PROGRAM ENKRIPSI DAN DEKRIPSI DENGAN MENGGUNAKAN METODE DES DAN METODE GOST

Oleh :

Muhammad Ghoffar[1], Manahan P. Siallagan[2], Andri Heryandi[3]
[1] Jurusan Teknik Informatika, FTIK UNIKOM, Jl. Dipati Ukur Bandung
[2] Dosen Teknik Informatika, FMIPA, Jl. Ganesha Bandung
[3] Dosen Teknik Informatika, FTIK UNIKOM, Jl. Dipati Ukur Bandung

Abstraksi

Perkembangan teknologi telekomunikasi dan penyimpanan data menggunakan komputer memungkinkan pengiriman data jarak jauh yang relatif cepat dan murah. Di lain pihak, pengiriman data jarak jauh melalui gelombang radio maupun medium lain yang digunakan masyarakat luas (public) sangat memungkinkan pihak lain dapat menyadap dan mengubah data yang dikirimkan. Demikian juga pada sistem jaringan komputer dengan jumlah pemakai yang banyak.

Dalam teknologi informasi, telah dan sedang dikembangkan cara-cara untuk menangkal berbagai bentuk serangan semacam ini. Salah satunya yang ditempuh untuk mengatasi masalah ini ialah dengan menggunakan kriptografi dalam bentuk enkripsi-dekripsi menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak ketiga.

Dalam tugas akhir ini akan disajikan analisis algoritma DES dan Gost dengan panjang blok cipher 64 bit .Tugas akhir ini juga akan menampilkan implementasi program untuk menampilkan bagaimana cara mengenkripsi dan mendekripsi file dengan menggunakan kedua algoritma tersebut.

Kata kunci : Enkripsi, Denkripsi, DES, Gost.

Abstract

The development of telecommunication and data storage technology using computer might able us to transferring data in long distances with relatively fast and cheap cost. On the other hand, transfer data in long distances trough the radio waves, wireless connection and other mediums that used in wide community, is very possible for the third side to adapt and change the data. This can also happen in computer network systems with many users.

In information technology, had and has been develop ways to protect every kind of this attack. One of them is with using cryptography in form of encryption and decryption. This way is using data transformation which produce data that cannot readable or understood by the third side.
In this final assignment, the writer will presents DES (Data Encryption standard) and Gost (Gosudarstvennyi Standard) algorithm analysis with 64 bit *cipher* block length. It's also will presents program implementation to view how to encrypt and decrypt a file using both algorithm.

Keywords : Encryption, Decryption, DES, Gost.

1. Pendahuluan

Perkembangan sistem komputerisasi manajemen dalam berbagai organisasi selama beberapa dasawarsa terakhir ini, menunjukkan bahwa pemanfaatan teknologi informasi dapat menjadi penunjang dalam pembuatan keputusan yang baik dan tepat bagi sebuah organisasi, baik organisasi pemerintah ataupun swasta.

Dengan berkembangnya teknik telekomunikasi dan sistem pengolahan data yang berkaitan erat dengan komunikasi antar pengguna komputer yang satu dengan komputer lain yang berfungsi untuk menyalurkan data sehingga masalah keamanan merupakan salah satu aspek penting dari suatu sistem informasi. Dalam komunikasi data terdapat sebuah metode pengamanan data yang dikenal dengan nama kriptografi. Kriptografi merupakan salah satu metode pengamanan data yang dapat

digunakan untuk menjaga kerahasiaan data, keaslian data, serta keaslian pengirim.

Kriptografi biasanya dalam bentuk enkripsi. Proses enkripsi merupakan proses untuk meng-*encode* atau mengubah data dari bentuk yang dapat dibaca menjadi bentuk yang tidak dapat dibaca. Sedangkan proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu mengubah data hasil enkripsi menjadi bentuk yang dapat dibaca. Dalam kriptografi terdapat berbagai macam sistem sandi yang tujuan penggunaan dan tingkat kerahasiaannya berbeda-beda.

Maksud dari penulisan ini yaitu sebagai salah satu syarat untuk menempuh ujian akhir sarjana fakultas teknik program strata I jurusan teknik informatika Universitas Komputer Indonesia

Sedangkan **tujuan** dari penulisan ini adalah :

- Memberikan sebuah solusi awal untuk keamanan data dengan metode kriptografi.
- Melanjutkan program dan memodifikasi sehingga menghasilkan sistem sandi yang benar-benar bagus.
- Menciptakan algoritma kriptografi yang lebih handal.

