BAB II

TINJAUAN PUSTAKA

2.1 Kriptografi

2.1.1 Pengertian Kriptografi

Ada beberapa definisi yang telah dikemukakan di dalam berbagai literatur. Kata kriptografi berasal dari bahasa Yunani yaitu *cryptos* yang berarti rahasia dan *graphein* yang berarti tulisan [5]. Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia dengan menggunakan teknik-teknik metematika.

Dalam menjaga kerahasiaan data dengan kriptografi, data sederhana yang dikirim (plainteks) diubah ke dalam bentuk data sandi (cipherteks), kemudian data sandi tersebut hanya dapat dikembalikan ke bentuk data sebenarnya hanya dengan menggunakan kunci (*key*) tertentu yang dimiliki oleh pihak yang sah saja. Tentunya hal ini menyebabkan pihak lain yang tidak memiliki kunci tersebut tidak akan dapat membaca data yang sebenarnya sehingga dengan kata lain data akan tetap terjaga

2.1.2 Tujuan Kriptografi

Kriptografi bertujuan untuk memberikan layanan pada aspek-aspek keamanan antara lain [5] :

1. kerahasiaan (*confidentiality*), yaitu menjaga supaya pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak,
2. integritas data (*data integrity*), yaitu memberikan jaminan bahwa untuk tiap bagian pesan tidak akan mengalami perubahan dari saat data

dibuat/dikirim oleh pengirim sampai dengan saat data tersebut dibuka oleh penerima data,

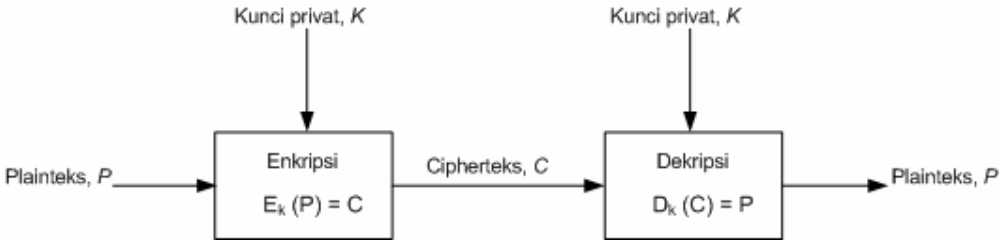
3. otentikasi (*authentication*), yaitu berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan,
4. nirpenyangkalan (*non repudiation*), yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang tertentu sehingga apabila ada seseorang yang mencoba mengakui memiliki dokumen tersebut, dapat dibuktikan kebenarannya dari pengakuan orang tersebut.

2.1.3 Jenis Algoritma Kriptografi

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi 2 macam, yaitu kriptografi simetri (*symetric cryptography*) dan kriptografi asimetri (*asymetric cryptography*) [9].

2.1.3.1 Kriptografi Simetri (*Symetric Cryptography*)

Pada sistem kriptografi simetri, kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kunci. Istilah lain untuk kriptografi simetri adalah kriptografi kunci privat (*private key cryptography*) atau kriptografi konvensional (*conventional cryptography*).



Gambar 2.1 Kriptografi Simetri (*Symetric Cryptography*)

Algoritma kriptografi simetri dapat dikelompokkan menjadi dua kategori antara lain :

1. *Cipher* aliran (*stream cipher*)

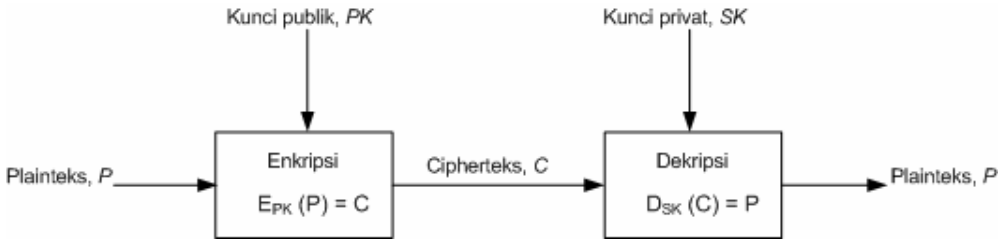
   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal yang dalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit. Cipher aliran mengenkripsi satu bit setiap kali.

2. *Cipher* blok (*block cipher*)

   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya. *Cipher* blok mengenkripsi satu blok bit setiap kali.

2.1.3.2 Kriptografi Asimetri (*Asymetric Cryptography*)

Pada sistem kriptografi asimetri, kunci untuk proses enkripsi tidak sama dengan kunci untuk proses dekripsi. Istilah lain untuk kriptografi asimetri adalah kriptografi kunci publik (*public key cryptography*), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan.



