (2014:54) “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”. Oleh karena itu untuk

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.

Menurut Jurdam (2018:69) Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara object-oriented untuk menampilkan interaksi antar objek”.

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa, Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Tabel 2.4 Simbol Sequence Diagram

No	Simbol	Deskripsi
	Aktor  atau 	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.”

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.”



Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa, 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 performasi query dari databasenya yang saat itu bias di katakana paling cepat, dan jarang bermasalah”. MySQL telah terseddia juga di lingkungan windows.”

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.”

Dari pengertian-pengertian diatas, penulis menyimpulkan bahwa, MySQL merupakan database yang termasuk paling populer di Linux, kepopuleran ini karena ditunjang karena performasi query dari databasenya yang saat itu bias di katakana paling cepat, dan jarang bermasalah”. MySQL telah terseddia juga di lingkungan windows.