BAB II

TINJAUAN PUSTAKA

II.1 Sistem Keamanan Data

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama informasi sensitif yang hanya boleh diketahui oleh pihak yang berhak saja. Informasi yang merupakan hasil pengolahan dari data, mempunyai nilai yang berbeda lagi setiap orang. Seringkali sebuah informasi menjadi sangat berharga dan tidak semua orang diperkenankan untuk mengetahuinya, namun selalu saja ada pihak yang berusaha untuk mengetahui informasi dengan cara-cara yang tidak semestinya bahkan bermaksud untuk merusaknya. (sumber : Harry Abdulrachman, Erwin Gunadhi : 2015 : 23)

Keamanan merupakan komponen yang vital dalam komunikasi data elektronis. Masih banyak yang belum menyadari bahwa keamanan (*Securyti*) merupakan sebuah komponen penting yang tidak murah. Teknologi kriptografi sangat berperan juga dalam proses komunikasi yang digunakan untuk melakukan enkripsi (pengacakan) data yang ditransaksikan selama perjalanan dari sumber ke tujuan dan juga melakukan dekripsi (menyusun kembali) data yang telah teracak tersebut. Berbagai sistem yang telah dikembangkan adalah seperti sistem *private key* dan *public key*. Penguasaan algoritma-algoritma populer digunakan untuk mengamankan data jaga sangat penting. Contoh-contoh algoritma ini antara lain : DES, IDEA, RC5, RSA dan ECC (*Elliptic Curve Cryptography*). (sumber : Febriansyah : 2012 : 23)

Dari sisi tindakan pihak yang bertanggung jawab, keamanan jaringan komputer terbagi dua level, yaitu :

1. Keamana fisik peralatan mulai dari server, terminal / client router sampai dengan *cabling*.

2. Keamanan sistem data sekitarnya ada penyeludup yang berhasil mendapatkan akses ke seluruh fisik jaringan komputer.

II.2 SMS (*Short Messaging Service*)

SMS merupakan salah satu media yang paling banyak digunakan sekarang ini dikarenakan murah dan prosesnya cepat, langsung kepada tujuan. SMS merupakan salah satu fitur di GSM yang dikembangkan dan distandarisasi oleh ETSI. Meskipun telah banyak fitur-fitur dari GSM seperti SMS, MMS dan GPRS keberadaan jasa dan industri yang menggunakan SMS khususnya semakin lama semakin banyak dijumpai. Hal itu juga didukung oleh faktor *hardware* yang semakin hari semakin terjangkau.

SMS merupakan sebuah sistem pengiriman data dalam paket dengan bandwitch kecil. Dengan karakteristik ini, pengiriman suatu data yang dapat dilakukan dengan efisiensi yang sangat tinggi. Pada awalnya SMS diciptakan untuk menggantikan layanan paging dengan menyediakan layanan serupa yang bersifat *two-way messaging* ditambah dengan *notification service*, khusunya untuk *voice mail*. Pada perkembangan selanjutnya, muncul jenis-jenis layanan lain seperti *mail, fax* dan *pagging intergration, interactive banking, information service*, dan integrasi dengan aplikasi berbasis internet. Selain itu juga berkembang layanan wireless seperti *SIM download for active, debet*, dan *profile editing, Wireless Point of Sale* (WPS). Serta layanan aplikasi lapangan seperti *remote reasing, remote sensing dan location Base Service* (LBS). Integerasi dengan aplikasi berbasis internet mendorong timbulnya layanan seperti *web-based messaging, gamming* dan *chatting*.

Layanan SMS juga memungkinkan pengiriman pesan dalam bentuk *alphanumeric*, layanan SMS ini banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*). Teknologi

wireless dipelopori dari kawasan eropa yang diawali pada kebutuhan bersama terhadap satu sistem jaringan baru yang dapat menjdai standart jaringan yang berlaku dan dapat diterapkan di seluruh kawasan eropa. Dalam sistem baru juga harus terdapat kemampuan yang dapat mengantisipasi mobilitas pengguna untuk menampung penambahan jumlah pelanggan baru.

Pada tahun 1992, dilakukan pengiriman pesan menggunakan SMS dari sebuah *Personal Computer* (PC) ke telepon *mobile* dalam jaringan GSM milik Vodafone Inggris, kemudian merambah ke beuna amerika yang di pelopori oleh beberapa operator komunikasi *mobile* berbasis digital seperti *BellSouth Mobility, Primeco* dan operator lainya. Teknologi yang digunakan dari pengirim SMS yaitu *Store and forwad service*.(Sumber : Hinova rezha U : 2013 : 12)

II. 3 Sejara Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *cryptos* yang artinya "*secret*" (yang tersembunyi) dan gráphein yang artinya "*writting*" (tulisan). Jadi, kriptografi berarti "*secret writting*" (tulisan rahasia). Definisi yang dikemukakan oleh Schneier (1996), Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. (Emy Setyaningsih,S.Si, M.Kom, 2015 : 8)

II.3.1. Terminologi Kriptografi

Ada beberapa istilah-istilah yang penting dalam kriptografi, yaitu :

1. Pesan (*Plaintext dan Ciphertext*) : Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Pesan asli disebut plainteks (*plaintext*) anda teks-jelas

(*cleartext*). Sedangkan pesan yang sudah disandikan disebut cipherteks (*chipertext*).