Gambar 2.2 Kriptografi Asimetri (*Asymetric Cryptography*)

2.1.4 Tinjauan Matematis

2.1.4.1 Konsep Dasar Kriptografi

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi-relasi himpunan yaitu himpunan yang berisi elemen plainteks dan himpunan yang berisi elemen cipherteks [4]. Enkripsi dan dekripsi merupakan fungsi transformasi antara dua himpunan tersebut. Bila himpunan plainteks dinotasikan dengan *P* dan himpunan cipherteks dinotasikan dengan *C*, sedangkan fungsi enkripsi dinotasikan dengan *E* dan fungsi dekripsi dinotasikan dengan *D* maka proses enkripsi dan dekripsi dapat dinyatakan dalam notasi matematis sebagai berikut :

$$E(P) = C \tag{2.1}$$

$$D(C) = P \tag{2.2}$$

Karena proses enkripsi dan dekripsi bertujuan untuk memperoleh kembali data asal, maka :

$$D(E(P)) = P \tag{2.3}$$

Relasi antara himpunan plainteks dengan himpunan cipherteks merupakan korespondensi satu-satu. Hal ini merupakan keharusan untuk mencegah terjadinya

ambigu dalam dekripsi yaitu satu elemen cipherteks menyatakan lebih dari satu elemen plainteks.

Pada metode kriptografi simetris hanya menggunakan satu buah kunci untuk proses enkripsi dan dekripsi. Bila kunci dinotasikan dengan *k* maka proses enkripsi dan dekripsi kriptografi simetris dapat dinyatakan sebagai berikut :

$$E_k(P) = C$$

$$D_k(C) = P$$

maka

$$D_k(E_k(P)) = P \tag{2.4}$$

Pada metode kriptografi asimetris menggunakan kunci umum untuk enkripsi dan kunci pribadi untuk dekripsi. Bila kunci umum dinotasikan dengan *pk* dan kunci pribadi dinotasikan dengan *sk*, maka proses enkripsi dan dekripsi kriptografi asimetris dapat dinyatakan sebagai berikut :

$$E_{pk}(P) = C$$

$$D_{sk}(C) = P$$

maka

$$D_{sk}(E_{pk}(P)) = P \tag{2.5}$$

2.1.4.2 Aritmatika Modular

Aritmatika modular merupakan operasi matematika yang banyak diimplementasikan pada metoda kriptografi [2]. Pada metoda kriptografi simetris, operasi aritmatika modular yang sering dipakai adalah operasi penjumlahan

modulo dua dalam operasi XOR (Exclusive OR) dengan dinotasikan $\oplus$. Nilai operasi XOR dapat dilihat pada tabel berikut ini :

Tabel 2.1 Tabel nilai operasi XOR

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Dari tabel 2.1 dapat dilihat sifat-sifat untuk operasi XOR :

$A \oplus A = 0$

$A \oplus 0 = A$

$A \oplus 1 = A'$ dengan A' adalah kebalikan (komplemen) nilai A

2.1.5 *Cipher* Blok

Pada *cipher* blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama. Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan blok plainteks). Algoritma enkripsi menghasilkan blok cipherteks yang berukuran sama dengan blok plainteks. Dekripsi dilakukan dengan cara yang serupa seperti enkripsi.

Misalkan blok plainteks (P) yang berukuran n bit dinyatakan sebagai vektor

$$P = (p_1, p_2, \ldots, p_n) \tag{2.6}$$

yang dalam hal ini p_i adalah bit 0 atau bit 1 untuk $i = 1,2,\ldots,n$ dan blok cipherteks (C) adalah

$$C = (c_1, c_2, \ldots, c_n) \tag{2.7}$$

yang dalam hal ini c_i adalah bit 0 atau bit 1 untuk $i = 1,2,\ldots,n$. Bila plainteks dibagi menjadi m buah blok, barisan blok-blok plainteks dinyatakan sebagai

$$(P_1, P_2, \ldots P_m)$$

Untuk setiap blok plainteks P_i, bit-bit penyusunnya dapat dinyatakan sebagai vektor

