SENI KRIPTOGRAFI DAN ALGORITMA ENKRIPSI DATA SEDERHANA DENGAN MENGGUNAKAN VISUAL BASIC 6.0

Oleh: **Nandang Iriadi**

ABSTRACT

Cryptography is essential in information system. It provides information with a guarantee in its accuracy and secrecy. Cryptography can prevent deception in e-commerce and protect the legality of a financial transaction. It keeps identity originality and keeps an identity a secret. It also prevents acts of damaging documents and protects them from competitors or irresponsible parties. On the contrary, many of cryptography products on the markets do not emphasize on safety level. Many of those are not designed by cryptographers but by technical engineers who do not view cryptography as the essential part. A secure system cannot be designed only by "giving/attaching" cryptography inside but by knowing the procedures inside the system starting from its concept until it is attached on.

Millions of Rupiahs has been spent for computer security. Yet it seems ineffectual since computers are still insecure. A powerful and weak cryptography product looks alike. Two e-mail encryption products look the same. One is secure while the other is fragile. After being tested, both programs have specification similarities. Only there is one with security holes. A well-experienced cryptographer and criminals can only see this discrepancy.

Kriptografi merupakan bagian yang esensial dari sistem informasi. Kriptografi membantu menyediakan informasi tetapi dapat dipertanggungjawabkan keakuratan dan kerahasiaannya. Kriptografi juga dapat mencegah penipuan dalam e-commerce serta melindungi keabsahan transaksi keuangan. Selain itu kriptografi dapat menjaga keaslian identitas ataupun merahasiakan identitas. Kriptografi juga dapat mencegah aksi perusakan dokumen dan melindungi dari kompetitor atau orang-orang yang tidak berhak, tetapi saat ini di pasaran banyak produk kriptografi tidak menitikberatkan pada level keamanan. Kebanyakan kriptografi tersebut tidak dirancang dan dibuat oleh kriptografer (sandiman) tetapi oleh Insinyur teknik, yang mana mereka menganggap kriptografi hanya merupakan bagian kecil saja. Kita tidak dapat membuat suatu sistem aman dengan hanya "memberikan/menempelkan" kriptografi di dalamnya. Kita harus mengetahui setiap langkah dalam sistem tersebut mulai dari konsep sampai terpasang.

Jutaan rupiah telah dibelanjakan untuk keamanan komputer, tetapi semuanya seperti sia-sia karena komputer tetap dalam kondisi tidak aman. Hal ini karena produk kriptografi yang kuat sepintas terlihat sama dari yang lemah. Dua produk e-mail encryption terlihat sama. Tetapi yang satu aman dan yang lainnya mudah dibongkar. Setelah diuji kedua program mempunyai kesamaan spesifikasi, tetapi yang satu terdapat security holes dan lainnya tidak. Perbedaan itu hanya bisa dilihat oleh sandiman berpengalaman dan juga oleh para kriminal.

I. PENDAHULUAN

Dalam dunia perdagangan bentuk-bentuk kejahatan seperti penipuan, pemalsuan, penolakan dan kecurangan selalu dihadapi. Kenyataannya komputerisasi membuat risiko ancaman kejahatan menjadi semakin besar. Kita tidak dapat berjalan di jalanan dengan memakai topeng wajah (identitas) orang

lain. Tetapi di dunia digital hal tersebut sangat mudah. Hanya kriptografi yang kuat yang dapat melindungi serangan ini.

Serangan terhadap milik pribadi adalah ancaman berikutnya. Target serangan milik pribadi biasanya adalah mencuri informasi atau *e-mail* mengenai *public figure*, atau perusahaan kompetitor dan pencurian *account* pribadi atau perusahaan. Seseorang menyerang komputer dengan berbagai motivasi, misalnya untuk mencari popularitas, keperluan riset, mencuri uang, mencari data atau informasi rahasia, pembuktian kasus, kompetisi dalam perusahaan, atau balas dendam. Dalam era komputerisasi, kriminal dalam jaringan komputer (*e-crime*) merupakan masalah yang serius. Para penyerang tidak pernah mengikuti prosedur yang lazim, mereka mencuri dan merusak dengan teknik yang belum pernah terpikirkan sebelumnya. Mereka menggunakan teknologi yang selangkah lebih maju dari sistem yang akan diserangnya. Biasanya kita melindunginya dengan menutup *vulnerabilities* yang mungkin ditemukan, tetapi penyerang hanya membutuhkan satu saja untuk memasuki sistem dan merusaknya.