Sistem sandi yang akan digunakan berdasarkan metode DES dan metode Gost.

2. Metodologi

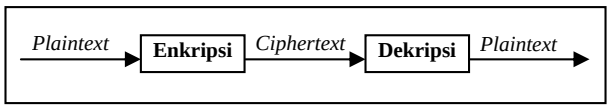
Metode yang dilakukan adalah metode deskriptif melalui observasi dan studi pustaka tentang metode DES dan metode Gost dari berbagai literature, data dan fakta yang berupa teori dan konsep, serta mencoba mengimplementasikannya dalam sistem sandi. Selain hal di atas, diadakan pengetesan terhadap program yang dibuat agar dapat melakukan proses penyandian dengan sempurna sehingga program yang dibuat tidak terdapat kesalahan.

Alat atau instrumen yang digunakan dalam penelitian menggunakan dua perangkat komputer untuk implementasinya dalam jaringan dengan spesifikasinya masing-masing :

- Komputer #1 : AMD Athlon XP 1600+, RAM 256 MB @ 133 MHz, monitor, LAN Card, dan sistem operasi Windows XP dengan SP1.
- Komputer #2 : AMD duron 1500+, RAM 128 MB, monitor, RAM 128 MB @ 266 MHz, monitor, LAN Card, dan sistem operasi Windows XP dengan SP1.

3. Landasan Teori

Bentuk asli dari sebuah pesan atau data disebut dengan *plaintext* dan bentuk asli dari pesan atau data yang terenkripsi di sebut *ciphertext.*



Gambar 3.1 Enkripsi dan Dekripsi

Data atau *Plaintext* dinotasikan dengan M *(Message)*, yang dapat berupa *bit tream, file text, digitized voice stream, digital video image*, dan sebagainya atau lebih singkatnya M adalah data ***binary.*** Fungsi Enkripsi **E**, berfungsi untuk mengubah **M** menjadi **C** atau *Ciphertext*, dalam matematika dinotasikan dengan:

E(M) = C.

Fungsi Dekripsi **D**, berfungsi untuk mengubah **C** menjadi **M**, dalam matematika dinotasikan dengan :

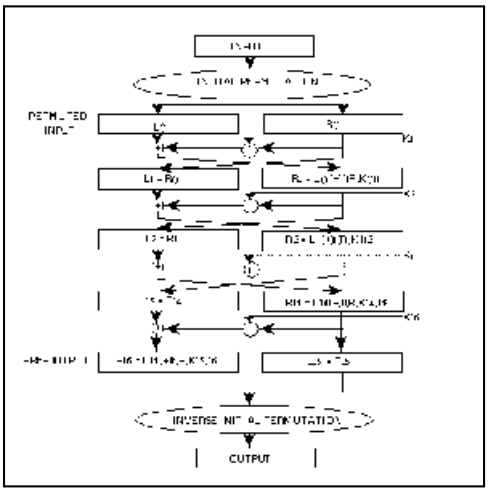
D(C) = M.

Program yang dibuat mengkombinasikan kedua metode pada proses enkripsi atau dekripsinya. Pada proses enkripsi, proses dikenakan metode DES terlebih dahulu baru kemudian metode Gost. Pada proses dekripsinya, kebalikan dari proses enkripsi, yaitu metode Gost terlebih dahulu baru kemudian metode DES.

Metode DES

DES merupakan kepanjangan dari *Data Encryption Standard* yang dibuat oleh National Berau of Standard USA pada tahun 1977 dan menjadi standar ANSI pada tahun 1981[2]. Algoritma DES dirancang untuk membaca berita blok data atau *plaintext* sebesar 64 bit / 8 byte di bawah kontrol kunci sebesar 64 bit / 8 byte[3], sebanyak 16 putaran.

