BAB II
KAJIAN PUSTAKA & DASAR TEORI

2.1 Pengertian Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa yunani "*cryptos*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan). Jadi, kriptografi berarti "*secret writing*" (tulisan rahasia).Ada beberapa definisi kriptografi yang telah dikemukakan di dalam beberapa literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat, dan mata-mata.Namun saat ini kriptografi lebih dari sekedar privasi, tetapi juga tujuan *data integrity*, *aunthetication*, dan *non-repudiation* (Munir, 2006).

2.2 Tujuan Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu (munir, 2006) :

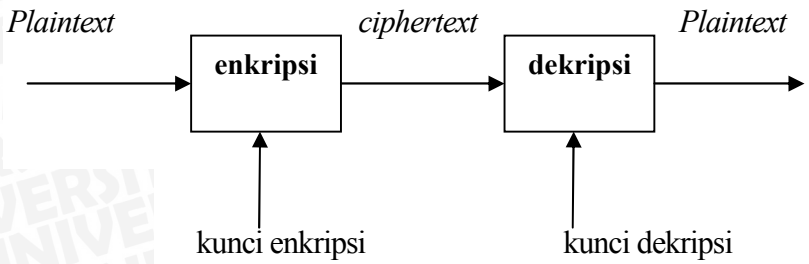
1. *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetaprahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatualgoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
2. *Data integrity* (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan)data yang tidak sah (oleh pihak lain).
3. *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
4. *Non-repudiation* (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

Seiring dengan berkembangnya jaman , kriptografi menitikberatkan kekuatan pada kerahasiaan algoritma yang digunakan (yang artinya apabila algoritma yang digunakan telah diketahui maka pesan sudah jelas tersebar dan dapat diketahui isinya oleh siapa saja yang mengetahui algoritma tersebut), kriptografi modern lebih menitikberatkan pada kerahasiaan kunci yang digunakan pada algoritma tersebut (oleh pemakainya) sehingga algoritma tersebut dapat saja disebarkan ke kalangan masyarakat tanpa takut kehilangan kerahasiaan bagi para pemakainya. (munir, 2006)

Berikut adalah istilah-istilah yang digunakan dalam bidang kriptografi :

1. ***Plaintext*** adalah pesan yang hendak dikirimkan (berisi data asli).
2. ***Ciphertext*** adalah pesan tersandi yang merupakan hasil enkripsi.
3. **Enkripsi** adalah proses pengubahan *plaintext* menjadi *ciphertext*.
4. **Dekripsi** adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext*, sehingga berupa data awal/asli.
5. **Kunci** adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses enkripsi dan dekripsi.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah *plaintext* menjadi *ciphertext* (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti. Gambaran langkah-langkah kriptografi modern akan ditunjukkan gambar 2.1.(munir , 2006)



Gambar 2.1 Langkah-langkah kriptografi modern

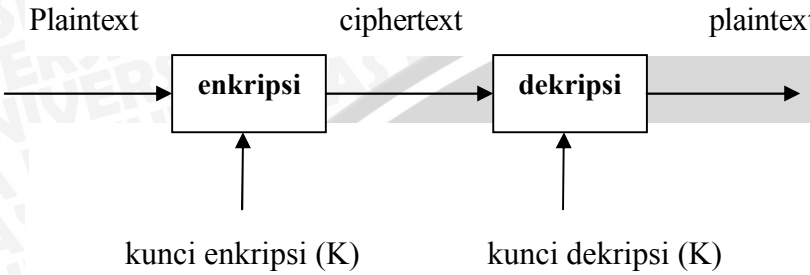
Sumber : kajian pustaka

Peranan kunci sangatlah penting dalam proses enkripsi dan dekripsi (disamping pula algoritma yang digunakan) sehingga kerahasiaannya sangatlah penting, apabila kerahasiaannya terbongkar, maka isi dari pesan dapat diketahui.

2.3 Algoritma Kriptografi

2.3.1 Algoritma Simetris

Algoritma simetris *(symmetric algorithm)* adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key algorithm.* Gambar diagram proses enkripsi dan dekripsi algoritma simetris ditunjukkan oleh gambar 2.2.