II. PEMBAHASAN

Saat di Paris, pakar simbologi Harvard, Robert Langdon, menerima telepon tengah malam yang penting. Seorang kurator senior di Museum Louvre, Jacques Sauniere, terbunuh dengan posisi tubuh terlentang, simbol pentakel (bintang lima dikelilingi lingkaran yang dikenal sebagai simbol pemuja setan) terukir di dada sang kurator dengan darahnya sendiri, dan pesan-pesan rahasia yang mengherankan ditemukan dekat tubuhnya :

"*13-3-2-21-1-1-8-5. O, Draconian devil. Oh, lame saint*", Langdom dan seorang kriptolog berbakat prancis, Sophie Neveu, mengupas lapis demi lapis teka-teki aneh itu. Ternyata, deret angka yang ditemukan itu adalah "Rangkaian Fibonancci" (Deret ukur matematika berkualitas dan terkenal sejak abad ketiga belas, diciptakan oleh ahli matematika Leonardo Fibonancci), yang diacak. Sedangkan kalimat "*O, Draconian devil. Oh, lame saint*" (O, setan draconia. Oh, orang yang suci) adalah sebuah anagram (Anagram adalah seni mengatur kembali huruf - huruf dari sebuah kata atau kalimat untuk membuat arti baru, raja-raja Prancis di zaman *Renaissance* percaya bahwa seni anagram mengandung kekuatan magis). Pesan tersebut adalah anagram yang sempurna dari "*Leonardo da Vinci*" dan "*The Monalisa*" yang merupakan kata kunci untuk teka-teki berikutnya dan akan membawa mereka ke sebuah rahasia besar yang telah disimpan selama berabad-abad dan memakan ribuan nyawa manusia.

Cerita diatas dikutip dari sebuah Novel yang baru-baru ini telah menggemparkan umat di seluruh dunia, "*The DaVinci Code*" karangan Dan Brown. Novel yang penuh muatan sejarah dan teka-teki yang "memukau nalar" pembacanya, sebuah seni kriptografi yang diangkat melalui sebuah novel fiksi. Dari cerita diatas, dapat disimpulkan bahwa seni kriptografi sudah ada sejak berabad-abad lamanya. Seiring dengan berkembangnya zaman, seni kriptografi juga berkembang menjadi seni modern yang diadopsi oleh sistem kemanan komputer untuk menjaga kerahasiaan sebuah informasi. Algoritma-algoritma kompleks telah diciptakan oleh para *programmer* untuk menjaga kerahasiaan informasi-informasi penting diseluruh dunia.

a. Kriptografi (*Cryptography*)

Kriptografi merupakan seni untuk menjaga pesan agar aman dari orang - orang yang tidak berhak menerima pesan tersebut. "*Crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan), jadi *criptography* bisa diartikan

pesan rahasia. Pesan tersebut diacak sedemikian rupa dengan metoda atau algoritma tertentu agar tidak mudah dibaca. Sebuah algoritma kriptografik *(cryptographic algorithm)* merupakan persamaan matematik yang digunakan untuk proses enkripsi dan deskripsi, kedua persamaan matematis untuk enkripsi dan deskripsi tersebut biasanya memiliki hubungan matematis yang cukup erat. Para pelaku atau praktisi kriptografi disebut *cryptographers* atau kriptolog. Pengamanan data dengan menggunakan kriptografi dilakukan dengan dua cara, yaitu Transposisi *(Transpotition)* dan Substitusi *(Substitution)*. Pada penggunaan transposisi, posisi dari huruf yang diubah-ubah.