2. Pengirim dan Penerima : Komunikasi data melibatkan pertukaran pesan antaradua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.
3. Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan.
4. Kriptanalisis dan Kriptologi : Kriptanalisis (*cryptanalysis*) adalah suatu ilmu dan seni membuka (*breaking*) *ciphertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis . kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisi.
5. Enkripsi dan Dekripsi : Proses untuk menyandikan plaintext menjadi ciphertext disebut enkripsi (*encryption*). Sedangkan proses untuk memperoleh kembali plaintext dari ciphertext disebut dekripsi (*decryption*).
6. Chiper dan Kunci : algoritma matematis untuk menyadikan plaintext menjadi ciphertext. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *encriphering* dan *dechipering*. (Emy Setyaningsih, S.Si, M.Kom : 2015 : 8)

II.4 Algoritma Kriptografi

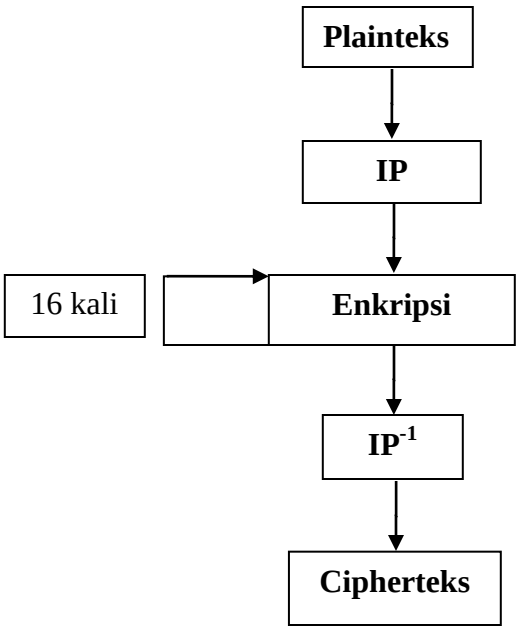
Algoritma-algoritma kriptografi dapat dibedakan menjadi dua macam yaitu simetrik dan asimetrik. Algoritama yang digunakan satu kunci untuk proses enkripsi dan dekripsi data. Sedangkan algoritma asimentrik (model enkripsi kunci public) menggunakan kunci yang berbeda dalam proses enkripsi dan dekrispi pesan. (Sumber : I Gede Andika Putra : 2012 : 30)

II.5 Algoritma DES (*Data Encryption Standard*)

Algoritma DES merupakan algoritma enkripsi yang paling banyak digunakan di dunia yang diadopsi oleh NIST (*National Institue of Standards and Technology*) sebagai standar pengolah informasi Federal AS. Sejara DES dimulai dari pemerintah pemerintah amerika serikat untuk memasukkan proposal enkripsi. Algoritma DES dibuat di IBM dan merupakan modifikasi dari algoritma terdahulu yang bernama Lucifer. Lucifer merupakan algoritma *cipher blok* yang beroperasi pada blok masukan 64 bit dan kuncinya berukuran 18 bit. Pengurangan jumlah bit kunci pada DES dilakukan dengan alasan agar mekanisme algoritma ini bias diimplementasikan dalam satu *chip*. Hosrt Feistel merupakan salah satu periset yang mula-mula mengembangkan DES ketika bekerja di IBM Watson Laboratory di Yorktown Heights, New York.

Algoritma DES terbagi menjadi tiga kelompok, yaitu pemrosesan kunci, enkripsi data 64 bit, dan dekripsi data 64 bit, dimana kelompok yang satu dengan yang lain saling berinteraksi dan terkait. Algoritma DES dirancang untuk mengenkripsi dan mendekripsi data dalam blok data yang terdiri atas 64 bit dibawah kontrol kunci 64 bit. Dekripsi data harus dikerjakan menggunakan kunci yang sama dengan yang dipakai untuk mengenkripsi data, dengan penjadwalan alamat kunci bit yang diubah sehingga proses membaca merupakan kebalikan dari proses menulis.

