BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer atau suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi *(application suite)*. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antar muka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. (Ernita Sitohang, 2013).

II.2. SMS

SMS (*Short Message Service*) merupakan sebuah layanan komunikasi yang ada pada telepon seluler untuk mengirim dan menerima pesan – pesan pendek. SMS pertama kali dikenalkan pada tanggal 3 Desember 1982. SMS pertama di dunia dikirimkan menggunakan jaringan *GSM* milik operator telepon bernama *Vodafone*. SMS pertama ini dikirimkan oleh ahli bernama Neil Papwort kepada Richard Jarvis menggunakan komputer. (Dwi P, 2012).

II.3. Algoritma *Gronfeld Cipher*

Algoritma *Gronsfeld* adalah satu *cipher* subsitusi sederhana *polyalphabetic*. Gaspar Schot adalah seorang kriptografer abad ke 17 di Jerman, yang belajar *cipher* ini selama perjalanan antara Mainz dan Frankfurt dengan menghitung *Gronsfeld*, maka terciptalah nama dari *chipper* tersebut yaitu *gronsfeld*. *System gronsfeld* menggunakan suatu kunci numeric yang biasanya cukup pendek misalnya 7341, kunci ini diulang secara periodic, sesuai dengan jumlah kata *plaintext*. Idenya adalah dengan mengganti huruf dengan bilangan decimal maka akan mengakibatkan *plaintext* tidak akan berupa huruf melainkan hanya berupa susunan angka. Kemudian enkripsi menggunakan prinsip yang sama dengan algoritma *gronsfeld* yaitu menggunakan table yang hanya berukuran 10x10. (Azanuddin, 2013).

Algoritma enkripsi *gronsfeld cipher* :

Ci = (Pi + (Ki-48)) mod 256

Algoritma dekripsi *gronsfeld cipher* :

Pi = (Ci – (Ki-48)) mod 256

Contoh Proses Enkripsi :

Plaintext : GRO

Kunci : OSG

Solusi :

Ascii Plaintext :

G = 71

R = 82

O = 79

Key :

O = 79

S = 83

G = 71

C1 = (G + (k1-48)) mod 256

= (71 + (79-48)) mod 256

= 102 mod 256

= 102

C2 = (R + (k2-48)) mod 256

= (82 + (83-48)) mod 256

= 117 mod 256

= 117

C3 = (O + (k3-48)) mod 256

= (79 + (71-48)) mod 256

= 102 mod 256

= 102

Ascii Chipertext : fuf

Contoh Proses Dekripsi :

P1 = (f– (k1-48)) mod 256

= (102 - (79-48)) mod 256

= 71 mod 256

= 71 = G

P2 = (u – (k2-48)) mod 256

= (117 – (83-48)) mod 256

= 82 mod 256

= 82 = R

P3 = (f – (k3-48)) mod 256

= (102 – (71-48)) mod 256

= 79 mod 256

= 79 = O

Plaintext : GRO

II.4. *Android*

Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencangkup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [10]Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open sourcepada perangkat mobile. Di lain pihak, Google

merilis kode - kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler (Andi Juansyah, 2015).

II.5. Data

Data dapat didefenisikan sebagai kenyataan yang digambarkan oleh nilai, bilangan-bilangan, untaian karakter atau symbol-simbol yang membawa arti tertentu. Informasi sendiri dapat didefenisikan sebagai hasil dari pengolahan data dalam bentuk yang lebih berguna bagi penerimaanya, yang digunakan sebagai alat bantu dalam pengambilan. (Ernita Sitohang, 2013).

II.5.1. Keamanan

Keamanan merupakan salah satu aspek terpenting dari sebuah system informasi. Masalah keamanan sering kurang mendapatkan perhatian dari para perancang dan pengelola sistem informasi. Masalah keamanan sering berada diurutan setelah tampilan, atau bahkan diurutan terakhir dalam daftar hal-hal yang dianggap penting. Keamanan adalah keadaan bebas dari bahaya. Istilah ini dapat digunakan dengan hubungan kepada kejahatan, dan segala bentuk kecelakaan. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap serangan teroris, keamanan komputer terhadap *hacker*, keamanan rumah terhadap maling dan penyelusup lainnya, keamanan financial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya. *Host* komputer yang terhubung ke *network*, mempunyai ancaman keamanan lebih besar dari pada host yang tidak berhubungan kemana-mana. Dengan mengendalikan network *security* resiko tersebut dapat dikurangi. (Ernita Sitohang, 2013).

II.5.2. Definisi Keamanan Data

Keamanan data dan informasi terdiri dari perlindungan terhadap aspek-aspek berikut :

1. *Confidentiality* (kerahasiaan) aspek yang menjamin kerahasiaan data atau informasi, memastikan bahwa informasi hanya dapat diakses oleh orang yang berwenang dan menjamin kerahasiaan data yang dikirim, diterima dan disimpan.