Sementara pada substitusi, huruf (atau kata) digantikan dengan huruf atau simbol lain. Jadi, pesan tersembunyi yang dibuat Jacques Sauniere menjelang detik-detik kematiannya pada novel *"The DaVinci Code"*, merupakan seni kriptografi dengan metoda transposisi ditambah dengan seni anagram. Tujuan kriptografi adalah sebagai berikut:

1. *Confidentiality*
   Untuk melindungi identitas pemakai atau isi pesan agar tidak dapat dibaca oleh orang lain yang tidak berhak.
2. Data *Integrity*
   Untuk melindungi pesan agar tidak diubah oleh orang lain.
3. *Authentication*
   Untuk menjamin keaslian pesan.
4. *Non repudiation*
   Membuktikan suatu pesan berasal dari seseorang, apabila ia menyangkal mengirim pesan tersebut.

Kumpulan *class* kriptografi terdapat dalam *namespace System.Security.Cryptography*, dan dapat dibagi menjadi empat divisi yaitu:

1. *Encryption Algorithms* (Algoritma Enkripsi)
   Sekumpulan *class* yang dapat digunakan untuk mengimplementasikan algoritma simetris, asimetris dan *hash*.
2. *Helper Classes* (*Class-class* penolong)
   Class-class yang digunakan untuk menghasilkan angka acak, melakukan konversi, interaksi dengan penyimpanan *Crypto* API, dan melakukan enkripsi menggunakan model berbasis *stream* atau aliran data.
3. *X.509 Certificates* (Sertifikasi X.509)
   Class-class yang terdapat pada *namespace System.Security.Cryptography.X509*, dapat digunakan untuk memberikan sertifikasi digital.
4. *XML Digital Signatures* (Tanda tangan digital XML)
   Class-class yang terdapat pada *namespace System.Cryptography.Xml*, dapat digunakan untuk memberikan tanda tangan digital dalam dokumen XML.

Tidak ada yang dapat menjamin keamanan 100 %. Kriptografi yang kuat hanya dapat mereduksi risiko keamanan menjadi seminimal mungkin. Dan program kriptografi bagaimanapun bagusnya, tidak akan dapat melindungi serangan dari dalam. Tetapi kriptografi tetap dapat melindungi data dari pencurian dan manipulasi serta melindungi dalam komunikasi walaupun seseorang menyadapnya.

Saat ini kriptografi telah mengembangkan algoritma dan protokol yang diperlukan dalam perlindungan sistem komputer. Tetapi dalam pelaksanaannya membutuhkan kemampuan sumber daya manusia yang tangguh. Karena keamanan meliputi interaksi antara manusia dan sistem komputer yang membutuhkan

manajemen dan disiplin dengan sedikit sekali ruang bagi kelalaian.

Pada kenyataannya kriptografi jarang dikupas dengan analisis matematika, ada bagian lain dari sistem yang lebih mudah disadap. Sering sebuah Institusi mengimplementasikan algoritma dan protokol yang memiliki kelemahan. Kelemahan bisa berada dimana saja, pada *hardware, software*, model, disain, manajemen ataupun sandimannya. Karena keamanan adalah suatu mata rantai proses, kelemahan satu bagian saja dapat dapat merusak seluruh sistem.

Keamanan adalah sesuatu proses yang sangat berbeda dari kegiatan lainnya. Berfungsinya sebuah produk keamanan, tidak menjamin kualitasnya. Hanya karena kombinasi kuncinya benar, tidak menjamin alat tersebut aman dari para *crypto cracker*. Sekali sebuah kelemahan ditemukan, kita dapat memperbaikinya. Tetapi mencari seluruh kelemahan dalam suatu produk *security* sangat sulit.

b. Sandiman sebagai pembangun kriptografi.

Kriptografi bisa juga dimasukan sebagai kesenian karena seorang sandiman dalam kegiatannya harus dapat menyeimbangkan antara aspek *security*, kemudahan, kecepatan, ketepatan dan dapat dipertanggungjawabkan. Hanya menguasai ilmu dasarnya saja tidaklah cukup. Pengalaman dan intuisi yang dibangun dari pengalaman itulah yang akan membantu seorang sandiman dalam melakukan tindakan pengamanan atau kriptografi menjadi efektif.