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \tag{2.8}$$

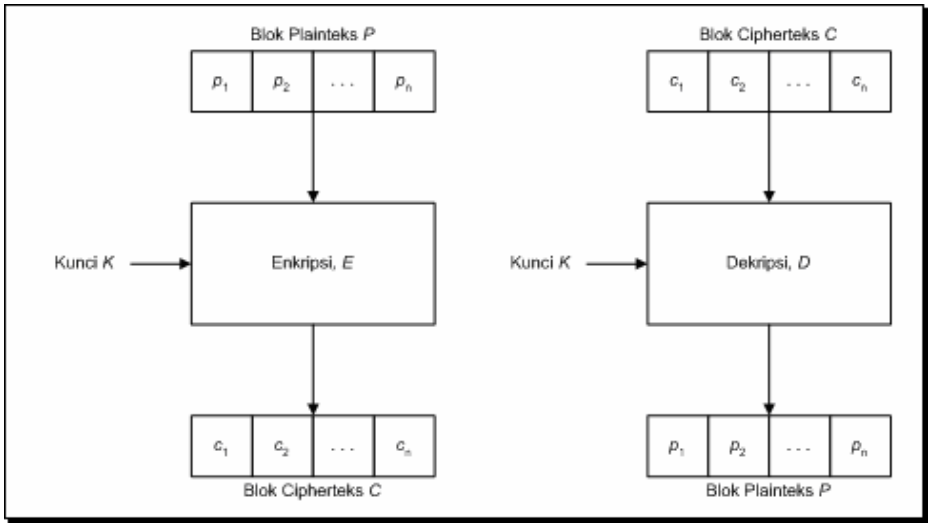
Enkripsi dengan kunci K dinyatakan dengan persamaan

$$E_k(P) = C \tag{2.9}$$

sedangkan dekripsi dengan kunci K dinyatakan dengan persamaan

$$D_k(C) = P \tag{2.10}$$

Skema enkripsi dan dekripsi dengan cipher blok dapat dilihat pada gambar 2-3.



Gambar 2.3 Proses Enkripsi dan Dekripsi pada *Cipher* Blok

2.1.6 *Avalanche Effect*

Salah satu karakteristik untuk menentukan baik atau tidaknya suatu algoritma kriptografi adalah dengan melihat *avalanche effect*-nya [4]. Perubahan yang kecil pada plainteks maupun kunci akan menyebabkan perubahan yang signifikan terhadap cipherteks yang dihasilkan. Atau dengan kata lain, perubahan satu bit pada plainteks maupun kunci akan menghasilkan perubahan banyak bit pada cipherteks. Suatu *avalanche effect* dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45-60% (sekitar separuhnya, 50 % adalah hasil yang sangat baik) [7]. Hal ini dikarenakan perubahan tersebut berarti membuat perbedaan yang cukup sulit untuk kriptanalis melakukan serangan. Nilai *avalanche effect* dirumuskan dengan :

$$Avalanche_Effect(AE) = \frac{\sum bit_berubah}{\sum bit_total} * 100\% \quad (2.11)$$

2.1.7 Proses *Padding*

Proses *padding* adalah proses penambahan byte-byte *dummy* berupa karakter *NULL* pada byte-byte sisa yang masih kosong pada blok terakhir plaintext, sehingga ukurannya menjadi sama dengan ukuran blok penyandian [4]. Sebagian besar ukuran arsip yang akan disandikan tidak merupakan kelipatan dari ukuran blok penyandian, sehingga blok terakhir dari arsip memiliki ukuran yang lebih kecil dari ukuran blok penyandian. Hal ini mengakibatkan tidak sesuainya perhitungan matematis pada blok terakhir pada penerapan beberapa algoritma kriptografi yang perhitungan matematisnya mengharuskan ukuran blok penyandian sesuai dengan yang telah ditentukan.

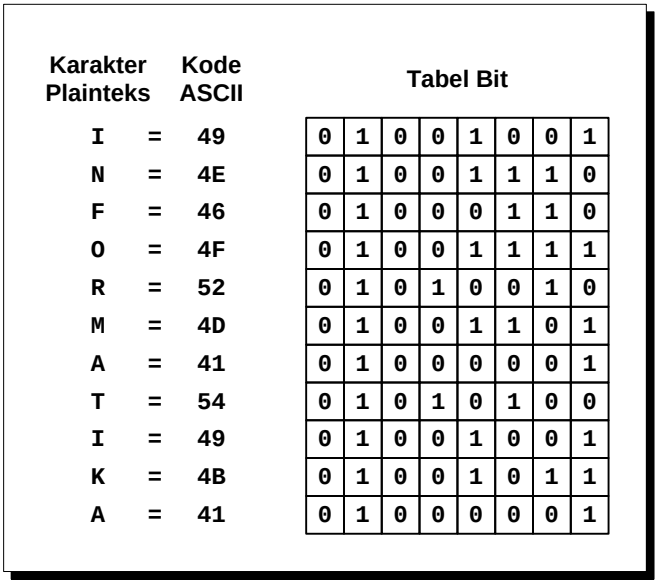
2.2 Algoritma Kriptografi *Vertical Bit Rotation* (VBR)

Algoritma kriptografi *Vertical Bit Rotation* (VBR), dibuat oleh Hanson Prihantoro Putro dalam makalahnya berjudul “Teknik Kriptografi Block Chiper dengan VBR (Perputaran Bit Vertical)” [3]. Algoritma kriptografi ini masuk dalam jenis kriptografi simetris dan masuk dalam kategori *cipher* blok. Pembuatan algoritma kriptografi ini lebih bertujuan pada membuat algoritma kriptografi modern namun memiliki kesederhanaan seperti algoritma kriptografi klasik.