Untuk lebih jelasnya, gambar flowchart proses enkripsinya sebagai berikut :



Gambar 3.2 Flowchart enkripsi dengan metode DES

Pertama-tama, input berupa 64 bit data akan disubtitusi dengan blok IP (*initial permutation*) dengan ketentuan sebagai berikut :

IP = 58, 50, 42, 34, 26, 18, 10, 2,
60, 52, 44, 36, 28, 20, 12, 4,
62, 54, 46, 38, 30, 22, 14, 6,
64, 56, 48, 40, 32, 24, 16, 8,
57, 49, 41, 33, 25, 17, 9, 1,
59, 51, 43, 35, 27, 19, 11, 3,
61, 53, 45, 37, 29, 21, 13, 5,
63, 55, 47, 39, 31, 23, 15, 7.

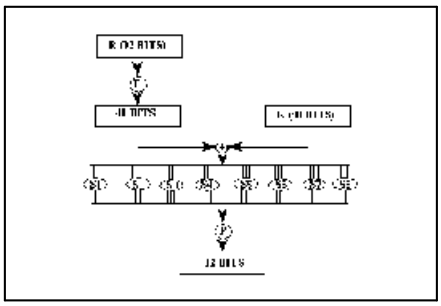
Perhitungan di atas sebagai berikut : blok IP akan memindahkan bit ke-58 dari *plaintext* menjadi bit ke-1, bit ke-50 menjadi bit ke-2, bit ke-42 menjadi bit ke-3, dan seterusnya. Perhitungan di atas berlaku sama untuk perhitungan pada blok-blok permutasi lainnya.

IP (*initial permutation*) dan *final permutation* / *invers* IP tidak mempengaruhi keamanan dari DES. Tujuan utamanya yaitu hanya untuk memudahkan dalam memanggil data *plaintext* ke dalam chip DES yang berbentuk potongan-potongan *byte* dalam s*oftware* / program komputer[1].

Plaintext yang telah disubstitusi oleh blok IP di atas akan dipecah menjadi dua bagian sebesar 32 bit kiri (L) dan 32 bit kanan (R). Pada setiap putarannya data kiri akan menjadi data kanan, dan data kanan akan dilakukan operasi : data kiri di-Xor-kan dengan *fungsi* f :

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \text{ Xor } f(R_{i-1}, K_i)$$

Berikut adalah gambar dari pembentukan *fungsi* f :



Gambar 3.3 Flowchart fungsi f (metode DES)

Pada *fungsi* f, data sebelah kanan (R) sebesar 32 bit akan dipermutasi dengan *Expansion Permutation* (E) sehingga menghasilkan *ciphertext* sebesar 48 bit. Kemudian dilakukan operasi Xor dengan blok kunci (48 bit) dan diinputkan ke dalam Sbox yang terdiri dari 8 buah Subtitution box (Sbox). Hasilnya akan membentuk data menjadi 32 bit lagi sekaligus disubstitusi lagi dengan P-Box Permutation (P). Berikut isi untuk E dan P :

E = 32, 1, 2, 3, 4, 5, 4, 5,
6, 7, 8, 9, 8, 9, 10, 11,
12, 13, 12, 13, 14, 15, 16, 17,
16, 17, 18, 19, 20, 21, 20, 21,
22, 23, 24, 25, 24, 25, 26, 27,
28, 29, 28, 29, 30, 31, 32, 1.

P = 16, 7, 20, 21, 29, 12, 28, 17,
1, 15, 23, 26, 5, 18, 31, 10,
2, 8, 24, 14, 32, 27, 3, 9,
19, 13, 30, 6, 22, 11, 4, 25.

Dalam putaran terakhirnya, antara blok R_{16} dan L_{16} tidak terjadi pertukaran data tetapi kedua blok ini menjadi disatukan kembali dan menjadi input untuk *final permutation* atau *invers* IP (IPinv). IPinv merupakan *inverse* dari IP dan digambarkan dengan ketentuan sebagai berikut :

IPInv = 40, 8, 48, 16, 56, 24, 64, 32,
39, 7, 47, 15, 55, 23, 63, 31,
38, 6, 46, 14, 54, 22, 62, 30,
37, 5, 45, 13, 53, 21, 61, 29,
36, 4, 44, 12, 52, 20, 60, 28,
35, 3, 43, 11, 51, 19, 59, 27,
34, 2, 42, 10, 50, 18, 58, 26,
33, 1, 41, 9, 49, 17, 57, 25.