Banyak sekali sistem keamanan dapat di akses orang yang tidak bertanggung jawab karena para penggunanya sendiri yang tidak teliti. Para pengguna data rahasia atau bahkan Sandimannya seringkali mengabaikan keamanan data tersebut, mereka lebih mengutamakan kemudahan, dan prinsip "yang penting sudah dikerjakan". Memang bagian paling sukar dalam kriptografi adalah memberikan pengertian pada pengguna untuk selalu menjaga keamanan data tersebut. Memberikan pengertian mengenai pentingnya keamanan seperti menjaga harta mereka dari para pencuri.

Saat ini lebih banyak *vendor* atau *agent* dalam membandingkan sebuah produk *security* dengan mendaftar spesifikasi atau fungsi, bukan dalam tingkat keamanannya. Sandiman sulit belajar dari kegagalan suatu produk *security*. Karena tidak seperti produk pesawat misalnya yang bila jatuh atau *crash* akan dapat dianalisa secara detail dan dipublikasikan secara umum. *Security* sangat berbeda, sebuah institusi akan menutupi kelemahan sistemnya sehingga sulit mendapatkan analisa detail dari kelemahan tersebut.

Teknologi Informasi berkembang sangat cepat, dan pengamanan data informasi selangkah dibelakangnya. Sandiman harus memacu diri untuk selalu dapat mendampingi dalam pengamanan informasi berbasis teknologi komputer tersebut dengan memikirkan perkembangan teknologi yang semakin cepat. Bila terlambat akan memberikan jarak yang akan semakin sulit dikejar. Jangan menganggap terlalu tinggi kekuatan suatu sistem keamanan. Jauh lebih baik selalu mengasumsikan hal terburuk dalam sistem, sehingga selalu waspada. Berikan ruang untuk kesalahan (*error*), berikan keamanan lebih dari yang dibutuhkan, bila sesuatu yang tidak diinginkan muncul, sandiman telah siap.

c. Enkripsi (*Encryption*) dan Deskripsi (*Descryption*).

Enkripsi *(encryption)* yang merupakan bagian dari seni kriptografi, adalah proses yang dilakukan untuk mengamankan sebuah pesan *(plaintext)* menjadi pesan yang tersembunyi

(ciphertext). *Plaintext* adalah pesan yang belum dienkripsi, sedangkan *ciphertext* adalah pesan setelah dienkripsi yang sudah tidak dapat dibaca dengan mudah. Sedangkan proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext* atau proses untuk menterjemahkan pesan yang sudah dienkripsi kedalam bentuk pesan aslinya disebut dekripsi *(decryption)*. Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi, data disandikan *(encrypted)* dengan menggunakan sebuah algoritma. Untuk membuka pesan rahasia tersebut *(decrypt)* digunakan juga sebuah algoritma yang berkaitan erat dengan algoritma untuk enkripsi.

Kunci biasanya dipergunakan untuk menutup pintu serta bermanfaat untuk keamanan seorang yang mempunyai pintu. Demikian juga dengan data yang terenkripsi mempunyai dua kunci yaitu:

1. Kunci simetris
   Algoritma simetris menggunakan kunci yang sama untuk mengenkrip dan mendeskrip.
2. Kunci asimetris
   Algoritma asimetris menggunakan dua kunci, kunci publik untuk enkripsi dan kunci pribadi untuk melakukan deskripsi.

Contoh cara kerjanya, jika Rangga ingin mengirim pesan kepada Cinta, maka Rangga akan mengenkripsi pesannya menggunakan kunci publik dari Cinta. Ketika Cinta menerima pesan dari Rangga, maka Cinta akan menggunakan kunci pribadinya untuk mendeskripsi pesan dari Rangga. (jadi, Ada Apa Dengan Cinta?)

d. *Substitution Cipher, Caesar Cipher*

Substitution cipher adalah metoda enkripsi dengan cara menggantikan huruf yang digunakan untuk mengirim pesan dengan huruf lain. Salah satu dari *substitution cipher* yang cukup terkenal adalah *Caesar Cipher* yang digunakan oleh Julius Caesar untuk mengirimkan pesan rahasia pada jamannya. Pada prinsipnya, *caesar chiper* mempunyai cara kerja mengganti setiap huruf didalam pesan dengan huruf yang berada tiga (3) posisi dalam urutan alphabet.