2. *Integrity* (integritas) aspek yang menjamin bahwa data tidak dirubah tanpa ada ijin pihak yang berwenang (*authorized),* menjaga keakuratan dan keutuhan informasi serta metode prosesnya untuk menjamin aspek *intefrity* ini.

3. *Availability* (ketersediaan) aspek yang menjamin bahwa data akan tersedia saat dibutuhkan, memastikan *user* yang berhak dapat menggunakan informasi dan perangkat terkait (aset yang berhubungan bilamana diperlukan).

Keamanan data dan informasi diperoleh dengan mengimplementasi seperangkat alat control yang layak, yang dapat berupa kebijakan-kebijakan, praktek-praktek, prosedur-prosedur, struktur-stuktur organisasi dan piranti lunak. (Ernita Sitohang, 2013).

II.6. Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam

mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. (Abdul Kadir, 2014).

II.6.1. Elemen Sistem

Elemen-elemen yang membentuk sebuah sistem yaitu :

a. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem lain berbeda-beda. Begitu pula yang berlau pada sistem informasi. Setiap sistem informasi memiliki suatu tujuan, tetapi denga tujuan yang berbeda-beda. Walaupun begitu, tujuan utama yang umum ada tiga macam, yaitu :

1. Untuk mendukung fungsi kepengurusan manajemen,
2. Untuk mendukung pengambilan keputusan manajemen,
3. Untuk mendukung kegiatan operasi perusahaan.

b. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk kedalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa dari pelanggan). Pada sistem informasi, masukan dapat berupa data transaksi, dan data non-transaksi (misalnya, surat pemberitahuan), serta instruksi.

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bias berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien. Pada sistem informasi, proses dapat berupa suatu tindakan yang bermacam-macam. Meringkas data, melakukan perhitungan, dan mengurutkan data merupakan beberapa contoh proses.

d. Keluaran

Keluaran (*output*) merupakan hasi dari pemrosesan. Pada sistem informasi, keluaran bias berupa suatu informasi, saran, cetakan laporan, dan sebagainya. (Abdul Kadir, 2014).

II.7. Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian kriptografi modern kriptografi adalah ilmu yang bersabdarkaan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. (Rifki Sadikin, 2012).

1. Fungsi *Hash*

Fungsi *hash* adalah fungsi yang melakukan pemetaan pesan dengan panjang sembarang ke sebuah teks khusus yang disebut *message digest* dengan panjang tetap. Fungsi *hash* umumnya dipakai sebagai nilai uji (*check value*) pada mekanisme keutuhan data. (Rifki Sadikin, 2012)

2. Penyandian Dengan Kunci Simetrik

Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernialai sama. Kunci pada penyandian simetrik diasumsikan bersifat rahasia hanya pihak yang melakukan enkripsi dan dekripsi yang mengetahui nilainya. Oleh karena itu penyandian dengan kunci simetrik disebut juga penyandian dengan kunci rahasia *secret key encipherment*. (Rifki Sadikin, 2012)

3. Penyandian dengan kunci asimetrik

Penyandian dengan kunci asimetrik atau sering disebut juga dengan penyandian kunci publik (*public key*) adalah penyandian dengan kunci enkripsi dan dekripsi berbeda nilai. Kunci enkripsi yang juga disebut dengan kunci publik (*public key*) bersifat terbuka. Sedangkan, kunci dekripsi yang juga disebut kunci privat (*private key*) bersifat tertutup/ rahasia. (Rifki Sadikin, 2012).

II.7.1. Sistem Kriptografi

Sistem kriptografi terdiri dari 5 bagian yaitu :

1. *Plaintext*

Pesan atau data dalam bentuk aslinya yang dapat terbaca. *Plaintext* adalah masukan bagi algoritma enkripsi. (Rifki Sadikin, 2012).

2. *Secret Key*

Secret key yang juga merupakan masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi. (Rifki Sadikin, 2012).

3. *Ciphertext*

Ciphertexti adalah keluaran algoritma enkripsi. *Ciphertext* dapat dianggap sebagai pesan dalam bentuk tersembunyi. Algoritma enkripsi yang baik akan menghasilkan *ciphertext* yang terlihat acak. (Rifki Sadikin, 2012).

4. Algoritma Enkripsi

Algortima enkripsi memiliki 2 masukan teks asli dan kunci rahasia. Algoritma enkripsi melakukan transformasi terhadap teks asli sehingga menghasilkan teks sandi. (Rifki Sadikin, 2012).