Blok kunci terdiri dari 16 blok, masing-masing berjumlah 56 bit. Sebelumnya, inputan untuk blok kunci berjumlah 64 bit. Kemudian blok tersebut akan disubstitusi terlebih dahulu dengan *Permuted Choice 1* (Pc1) atau disebut juga sebagai *Key Permutation* sehingga menghasilkan blok sebesar 56 bit. Kemudian data tersebut dipecah menjadi 2 bagian dengan setiap

bagian sebesar 28 bit. Setiap blok ke 1, 2, 9, dan 16 akan dirotasi 1 bit ke kiri, sisanya 2 bit ke kiri kemudian digabungkan kembali menjadi 56 bit.

Putaran :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Jumlah bit subkunci yang dirotasi / *shifted* (ke kiri) :

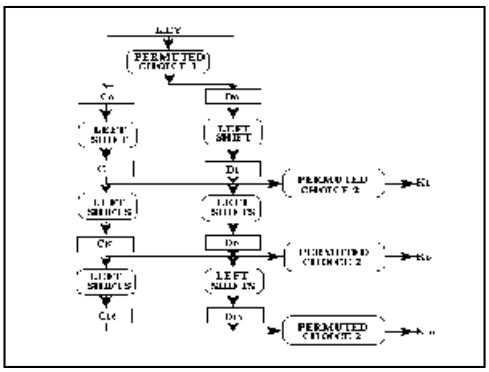
1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

Terakhir pada proses blok kunci akan dilakukan permutasi kembali dengan *permuted Choice 2* (Pc2) atau disebut juga sebagai *Compression Permutation* sehingga blok kunci menjadi 48 bit. Berikut isi dari Pc1 dan Pc2 :

Pc1 = 57, 49, 41, 33, 25, 17, 9, 1,
58, 50, 42, 34, 26, 18, 10, 2,
59, 51, 43, 35, 27, 19, 11, 3,
60, 52, 44, 36, 63, 55, 47, 39,
31, 23, 15, 7, 62, 54, 46, 38,
30, 22, 14, 6, 61, 53, 45, 37,
29, 21, 13, 5, 28, 20, 12, 4.

Pc2 = 14, 17, 11, 24, 1, 5, 3, 28,
15, 6, 21, 10, 23, 19, 12, 4,
26, 8, 16, 7, 27, 20, 13, 2,
41, 52, 31, 37, 47, 55, 30, 40,
51, 45, 33, 48, 44, 49, 39, 56,
34, 53, 46, 42, 50, 36, 29, 32.

Berikut gambar dari pembentukan 16 blok kunci :



Gambar 3.4 Flowchart blok kunci DES

Untuk proses dekripsinya, operasinya akan dibalik dengan operasi enkripsi, tetapi dengan urutan kunci terbalik :

$L_i = R_{i-1}$

$R_i = L_{i-1}$ Xor $f(R_{i-1},K_i)$

Perlu diperhatikan di sini yaitu setiap blok : IP, *invers* IP, Sbox, blok P, blok E, blok Pc1 dan blok Pc2 pada metode DES adalah blok yang isinya merupakan ketetapan dari metode DES.

Metode Gost

Hasil enkripsi dari metode DES di atas akan menjadi inputan untuk enkripsi selanjutnya yaitu dengan metode Gost. Gost merupakan blok *cipher* dari bekas negara Uni Sovyet, yang merupakan singkatan dari "Gosudarstvennyi Standard" atau Standar Pemerintah. Standar ini bernomor 28147-89, oleh sebab itu metode ini sering disebut sebagai GOST 28147-89.

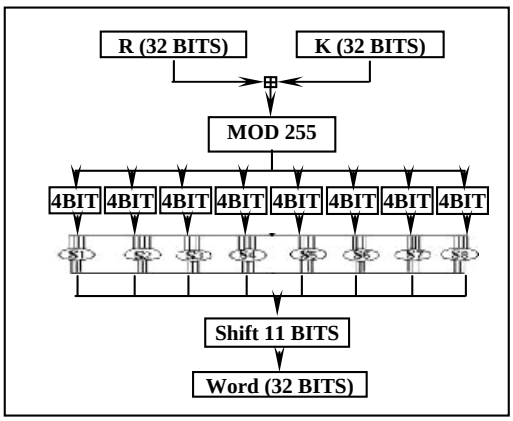
Hampir sama seperti metode DES, Gost merupakan algoritma dengan input data atau *plaintext* sebesar 64 bit / 8 byte, tetapi panjang kuncinya sebesar 256 bit (32 byte). Algoritma ini mengiterasi algoritma enkripsi sederhana sebanyak 32 putaran (*round*) [4].