Sebagai contoh huruf "A" digantikan dengan huruf "D", huruf "B" digantikan dengan huruf "E" dan seterusnya. Transformasi yang digunakan untuk menggeser huruf tersebut adalah :

Tabel II.1 Transformasi Caesar Cipher (geser 3).

Plaintext	:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Ciphertext	:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

Jadi, kata "BSI" setelah enkripsi akan menjadi "EVL" (Lihat Tabel II.1). Penggunaaan *Caesar Cipher* ini dapat dimodifikasi jumlah geseran (bukan hanya 3) dan juga arah geserannya (kekanan atau kekiri). Jadi kita dapat menggunakan *Caesar Cipher* ini dengan menggeser 13 ke kiri misalnya. Hal ini dapat menyulitkan orang yang ingin menyadap pesan, sebab dia harus mencoba semua kombinasi (52 kemungkinan).

Memodifikasi satu huruf didalam pesan menjadi dua huruf dengan jumlah geseran yang berbeda setelah dienkripsi. Misalnya geser tiga (3) dan geser lima (5) dengan transformasi sebagai berikut :

Tabel.II.2. *Substitution cipher* dengan kombinasi geser 3 dan 5

Plaintext	:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Geser 3	:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Geser 5	:	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Jadi, kata "AMIK" setelah dikonversi akan menjadi : "DFPRLNNP"

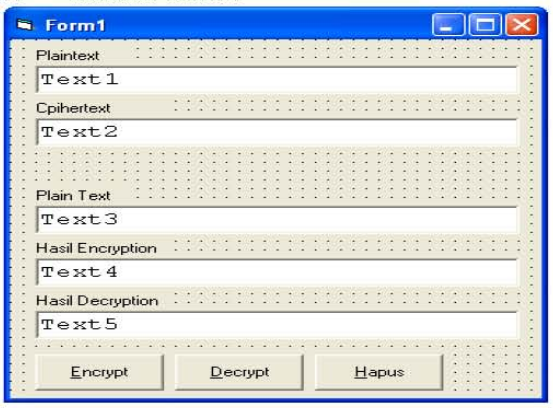
"DF" hasil konversi dari "A" (A+3=D dan A+5=F)
"PR" hasil konversi dari "M" (M+3=P dan M+5=R)
"LN" hasil konversi dari "I" (I+3=L dan I+5=N)
"NP" hasil konversi dari "K" (K+3=N dan K+5=P

Setelah membahas enkripsi data secara teoritis dan sedikit sejarah dan ilustrasinya, penulis akan memberikan trik-trik enkripsi data sederhana secara praktis menggunakan bahasa pemrograman Visual Basic 6.0. Penulis akan memberikan beberapa contoh program sederhana untuk menerapkan cara-cara enkripsi data dengan metoda *substitution cipher* yang digunakan oleh *Julius Caesar* dan dengan sedikit modifikasi untuk menghasilkan pesan rahasia yang akan sulit dibaca oleh mata-mata musuh *Julius Caesar*.

Trik Pertama :

Kita mulai dari *substitution cipher* pertama yang menggunakan cara menggeser tiga (3) huruf yang terdapat pada pesan *(plaintext)* (Lihat Tabel .1). Kita mulai dari *substitution cipher* pertama yang menggunakan cara menggeser tiga (3) huruf yang terdapat pada pesan *(plaintext)* (Lihat Gambar II.1)

1. Desain *form* :



Gambar II.1 *Desain Form Substitution Cipher* dengan Visual Basic 6.0

2. Skenario Program :

Setiap karakter yang dapat kita ketikan melalui *keyboard* selalu memiliki kode ASCII berupa angka-angka (data numerik) yang terurut. Misalnya kode ASCII dari karakter "A" adalah 65, "B" adalah 66 dan seterusnya. Jadi untuk merubah atau menggeser karakter "A" menjadi "D" (geser 3) dapat menggunakan perintah :

CipherText = Chr(Asc("A")+3)