Dekripsi mengekripsikan 64 bit plaintext menjadi 64 bit ciphertext dengan menggunakan 56 bit kunci internal yang dibangkitkan dari 64 bit kunci eksternal. Kunci ekternal merupakan kunci yang dimasukkan oleh pengguna pada sistem, sedangkan kunci internal merupakan kunci yang digunakan untuk melakukan enkripsi pada setiap putaran DES (ada 16 putaran) yang diperoleh dari kunci eksternal yang telah diproses. (Emy Setyaningsih, S.Si, M.Kom : 2015 :)



Gambar II.1. Skema Global Algoritma DES

Skema global algoritma DES seperti diperlihatkan pada Gambar II.1 adalah sebagai berikut :

1. *Blok plaintext* dipemutasi dengan matriks permutasi awal (*initial permutation atau IP*). Plaintext merupakan data yang akan dienkripsi. Plaintext ini direpresentasikan sebagai bit, misalnya huruf P direpresentasikan sebagai 01010000. Keseluruhan plaintext dibagi ke dalam blok-blok. Setiap blok terdiri atas 64 bit.
2. Hasil permutasi awal kemudian dienkripsi sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enkripsi kemudian dipermutasikan dengan matriks permutasi balikan (*inverse initial permutation atau* IP^{-1}) menjadi *blok ciphertext.*

II.6 Algoritma AES (*Anvanced Encryption Standard*)

Pada bulan Januari 1997 inisiatif AES diumumkan dan pada bulan September 1997 publik diundang untuk mengajukan proposal *block cipher* yang cocok sebagai kandidat untuk AES. Pada tahun 1999 NIST mengumumkan lima kandidat finalis yaitu MARS, RC6, Rijndael, Serpent, dan Twofish. Algoritma AES dipilih pada bulan Oktober 2001 dan standarnya diperkenalkan pada bulan November 2002. Algoritma AES (*Advanced Encryption Standard*) adalah algoritma *cipher* blok yang menggunakan teknik subtitusi, permutasi dari sejumlah putaran pada setiap blok yang akan dienkripsi. Sistem permutasi dan subtitusi (*S-box)* yang diguanakan pada EAS tidak menggunakan jaringan *Feistel* sebagaimana *cipher blok* pada umumnya.(Emy Setyaningsih, S.Si, M.Kom : 2015 : 168)

Algoritma Rijndael dirancang untuk memiliki properti berikut :

1. Ketahanan terhadap semua jenis serangan yang diketahui.
2. Kesederhanaan rancangan.
3. Keko]mpakan kode dan kecepatan pada berbagai *platform*.

AES terbagi dalam tiga jenis yaitu :

1. AES-128
2. AES-192
3. AES-256

Tabel : II.1 Perbandingan jumlah round dan kunci
(Sumber : Wibowo dan Wihartantyo, 2004)

	Jumlah Kunci (Nk)	Besar Blok (Nb)	Jumlah Round (Nr)

AES-128	16 *byte*	16 *byte*	10
AES-129	24 *byte*	16 *byte*	12
AES-256	32 *byte*	16 *byte*	14

II.7 UML (*Unified Modeling Language*)

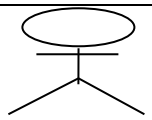
Diagram *Unified Modeling Language* (UML) adalah menjadi diagram yang populer dan telah banyak digunakan oleh para pengembang sistem berorientasi objek. Saat ini, telah banyak perangkat lunak yang membantu membuat model sistem dengan metodologi berbasis objek. (Hamim Tohari ; 2014 : 22) .

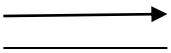
Adapun jenis – jenis dari tipe diagram UML adalah sebagai berikut :

II.7 .1. *Use Case Diagram*

Merupakan rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. sebuah *Use Case Diagram* merepresentasikan sebuah interaksi antara actor dengan sistem. *Use Case Diagram* menyatakan sebuah aktivitas atas pekerjaan tertentu. Simbol yang digunakan dalam *Use Case Diagram* ditunjukkan oleh tabel II.

Tabel II.2. Simbol *Use Case Diagram*
(Sumber : Hamim Tohari : 2014 : 49)

Simbol	Keterangan
	*Actor* dapat berupa manusia, sistem atau *device* yang memliki peranan

	dalam keberhasilan operasi dari sistem.
	*Use case* mengidentifikasikan fitur kunci dari sistem. Tanpa *use case* sistem tidak akan memenuhi permintaan *user*/*actor*. Setiap *use case* mengekspesikan goal-nya dan digambarkan dengan *elips*.
	*Assosiation* mengidentifikasikan interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu.