2.2.1 Pembentukan blok

Algoritma kriptografi VBR menggunakan blok dengan ukuran maksimal 256 byte. Namun untuk mendapatkan satu blok, tidak perlu dilakukan *padding* karena tanpa *padding* pun algoritma kriptografi ini masih dapat berjalan.

Pertama-tama, sebuah arsip yang akan disandikan dibaca bit-bitnya, lalu dibagi sesuai dengan ukuran blok penyandian yaitu sebesar *N* byte. Dari satu blok, dipecah lagi per satu buah karakter, yaitu 8 bit, lalu pecahan-pecahan tersebut diurutkan secara vertikal. Sehingga didapatkan sebuah tabel bit yang terdiri dari 8 kolom dan *N* baris untuk *N* adalah jumlah (ukuran) blok penyandian. Apabila blok penyandian berukuran 256 byte, maka akan didapatkan tabel bit berukuran 8 kolom dan 256 baris. Sebagai contoh dapat dilihat pada gambar 2.4.



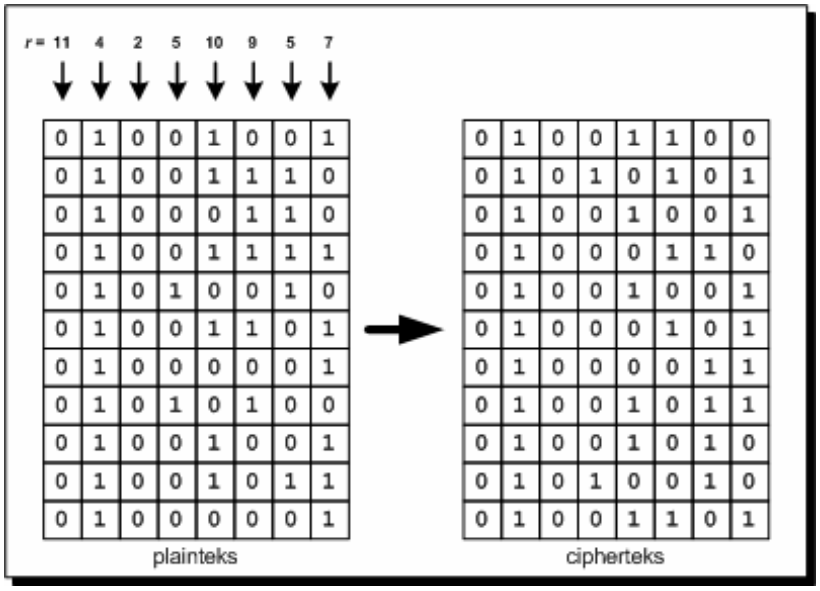
Gambar 2.4 Pembentukan blok penyandian pada algoritma kriptografi VBR

2.2.2 Proses Enkripsi

Proses enkripsi dilakukan pada setiap blok penyandian, blok per blok, hingga pada blok terakhir. Apabila ukuran blok terakhir lebih kecil dari blok penyandian yang telah ditentukan, tidak dilakukan *padding* sebagaimana dilakukan pada beberapa algoritma kriptografi lainnya.