Fungsi Chr() digunakan untuk mengkonversi data angka menjadi karakter sesuai dengan tabel ASCII, sedangkan Fungsi Asc() digunakan sebaliknya, untuk mengkonversi data karakter menjadi data numerik sesuai dengan kode ASCII. Pada contoh ini, kedua fungsi tersebut selalu digunakan. Penjelasan program pada saat *form* aktif, *text* 1 akan berisi urutan huruf dari "A" sampai "Z", sedangkan Text 2 akan berisi urutan huruf dari Text 1 yang digeser 3 dimulai dari "D" sampai "Z", "A", "B", "C" (3 huruf pertama akan berada dibelakang). (Lihat Tabel II.1). Masukan sebuah kata pada Text 3 (Text 3 hanya bisa menerima masukan huruf "A" sampai "Z" atau huruf besar tanpa spasi dan *backspace*). Klik tombol *Encrypt* untuk mengkonversi (Enkripsi) kata pada Text 3, hasil enkripsi *(CipherText)* dari Text 3 akan keluar pada Text 4. Klik tombol *Decrypt* untuk mengkonversi (Dekripsi) kata pada Text 4, hasil dekripsi *(PlainText)* dari Text 4 akan keluar pada Text 5. Klik tombol Hapus untuk mengosongkan Text 3, Text 4 dan Text 5.

3. *Listing* program :

```
Private Sub cmdDecrypt_Click()
    Text5 = ""
'Mengetahui panjang text pada Text4
N = Len(Trim(Text4))
'Mengkonversi setiap karakter satu-
persatu pada text4 (geser -3)
For i = 1 To N
Cipher = Mid(Text4, i, 1)
Plain = Chr(Asc(Cipher) - 3)
```

```
'Jika Plaintext adalah huruf A, B atau
C, maka ubah menjadi X, Y atau Z
If Asc(Plain) < 65 Then
Text5.SelText = Chr((Asc(Plain)-64)+ 90)
Else
    Text5.SelText = Plain
End If
Next
End Sub

Private Sub cmdEncrypt_Click()
    Text4 = ""
'Mengetahui panjang text pada Text3
    N = Len(Trim(Text3))
'Mengkonversi setiap karakter satu-
persatu pada text3 (geser 3)
    For i = 1 To N
    Plain = Mid(Text3, i, 1)
    Cipher = Chr(Asc(Plain) + 3)
 'Jika Ciphertext adalah huruf X, Y
 atau Z
 'maka ubah menjadi A, B, atau C
 If Asc(Cipher) > 90 Then
Text4.SelText = Chr(64 + (Asc(Cipher) -
90))
 Else
   Text4.SelText = Cipher
 End If
 Next
 End Sub
 Private Sub cmdHapus_Click()
    Text3 = "": Text4 = "": Text5 = ""
    Text3.SetFocus
 End Sub

Private Sub Form_Activate()
    Text1 = "": Text2 = "": Text3 = ""
    Text4 = "": Text5 = ""
    For i = 65 To 90
    Text1.SelText = Chr(i)
    Next
 'Menampilkan karakter A - Z setelah
digeser 3 ke Text2
    For i = 65 + 3 To 90 + 3
    If i > 90 Then
   Text2.SelText = Chr(64 + (i - 90))
    Else
   Text2.SelText = Chr(i)
    End If
    Next
End Sub

Private Sub Text3_KeyPress(KeyAscii
As Integer)
    'Jika yang ditekan tombol
Backspace,
    'maka keluar dari sub procedure
     If KeyAscii = 8 Then Exit Sub
    'Mengubah huruf kecil menjadi
huruf besar
    KeyAscii =
Asc(UCase(Chr(KeyAscii)))
    'jika karakter yang dimasukan di
Text3
    'bukan karakter A - Z, maka ubah
karakter menjadi kosong
    If Not (KeyAscii >= 65 And
KeyAscii <= 90) Then
    KeyAscii = 0
    End If
End Sub
```

4. Jalankan program dan masukan sebuah kata pada Text3. Klik *Encrypt* untuk mengenkripsi kata tersebut. Klik *Descrypt* untuk mengembalikan pesan yang sudah di-enkripsi ke pesan semula. Klik Hapus untuk mengulangi langkah diatas.

Algoritma Enkripsi diatas masih bisa dikembangkan lebih jauh lagi, misalnya kita ingin menyimpan data penting seperti data *password* ke *database*. Dengan cara diatas, *password* yang akan disimpan di *database* bisa kita enkripsi terlebih dahulu, jadi yang tersimpan di *database* adalah *ciphertext*-nya.