5. Algoritma Dekripsi

Algoritma dekripsi memiliki 2 masukan yaitu teks sandi dan kunci rahasia. Algoritma dekripsi memulihkan kembali teks sandi menjadi teks asli bila kunci rahasia yang dipakai algoritma dekripsi sama dengankunci rahasia yang dipakai algoritma enkripsi. (Rifki Sadikin, 2012).

II.7.2. Karakteristik Sistem Kriptografi

Sistem kriptografi dapat dikarakteristikan berdasarkan :

1. Tipe Operasi Dipakai Dalam Enkripsi Dan Dekripsi

Dua tipe operasi yang dipakai dala enkripsi dan dekripsi : subsitusi, elemen pada pesan (karakter, *byte* atau *bit*) ditukar/ disubsitusikan dengan elemen lain dari ruang pesan. Misalnya subsitusi sederhana A ditukar B, B ditukar D, dan C ditukar Z, pesan ”BACA” menjadi ”DBZB”. Tipe operasi lainnya adalah transposisi, elemen pada pesan berpindah posisi misalnya posisi 1 menjadi posisi 4 dan posisi 2 menjadi posisi 3, posisi 3 menjadi posisi 1 dan posisi 4 menjadi posisi 2, pesan ”KAMI” menjadi ”MAIK”. Sistem kriptografi modern mencakup kedua tipe operasi ini. (Rifki Sadikin, 2012).

2. Tipe Kunci Yang Dipakai

Umumnya sistem kriptografi klasik dan beberapa sistem kriptografi modern menggunakan kunci yang sama pada sisi penyandi dan penyulih sandi. Sistem kriptografi seperti ini disebut kriptografi dengan kunci simetri. Baru pada tahun 1976 sistem kriptografi yang memperbolehkan kunci yang tidak sama diusulkan oleh Whitfield Deffie dan Martin Hellman, (Deffie & Hellman, 1976). Sistem kriptografi ini disebut dengan kriptografi dengan kunci asimetri. (Rifki Sadikin, 2012).

3. Tipe Pengolahan Pesan

Ketika melakukan penyandian pesan yang akan dienkripsi ataupun didekripsi diolah pesatuan blok elemen disebut dengan *block cipher*. Cara lain adalah dengan menganggap masukan untuk enkripsi dan dekripsi sebagai aliran elemen secara terus menerus disebut dengan *stream cipher*. (Rifki Sadikin, 2012).

II.8. *Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer maupun telepon genggam. Bahasa pemrograman ini dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, di mana saat ini merupakan bagian dari Oracle yang dirilis pada tahun1995. Bahasa ini banyak mengadopsi sintaks yang terdapat pada C dan C++, tetapi dengan sintaksis mode objek yang lebih sederhana. *Java* merupakan bahasa pemrograman yang bersifat umum/nonspesifik dan secara khusus didesain untuk memanfaatkan implementasi semaksimal mungkin. Fungsi *Java* memungkinkan aplikasi *Java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda. *Java* dikenal pula dengan slogannya “tulis sekali, jalankan di mana pun”. Saat ini *Java* secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*. (Wahana Komputer, 2015 : 2).

II.9. *Unified Modeling Language* (*UML*)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (*UML*). *UML* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

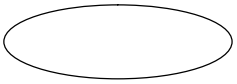
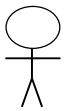
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015).

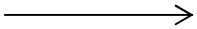
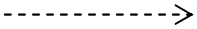
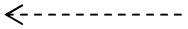
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *UML* adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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 di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 sebagai berikut :

Tabel II.1. Simbol *Use Case*

Gambar	**Keterangan**
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks

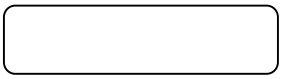
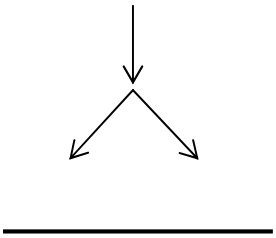
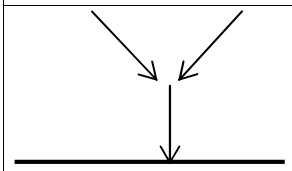
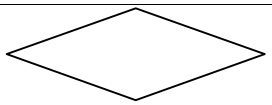
	target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 sebagai berikut :

Tabel II.2. Simbol *Activity Diagram*

Gambar	**Keterangan**
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*

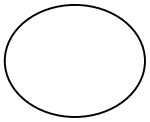
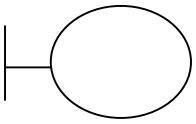
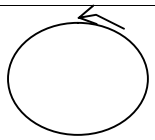
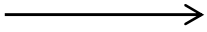
New Swimline	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

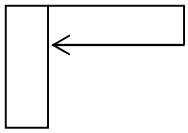
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 sebagai berikut :

Tabel II.3. Simbol *Sequence Diagram*

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

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

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility,* tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 sebagai berikut :

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 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015)