

BAB II

TINJAUAN PUSTAKA

II.1. Uraian Teoritis

Adapun teori-teori yang digunakan sebagai landasan dalam penulisan skripsi ini adalah sebagai berikut :

II.2.1. Aplikasi

Pengertian aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rudi Hartono;2015).

II.2.2. Kriptografi

Kriptografi berasal dari bahasa Yunani, "*kryptós*" yang berarti tersembunyi dan "*gráphein*" yang berarti tulisan. Sehingga kata kriptografi dapat diartikan berupa frase "tulisan tersembunyi". Menurut *Request for Comments* (RFC), kriptografi merupakan ilmu matematika yang berhubungan dengan transformasi data untuk

membuat artinya tidak dapat dipahami (untuk menyembunyikan maknanya), mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga bisa diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang dapat dipahami. Artinya, kriptografi dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas (Soehardi; 2015).

II.2.2.1 *Affine Cipher*

Menurut Diah Ayu Puspanigrum dan Ajib Susanto *Affine Cipher* adalah algoritma kriptografi yang merupakan perluasan dari *Caesar Cipher*. Dalam *Affine Cipher*, plainteks akan dikalikan dengan suatu nilai lalu di tambahkan dengan pergeseran tertentu.

Rumus Enkripsi :

$$C \equiv m P + b \pmod{n} \dots \dots \dots (1)$$

Rumus deskripsi :

$$P \equiv m^{-1}(C - b) \pmod{n} \dots \dots \dots (2)$$

Yang mana n adalah ukuran alphabet, m adalah bilangan bulat yang harus relatif prima dengan n (jika tidak relatif prima, maka dekripsi tidak bisa dilakukan) dan b adalah jumlah pergeseran (*Caesar Cipher* adalah bentuk khusus dari *Affine Cipher* dengan $m=1$). Untuk melakukan dekripsi, persamaan (2) harus dipecahkan untuk memperoleh P . Solusi kekongruenan tersebut hanya ada jika inver $m \pmod{n}$,

dinyatakan dengan m^{-1} . Jika m^{-1} ada maka dekripsi dilakukan dengan persamaan sebagai berikut: (Munir, 2006)

II.2.3. Website

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website* (anggihani Septima Riyadi,dkk; 2012).

II.2.4. PHP

PHP sendiri sebenarnya merupakan singkatan dari “*Hypertext Preprocessor*”, yang merupakan sebuah bahasa scripting tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Java dan Perl, namun pada PHP ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancang web yang dinamis dan dapat bekerja secara otomatis (Adis Lena Kusuma Ratna;2006).

II.2.5. HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa *formatting* yang digunakan untuk membuat sebuah halaman *website*. Di dalam dunia pemrograman berbasis *website* (*Web Programming*), HTML menjadi pondasi dasar pada halaman *website*. sebuah file HTML di di simpan dengan ekstensi *.html* (dot html). dan dapat di eksekusi atau diakses menggunakan web browser (Google Chrome, Mozilla Firefox, Opera, Safari dan lain-lain) (Diki Alfarabi Hadi; 2016).

II.2.6. MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang *multithread*, dan multi-user. MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuat oleh TeX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris (Adis Lena Kusuma Ratna; 2006).

II.2.7. Aplikasi *Mobile*

Aplikasi *mobile* dapat diartikan sebagai sebuah produk dari sistem komputasi *mobile*, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah *personal digital assistant* (PDA), *smartphone* dan ponsel (Muhdar Abdurahman: 2016).

II.2.8. Android

Android diciptakan oleh Android Inc yang selanjutnya di akuisisi oleh google Inc dan menjadi sistem operasi *open source* untuk pengembang agar dapat membuat aplikasi.

Android Studio merupakan sebuah Integrated Development Environment (IDE) khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. Android studio juga terintegrasi dengan Android *Software Development Kit* (SDK) untuk *deploy* ke perangkat android (Imaddudin Al Fikri, et al. 2016).

II.2.9. Pengertian UML

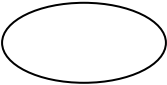
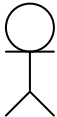

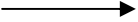
Menurut Sri Dharwiyanti dan Romi Satria Wahono (2003), UML adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

UML merupakan metode untuk merancang dan mengembangkan sebuah sistem juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

II. 2.9.1. Use Case Diagram

Use case Diagram 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. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016).

Tabel II.1. Use Case Diagram

Gambar	Arti	Keterangan
	<i>Use Case</i>	menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor	adalah Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi sikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i>
	Asosiasi tanpa panah	Asosiasi antara aktor dan <i>use case</i> digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi dengan panah	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem

<code><<include>></code>	<i>Include</i>	merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
<code><<extends>></code>	<i>Extend</i>	merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat



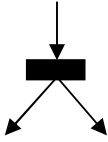
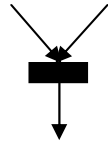
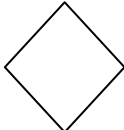
Sumber : Ade Hendini (2016), "Pemodelan UML Sistem Informasi Monitoring

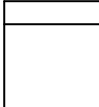
Penjualan dan Stok Barang".

II. 2.9.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016).

Tabel II.2. Activity Diagram

Gambar	Arti	Keterangan
	<i>Start Point</i>	diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i>	akhir aktivitas
	<i>Activities</i>	menggambarkan suatu proses/kegiatan bisnis
	<i>Fork/percabangan</i>	digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join (penggabungan) atau rake</i>	digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i>	menggambarkan pilihan untuk pengambilan keputusan, true atau false




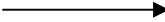
	<i>Swimlane</i>	pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa
---	-----------------	---

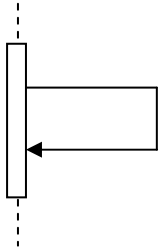

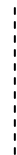
Sumber : Ade Hendini (2016), “Pemodelan UML system Informasi Monitoring Penjualan dan Stok Barang”.

II. 2.9.3. Sequence Diagram

Sequence diagram menggambarkan kelakuan obyek pada *use case* dengan mendeskripsikan waktu hidup obyek dan pesan yang dikirimkan dan diterima antar obyek (Ade Hendini, 2016).

Tabel II.3. Sequence Diagram

Gambar	Arti	Keterangan
	<i>Entity Class</i>	merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i>	berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form</i> cetak
	<i>Control class</i>	suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i>	simbol mengirim pesan antar class

	<i>Recursive</i>	menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i>	mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
	<i>Lifeline</i>	garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation

Sumber : Ade Hendini (2016), "Pemodelan UML system Informasi Monitoring

Penjualan dan Stok Barang".

II. 2.9.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan obyek yang dikoneksikan. *Class* diagram secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), dan *Visibility*, tingkat akses objek eksternal kepada suatu

operasi atau atribut. Hubungan antar Kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti (Ade Hendini, 2016).

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

Sumber : Ade Hendini (2016), “Pemodelan UML system Informasi Monitoring Penjualan dan Stok Barang”.