Jika kita ingin mengambil *password* tersebut dari *database*, kita bisa menggunakan cara dekripsi untuk mengembalikan *ciphertext* ke *plaintext*. Enkripsi data dengan cara ini masih sangatlah sederhana. Karena antara *plaintext* dan *ciphertext* panjangnya sama, maka orang lain akan dengan mudah menganalisa isi dari *ciphertext* dengan mencoba-coba *(brute atack)* beberapa kombinasi jumlah pergeseran

posisi huruf dalam alphabet dan arah pergeserannya (kanan atau kiri). Orang lain hanya perlu mencari tahu berapa jumlah pergeseran huruf yang dilakukan. Jika sudah diketahui bahwa pergeseran yang dilakukan (3 huruf kekanan), maka cara untuk menerjemahkan pesan rahasia itu hanya menggeser setiap huruf kearah sebaliknya (kiri) sebanyak tiga (3) juga. Jika pesan rahasia yang akan disampaikan adalah strategi perang sang Julius Caesar, dan pesan tersebut berhasil diterjemahkan oleh mata-mata musuh, maka tamatlah riwayat kejayaan Sang Caesar.

Untuk menghasilkan *ciphertext* yang lebih sulit dibaca, maka diperlukan modifikasi dari enkripsi diatas. Misalnya merubah satu huruf pada *plaintext* menjadi beberapa huruf dengan geseran yang berbeda.

III. KESIMPULAN.

Dari pembahasan diatas, maka dapat di ambil beberapa kesimpulan sebagai berikut :

a. Dalam pengamanan informasi berbasis teknologi komputer harus memikirkan perkembangan yang jauh melampaui.
b. Bila terlambat akan memberikan jarak yang akan semakin sulit dikejar.
c. *Firewall* dan *password* saja tidak cukup lengkap untuk dapat membuat data aman dari intaian maka perlu melindungi data tersebut ditempat data tersebut berada. Hal tersebut membutuhkan sebuah dukungan teknologi enkripsi yang "kuat".
d. Dengan pengimplementasikan sistem penyandian data akan memberikan organisasi jaminan akan sistem keamaan yang menyeluruh.
e. Sebuah sistem enkripsi harus cukup cepat sehingga tidak akan memberikan dampak pada kinerja dari sistem itu sendiri. Seorang pengguna tidak boleh merasakan adanya perubahan hasil proses dengan atau tanpa sistem enskripsi.
f. Teknologi enkripsi harus menjamin integritas seluruh data, baik pada saat penyandian dan pada saat proses sebaliknya, tanpa sebuah kegagalan. Solusi ini harus didasarkan pada standarisasi secara global, agar mudah terukur dan juga harus cukup "kuat" untuk menjamin telah sesuai dengan seluruh perangkat hukum yang ada. Sistem enkripsi harus dibuat oleh badan atau organisasi yang kompeten, bukan berasal dari sebuah ide yang "coba-coba". Karena hal ini berkaitan dengan pengolahan data.
g. Solusi pengamanan data yang inovatif seharusnya tidak hanya melindungi data yang berada, ada PC (Personal Computer) dan dalam jaringan komputer organisasi saja, tetapi juga di setiap "titik" dimana data disimpan, Seperti di PDA, *SmartPhone, Email*, CD, *Jump Drive (USB flash disk)*, atau sistem penyimpanan lainnya. Dan sistem tersebut harus "menyatu" dan "kompatibel" dengan "kebijakan" sistem komputasi organisasi yang ada, bukan merubahnya.

DAFTAR PUSTAKA

Ariyus, Dony. 2005. *Computer Security*. Penerbit Andi. Yogyakarta.

Brenton, Chris. Hunt, Cameron. 2005. *Network Security*. Penerbit PT. Elex Media Komputindo. Jakarta.

Irawan, Budi. 2005. Jaringan Komputer. Penerbit Graha Ilmu. Yogyakarta.

Rahardjo, Budi. 2002. Keamanan Sistem Informasi Berbasis Internet. PT. Insan Indonesia & PT Indocisc. Jakarta.