Gambar 2.2 Diagram proses enkripsi dan dekripsi algoritma simetris
Sumber : kajian pustaka

Sebelum melakukan pengiriman pesan, pengirim dan penerima harus memilih suatu suatu kunci tertentu yang sama untuk dipakai bersama, dan kunci ini haruslah rahasia bagi pihak yang tidak berkepentingan sehingga algoritma ini disebut juga algoritma kunci rahasia *(secret-key algorithm).*

Kelebihan :

1. Kecepatan operasi lebih tinggi bila dibandingkan dengan algoritma asimetrik.
2. Karena kecepatannya yang cukup tinggi, maka dapat digunakan pada sistem*real-time*

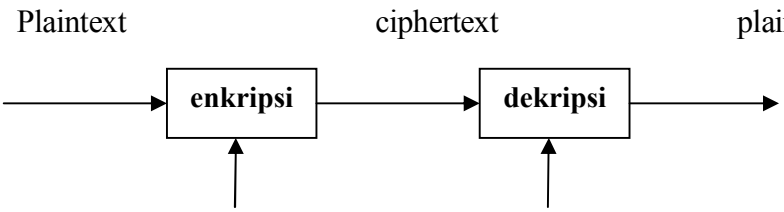
Kelemahan :

1. Untuk tiap pengiriman pesan dengan pengguna yang berbeda dibutuhkankunci yang berbeda juga, sehingga akan terjadi kesulitan dalam manajemenkunci tersebut.
2. Permasalahan dalam pengiriman kunci itu sendiri yang disebut *"keydistribution problem"*

Contoh algoritma :*TwoFish, Rijndael, Camelli*

2.3.2 Algoritma Asimetris

Algoritma asimetris *(asymmetric algorithm)* adalah suatu algoritma dimana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi. Pada algoritma ini menggunakan dua kunci yakni kunci publik *(public key)* dan kunci privat *(private key).*Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan. Diagram proses enkripsi dan dekripsi algoritma asimetris ditunjukkan gambar 2.3



Gambar 2.3 Diagram proses enkripsi dan dekripsi algoritma asimetris

Sumber : kajian pustaka

Pada umumnya kunci publik *(public key)* digunakan sebagai kunci enkripsi sementara kunci privat *(private key)* digunakan sebagai kunci dekripsi.

Kelebihan :

1. Masalah keamanan pada distribusi kunci dapat lebih baik
2. Masalah manajemen kunci yang lebih baik karena jumlah kunci yang lebihsedikit

Kelemahan :

1. Kecepatan yang lebih rendah bila dibandingkan dengan algoritma simetris
2. Untuk tingkat keamanan sama, kunci yang digunakan lebih panjangdibandingkan dengan algoritma simetris.

Contoh algoritma : *RSA, DSA, ElGamal*

2.3.3 Block Cipher dan Stream Cipher

Menurut Buchmann (2000) berdasarkan ukuran serta format data yang akan diproses, maka algoritma kriptografi dapat dibagi menjadi dua bagian yang utama yaitu :

1. ***Block Cipher*****,** algoritma kriptografi ini bekerja pada suatu data yang berbentuk blok/kelompok data dengan panjang data tertentu (dalam beberapa byte), jadi dalam sekali proses enkripsi atau dekripsi data yang masuk mempunyai ukuran yang sama.
2. ***Stream cipher*****,** algoritma yang dalam operasinya bekerja dalam suatu pesan berupa bit tunggal atau terkadang dalam suatu byte, jadi format databerupa aliran dari bit untuk kemudian mengalami proses enkripsi dandekripsi.

Pada algoritma penyandian blok *(block cipher),* plainteks yang masuk akan diproses dengan panjang blok yang tetap yaitu *n*, namun terkadang jika ukuran data ini terlalu panjang maka dilakukan pemecahan dalam bentuk blok yang lebih kecil. Jika dalam pemecahan dihasilkan

blok data yang kurang dari jumlah data dalam blok maka akan dilakukan proses *pading* (penambahan beberapa bit).

2.4 Algoritma *Rijndael*

Menurut Rijmen (2004), *Rijndael* adalah algoritma yang beroperasi dalam byte, bukan dalam bit. Algoritma ini mampu melakukan enkripsi terhadap *plaintext* sebesar 16 byte atau 128 bit. Algoritma *Rijndael* juga melakukan putaran enkripsi (*enciphering*) sebanyak 10 putaran namun bukan putaran yang merupakan jaringan Feistel.

Panjang kunci untuk keamanan dari sebuah teknik penyandian tergantung dari dua hal yaitu algoritma penyandian dan panjang kunci (key). Algoritma *Rijndael* sangat menentukan kekuatan dari sebuah teknik penyandian, tetapi panjang kunci juga tidak kalah penting dalam menentukan kekuatan sebuah teknik penyandian. Dengan melihat situasi ini, maka kriptografi yang baik akan memilih untuk menggunakan sepanjang mungkin kunci yang akan digunakan, namun hal ini tidak dapat diterapkan begitu saja. Semakin panjang kunci, semakin lama pula waktu yang digunakan oleh komputer untuk melakukan proses enkripsi. Oleh sebab itu, panjang kunci yang akan digunakan hendaknya memperhatikan tiga hal, yaitu seberapa penting data yang akan dirahasiakan, berapa lama waktu yang dibutuhkan agar data tersebut tetap aman, dan seberapa kuat kemampuan kriptanalis dalam memecahkan teknik penyandian kita. Untuk algoritma *Rijndael*ini sendiri memakai kunci dengan panjang 128 bit karena panjang kunci ini dianggap paling optimal untuk saat ini.