Untuk mengenkripsi, pertama-tama *plaintext* 64 bit dipecah masing-masing menjadi bilangan 32 bit pada bagian kiri (L), dan bilangan 32 bit pada bagian kanan (R). Pada setiap putarannya, data kiri akan menjadi data kanan, dan data kanan akan dilakukan operasi : data kiri di-Xor-kan dengan *fungsi* f :

$L_i = R_{i-1}$

$R_i = L_{i-1}$ Xor $f(R_{i-1},K_i)$

Untuk fungsi f-nya, pada setiap putarannya data sebelah kanan akan dilakukan operasi penambahan dengan blok kunci yang berbeda dimodulus dengan 2^{32} atau untuk hitungan per byte, setiap bytenya dari bilangan tersebut akan dimodulus 2^8 / 255 - dalam hitungan komputer (fungsi modulus yaitu agar setiap bilangan tidak melebihi 255 untuk hitungan per byte – dalam komputer).



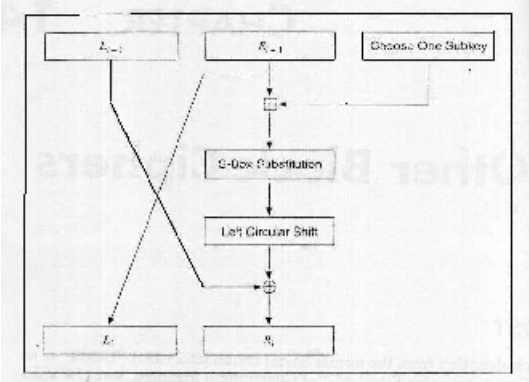
Gambar 3.5 Flowchart fungsi f (Metode Gost)

Baru kemudian data yang dihasilkan akan dipecah menjadi 8 bilangan (masing-masing 4 bit) dan dijadikan input yang berbeda untuk tiap Sboxnya (berjumlah 8 buah). Algoritma Gost tidak menentukan standar

dalam pembentukan Sboxnya. Hal ini menjadi salah satu penyebab kuat / tidaknya algoritma ini dalam implementasinya.

Terakhir data akan dikombinasikan lagi menjadi 32 bit, dan dirotasi 11 bit ke kiri serta menjadi bilangan 32 bit kembali.

Berikut gambar flowchart untuk satu putaran dari algoritma Gost :



Gambar 3.6 Flowchart satu putaran enkripsi (metode Gost)

Subkunci untuk Gost terdiri dari 8 buah blok dengan panjang setiap bloknya sebanyak 32 bit. Untuk mempersulit, maka setiap bloknya akan dirotasi 11 bit ke kiri sebelum dilakukan operasi penambahan (byte per byte) dengan data kanan – pada fungsi f.

Berikut adalah keterangan dari penggunaan kunci dalam tiap putarannya pada metode Gost:

Putaran :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Subkunci:
1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
Putaran :
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Subkunci :
1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1

Untuk Sboxnya, penulis menggunakan Sbox yang sama dengan yang digunakan oleh *Central Bank of the Russian Federation* pada tahun 1994[1]. Hal ini diharapkan dapat memperkuat algoritma Gost yang digunakan. Berikut isi dari 8 Sboxnya :

S-box 1:
4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3
S-box 2:
14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9
S-box 3:
5 8 1 13 10 3 4 2 14 15 12 7 6 0 9 11
S-box 4:
7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3
S-box 5:
6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2
S-box 6:
4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14
S-box 7:
13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12
S-box 8:
1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12

Untuk proses dekripsinya, operasinya akan dibalik dengan operasi enkripsi, tetapi dengan urutan kunci terbalik :

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \text{ Xor } f(R_{i-1}, K_i)$$

4. Implementasi Sistem

Program yang dibuat hanya dapat melakukan proses enkripsi dan dekripsi pada satu *file* saja. Hal ini berkaitan dengan lamanya waktu proses enkripsi dan dekripsi yang didapat dari beberapa *file* yang ukurannya mencapai lebih dari 10 MB. Tetapi hal ini nantinya diharapkan akan dapat diatasi oleh *user* menggunakan spesifikasi *hardware* lebih tinggi daripada yang digunakan oleh penulis pada waktu uji coba sebelumnya. Berikut adalah tabel hasil dari beberapa file yang telah dilakukan proses enkripsi dan dekripsi dalam uji coba tersebut (dengan menggunakan komputer #1) :