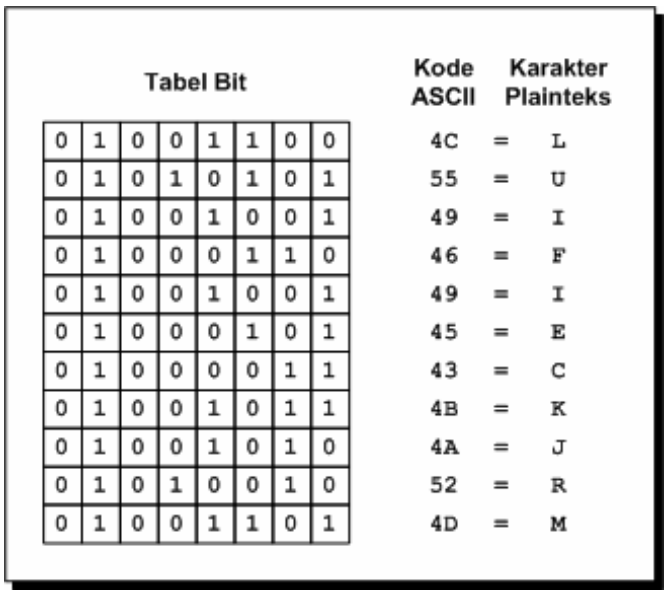
Enkripsi dilakukan dengan cara menggeser ke bawah dari masing-masing kolom pada tabel bit yang telah didapat pada saat pembentukan blok penyandian. Besarnya pergeseran telah ditetapkan terlebih dahulu, sebesar *r* bit, pada sebuah kolom, sehingga pada setiap blok akan mengalami besar pergeseran yang sama untuk kolom yang sama. Namun untuk masing-masing kolom dapat mengalami pergeseran dengan besar yang berbeda. Karena pada tabel bit terdapat 8 kolom bit, maka diperlukan pula 8 variabel nilai untuk menggeser bit-bit pada setiap kolom (r_1, r_2, r_3, … r_8). Contoh untuk enkripsi kata INFORMATIKA menggunakan

nilai-nilai penggeser r_1 hingga r_8 masing-masing adalah 11, 4, 2, 5, 10, 9, 5, 7, dapat dilihat pada gambar 2.5.



Gambar 2.5 Proses enkripsi pada algoritma kriptografi VBR

Proses enkripsi akan menghasilkan cipherteks yang dapat dilihat sebagai karakter seperti pada gambar 2.6.



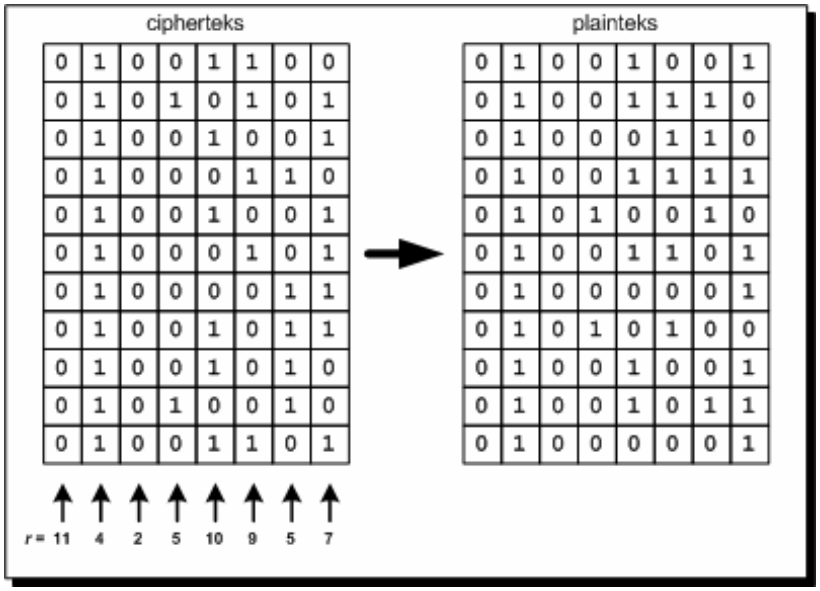
Gambar 2.6 Hasil enkripsi algoritma kriptografi VBR

Variabel rotasi yang berjumlah 8 buah dapat diambil dari kunci yang terdiri dari 8 karakter (8 byte = 64 bit) yang diambil nilai ASCIInya.

2.2.3 Proses Dekripsi

Sama halnya dengan proses enkripsi, proses dekripsi dilakukan pada setiap blok penyandian, blok per blok, hingga pada blok terakhir, dan tidak dilakukan *padding* pada blok terakhir.

Pada proses dekripsi, juga dilakukan pergeseran pada tiap-tiap kolom pada tabel bit. Hanya saja bila pada proses enkripsi dilakukan pergeseran ke bawah, maka pada proses dekripsi dilakukan sebaliknya, yaitu digeser ke atas. Nilai dari pergeseran setiap kolom pada proses dekripsi harus sama dengan nilai pergeseran setiap kolom pada proses enkripsi. Karena bila tidak sama, proses dekripsi menjadi tidak sempurna dan tidak akan didapatkan arsip yang benar sebagai hasil dari proses dekripsi. Sesuai dengan contoh pada proses enkripsi, contoh untuk proses dekripsi dapat dilihat pada gambar 2.7.



Gambar 2.7 Proses dekripsi pada algoritma kriptografi VBR