II.7 .2. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi yang mengasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Class menggambarkan keadaan (*atribut/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Dalam pemodelan statis dari sebuah sistem, *class diagram* biasanya digunakan untuk memodelkan salah satu dari tiga hal tersebut :

a. Perbendaharaan dari sistem
b. Kolaborasi
c. Skema basis data *logical*

Kelas memiliki tiga area pokok :

a. Nama dan *stereotype*

b. Atribut

c. Metode dan operasi

Atribut dan metode dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan

b. *Proctected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak- anak yang mewarisinya

c. *Public*, dapat dipanggil oleh siapa saja

Class dapat berupa implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metode pada saat *run-time*.

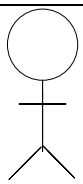
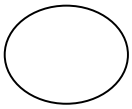
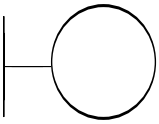
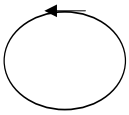
Tabel II.3 *Class Diagram*
(Sumber : Hamim Tohari : 2014 : 86)

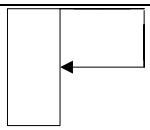
Indikator	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nola tau lebih
1..*	Satu atau lebih
N	Hanya n (dengan n > 1)
0..n	Nol sampai n (dengan n > 1)
1..n	Satu sampai n (dengan n > 1)

II.7 .3. *Sequene Diagram*

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirm antara objek juga interaksi antara objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Hamim Tohari : 2014 : 101)

Tabel II.4 Simbol *Sequence Diagram*
(Sumber : Gellysa Urva dkk : 2015 : 95)

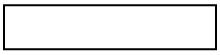
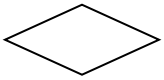
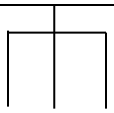
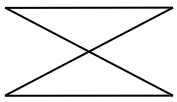
Gambar	Keterangan
	*Actor* dapat berupa manusia, sistem atau *device* yang memliki peranan dalam keberhasilan operasi dari sistem.
	*Entity Class* merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	*Boundary Class* berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih actor dengan sistem.
	*Control Class* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
	*Message* simbol mengirim pesan antar *class*.

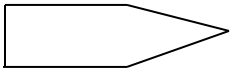
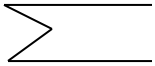
	*Recursive* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi .
	Lifeline yaitu garis titik titik yang terhubung dengan objek, sepanjang *lifeline* terdapat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

II.7 .4. *Activity Diagram*

Activity Diagram memodelkan workflow proses bisnis urutan dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip flowchart karena memodelkan workflow dai satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status .Membuat *Activity Diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity Diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*.

Tabel II.5. *Activity Diagram*
(Sumber : Hamim Tohari ; 2014)

Simbol	Keterangan
	Titik awal (*start*)
	Titik akhir (*akhir*)
	Activity
	Pilihan untuk pengambilan keputusan
	*Fork* : digunakan untuk menunjukkan kegiatan yang dilakukan secara 32 cenario atau untuk menggabungkan dua kegiatan 32 cenario menjadi satu.
	*Rake* : menunjukkan adanya dekomposisi
	Tanda waktu

	Tanda pengiriman
	Tanda penerima
	Aliran akhir (*Flow Final*)

II.8 Eclipse

Eclipse adalah sebuah IDE(*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent).* Berikut ini adalah sifat dari Eclipse :

1. *Multi-platform*: Target sistem operasi Eclipse adalah *Microsoft Windows, Linux,Solaris, AIX, HP-UXdan Mac OS X.*
2. *Mulit-language:* Eclipse dikembangkan dengan bahasa pemrograman *Java*, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti *C/C++*, *Cobol*, *Python*, *Perl*, *PHP*, dan lain sebagainya.
3. *Multi-role:* Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah dipasang (*diinstal*). Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform*(RCP). Berikut ini adalah komponen yang membentuk RCP:

- *Core platfor*
- *OSGi*

- SWT(*Standard Widget Toolkit*)
- J*Face*
- *Eclipse Workbench*

Eclipse selalu dilengkapi dengan JDT(*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program *Java*, dan PDE(*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. (Sumber : Wina Noviani Fatimah, ST : 2011 : 2)

II.9 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup 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 piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handesti Alliance*, konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.* Pada saat perilisan perdana *android*, 5 November 2007, *Android* bersama *Open Handest Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *android* dibawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. (Safaat H, Nazruddin, 2012)

Aplikasi *Android* ditulis dalam bahasa pemrograman *java*, yaitu kode *java* yang terkompilasi bersama-sama dengan data dan *file-file* sumber yang dibutuhkan oleh aplikasi yang digabungkan oleh *app tools* menjadi paket aplikasi *Android*, sebuah file yang ditandai dengan

akhiran *.apk.* file inilah yang didistribusikan sebagai aplikasi dan diinstal pada *handset Android.*(

Sumber : Ahmad : 2015 :191)