Tabel 4.1 Perbandingan waktu proses

No	Nama File	Ukuran Sebelum (Bytes)	Ukuran Sesudah (Bytes)	Waktu Enkrip (Detik)	Waktu Dekrip (Detik)
1	Data.rar	1.0	0.0	0.02	0.01
2	Readme.txt	9,59	9,59	1,07	1,07
3	Tugas1.doc	32768	32768	3,07	3,09
4	Full_TA3	33280	33281	4,27	4,29
5	Opening.wav	2524436	2525252	30,59	31,1
6	3705 Acrobat.CHM	3499519	3499568	41,12	41,54
7	Detail Design.mdb	6722048	6722048	80,56	79,89
8	LAM Jurkom.doc	10021440	10021536	121,69	123,82
9	Lil' Myself.mp3	5241284	5241264	704,69	655,29
10	...	7107584	7107680	971,81	889,67
11	01 - Elevation.mf	14899840	14958320	1111,64	1840,35

Dari hasil di atas, dapat diketahui bahwa kecepatan rata-rata untuk proses enkripsi yaitu :

$$\frac{29809876}{4004,9} = 7443,35 \text{ byte / detik.}$$

Sedangkan untuk proses dekripsinya, kecepatan rata-ratanya mencapai :

$$\frac{29810706}{3775,02} = 7896,83 \text{ byte / detik.}$$

Kecepatan rata-rata pada proses dekripsi lebih cepat. Hal ini disebabkan karena proses melakukan dekripsi dengan

menggunakan metode Gost terlebih dahulu yang operasinya tidak sampai kepada perhitungan bilangan-bilangan bit seperti pada metode DES, melainkan hanya sampai kepada bilangan-bilangan *byte*, baru kemudian proses melanjutkan dekripsi dengan metode DES.

Operasi dalam menghasilkan kunci juga mempengaruhi kecepatan proses enkripsi dan dekripsinya. Pada metode DES diperlukan dua kali permutasi (Pc1 dan Pc2) untuk menghasilkan kuncinya, sedangkan pada Gost hanya mengambil bilangan 32 bit dari 256 bit dan melakukan *shift* 11 bit ke kiri.

5. Kesimpulan

- Penyandian dengan metode DES yang dikombinasikan dengan metode Gost cukup memadai untuk menangkal penyerangan data oleh *user* yang tidak berhak terutama ditinjau dari tingkat kesulitan dalam memecahkan penyandian
- Perancang Gost mencoba mengambil keseimbangan antara efisiensi dan keamanan data. Mereka memodifikasi design dasar DES untuk menciptakan algoritma yang lebih baik dalam implementasinya pada *software*[1]. Mereka juga kurang yakin dalam keamanan algoritmanya, dan mencoba untuk melengkapinya dengan : membuat panjang kunci yang sangat panjang (256 bit), tetap merahasiakan isi dari Sboxnya, dan melipatgandakan jumlah iterasinya (32 putaran).
- Pengenkripsian kunci pada awal program dan kunci pada metode Gost dapat lebih menyulitkan dalam memecahkan penyandian pada *ciphertext*.
- Lamanya proses enkripsi dan dekripsi yang dilakukan dipengaruhi oleh kecepatan dan kemampuan komputer yang digunakan serta besarnya ukuran *file*. Selain itu, kecepatan prosesnya juga dipengaruhi oleh algoritma yang digunakan

Daftar pustaka

[1] Bruce Schneier, 1996, *Applied Cryptography : Protocols, Algorithms, and Source Code in C*, second edition, John Willey & Sons Inc, Kanada.

[2] WAHANA Komputer, 2003, Memahami Model Enkripsi & Security Data, ANDI, Yogyakarta.

[3] Gaithersburg, January 1993, *Data Encryption Standard (DES-FIPS PUB 46-2)*, National Institute standards and Technology.

[4] J. Kelsey, B. Schneier, D. Wagner, *Key-Schedule Cryptanalisis of IDEA, G-DES, GOST, SAFER, and Triple DES*, 1996.