Garis besar algoritma *rijndael* yang beroperasi pada blok 128 bit dengan kunci 128 bit adalah sebagai berikut(Rijmen , 2004) :

1. AddRoundKey, melakukan XOR antara plainteks dengan cipher key.
2. Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran adalah :
   a. SubBytes adalah substitusi byte dengan menggunakan tabel substitusi (S-Box).
   b. ShiftRows adalah pergeseran baris-baris array state secara wrapping.
   c. MixColumns adalah mengacak data di masing-masing kolom array state.
   d. AddRoundKey adalah melakukan XOR antara state sekarang dengan round key.
3. Final round, proses untuk putaran terakhir :
   a. SubBytes
   b. ShiftRows
   c. AddRoundKey

Sedangkan pada proses dekripsi yang dilakukan didapatkan hasil dari enkripsi tersebut dengan langkah-langkah sebagai berikut : (Rijmen , 2004):

1. AddRoundKey, melakukan penambahan *Round key* dengan *cipher key* atau *key schedule* ke-10
2. Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran adalah :
   e. Invers SubBytes adalah substitusi byte dengan menggunakan tabel substitusi (S-Box).
   f. Invers ShiftRows adalah pergeseran baris-baris array state secara wrapping.
   g. Invers MixColumns adalah mengacak data di masing-masing kolom array state.
   h. Invers AddRoundKey adalah melakukan XOR antara state sekarang dengan hasil key schedule ke-10.
3. Final round, proses untutk putaran terakhir
   a. Invers SubBytes
   b. Invers Shifrows
   c. Invers AddRoundKey

2.5 Avallanche Effect

Salah satu karakteristik untuk menentukan baik atau tidaknya suatu algorithma kriptografi adalah dengan melihat *avalancheeffect*-nya.

Avallanceeffect merupakan suatu karakteristik dimana perubahan yang kecil terhadap *plaintext* maupun *key* akan menyebabkan perubahan yang signifikan terhadap *ciphertext* yang dihasilkan. Atau bisa juga diartikan bahwa perubahan satu *bit* pada *plaintext* maupun *key* akan menghasilkan perubahan banyak *bit*pada *ciphertext*.

Menurut Bruce Schneir dalam "*AppliedCryptogrphy*", bahwa suatu *avallanceeffect* dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45–60% (50% adalah hasil yang sangat baik). Hal ini menyebabkan perbedaan yang cukup sulit bagi *cryptanalyst* untuk melakukan serangan.

Rumus Avalance Effect sebagai berikut:

$$AE = \frac{Jumlahbitbeda}{Jumlahtotalbit} x100\%$$

2.6 *Exhaustive Effect*

Percobaan yang dibuat untuk mengungkap plainteks atau kunci dengan mencoba semua kemungkinan kunci (*trial and error*).

Asumsi yang digunakan:

a. Kriptanalis mengetahui algoritma kriptografi

b. Kriptanalis memiliki sebagian plainteks dan chiperteks yang bersesuaian.

Caranya: plainteks yang diketahui dienkripsikan dengan setiap kemungkinan kunci, dan hasilnya dibandingkan dengan chiperteks yang bersesuaian. Jika hanya chiperteks yang tersedia, chiperteks tersebut didekripsi dengan dengan setiap kemungkinan kunci dan plainteks hasilnya diperiksa apakah mengandung makna.

Misalkan sebuah sistem kriptografi membutuhkan kunci yang panjangnya 8 karakter, karakter dapat berupa angka (10 buah), huruf (26 huruf besar dan 26 huruf kecil), maka jumlah kunci yang harus dicoba adalah

$$26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26 = 26^8 \text{ buah}$$

Secara teori, serangan secara *exhaustive* ini dipastikan berhasil mengungkap plainteks tetapi dalam waktu yang sangat lama (lihat Tabel 2.1).

Tabel 2.1 Waktu yang diperlukan untuk *exhaustive key search*
(Sumber: William Stallings, *Data and Computer Communication Fourth Edition*)

Ukuran kunci	Jumlah kemungkinan kunci	Lama waktu untuk 10^6 percobaan per detik	Lama waktu untuk 10^{12} percobaan per detik
16 bit	$2^{16} = 65536$	32.7 milidetik	0.0327 mikrodetik
32 bit	$2^{32} = 4.3 \times 10^9$	35.8 menit	2.15 milidetik
56 bit	$2^{56} = 7.2 \times 10^{16}$	1142 tahun	10.01 jam
128 bit	$2^{128} = 4.3 \times 10^{38}$	5.4×10^{24} tahun	5.4×10^{18} tahun

Untuk menghadapi serangan ini, perancang kriptosistem (kriptografer) harus membuat kunci yang panjang dan tidak mudah ditebak.