KURSOR Jurnal Ilmiah Menuju Solusi Teknologi Informasi
Vol. 3, No. 2, Juli 2007
ISSN 0216 - 0544

PEMBUATAN SISTEM KEAMANAN FILE DENGAN MENGGUNAKAN ALGORITMA BLOWFISH

Bain Khusnul Khotimah
Email: bain@trunojoyo.ac.id

Jurusan Teknik Informatika
Universitas Trunojoyo

ABSTRAK

Dalam era konektifitas elektronik universal, hacker, virus, penipuan elektronik maupun mendengar diam-diam secara elektronik, maka keamanan data benar-benar menjadi permasalahan yang sangat penting, terutama pada jaringan komputer. Pada jaringan komputer banyak sekali serangan (*attack*). Untuk mengamankan data atau message dijaringan diperlukan cryptography dengan metode encryption. Metode yang digunakan adalah Algoritma Blowfish. Kegunaan algoritma ini adalah menghasilkan suatu *self-decryption archive* setelah mengenkripsi file dan folder. Untuk mendekripsi file dan folder, user harus menjalankan archive tersebut juga memberikan *password* dan lintasan tujuan (*destination*). Penelitian ini juga akan menampilkan program simulasi untuk menampilkan bagaimana mengenkripsi/mendekripsi file dan folder serta mensplit archive tersebut.

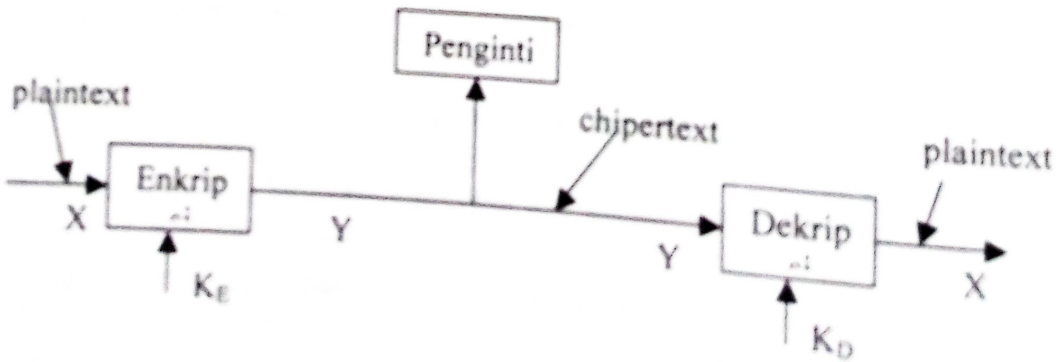
Kata Kunci: *hacker, blowfish, password, decription, file, folder*

1. PENDAHULUAN

Dalam era konektifitas elektronik universal, hacker, virus, penipuan elektronik maupun mendengar diam-diam secara elektronik, maka keamanan data benar-benar menjadi permasalahan yang sangat penting. Perkembangan sistem komputer dan interkoneksinya melalui jaringan telah meningkat, tentu saja hal ini membutuhkan keamanan data dan message yang handal agar terhindar dari serangan (attack)[3]. Untuk mengamankan data atau message dijaringan diperlukan *cryptography* dengan metode *encryption*. Salah satu metode *encryption* data/folder yang akan dibahas dalam penelitian ini adalah Algoritma Blowfish.

2. KONSEP DASAR CRYPTOSYSTEM

Cryptoystem digunakan untuk menjamin privasi dan autentik data dalam sistem komputer-komunikasi. Message yang tidak diproteksi disebut *plaintext*. Proses tersebut yang mana *plaintext* dibentuk dalam *ciphertext* dari suatu bentuk yang tak dapat dipahami yang disebut enkripsi atau *enchiperment*. Sebuah algoritma *dechipering* digunakan untuk dekripsi atau *decipherment* agar mengembalikan *plaintext* aslinya. Dalam *cryptosystom*, sekumpulan parameter yang memilih sebuah transformasi chipering khusus yang sebut sekumpulan key. Enkripsi dan dekripsi dikontrol oleh sebuah key atau beberapa key seperti ditunjukkan *cryptosystem* di bawah ini:



Gambar 1. Cryptosystem Secara Umum

Operasi enhipering dan deciphering dijelaskan secara umum sebagai berikut :

$Y = E_{K_E}(X)$ (enkripsi)

$X = D_{K_D}(Y)$ (dekripsi)

dimana: X = plaintext, Y = chipertext, K_E = key enkripsi, K_D = key dekripsi

2.1. Algoritma *Public-Key*

Algoritma *public-key* juga disebut algoritma asymmetric yang dirancang sehingga key yang digunakan untuk enkripsi berbeda dengan key yang digunakan untuk dekripsi. Selanjutnya key dekripsi tidak dapat dihitung dari key enkripsi. Algoritma tersebut disebut *public-key* karena key enkripsi dapat dibuat secara public. Orang asing dapat menggunakan key enkripsi tersebut untuk mengenkripsi sebuah message, tetapi hanya seorang tertentu dengan key dekripsi sepadan dapat mendekripsi message tersebut. Dalam sistem ini key enkripsi sering disebut *public key* dan key dekripsi disebut *private key*.

Enkripsi dengan public key (K) dinotasikan dengan :

$E_K(M) = C$

Dan didekripsi dengan private key dengan notasi sebagai berikut:

$D_K(C) = M$

Kadang-kadang message akan dienkripsi dengan private key dan didekripsi dengan public key, seperti yang digunakan dalam digital signatures.

2.2 Algoritma Blowfish

2.2.1. Enkripsi Algoritma Blowfish

Blowfish adalah cipher blok 64-bit yang memiliki sebuah kunci yang panjangnya variabel. Algoritma blowfish terdiri dari dua bagian yaitu key expansion dan enkripsi data..Key expansion mengkonversikan sebuah kunci sampai 448 bit ke dalam beberapa array subkey dengan total 4168 byte. Enkripsi data terdiri dari sebuah fungsi yang sederhana dengan iterasi 16 kali. Setiap round mempunyai sebuah permutasi key-dependent dan sebuah subsitusi key- dan data-dependent. Semua operasi, penjumlahan dan XOR pada word 32-bit. Hanya operasi tambahan diindek empat lookup data array per round. Blowfish menggunakan sejumlah subkey yang besar. Key ini harus dihitung awal sebelum enkripsi atau dekripsi.

P-array mempunyai 18 subkey 32-bit :

$P_1, P_2, P_3, \ldots\ldots\ldots\ldots P_{18}$

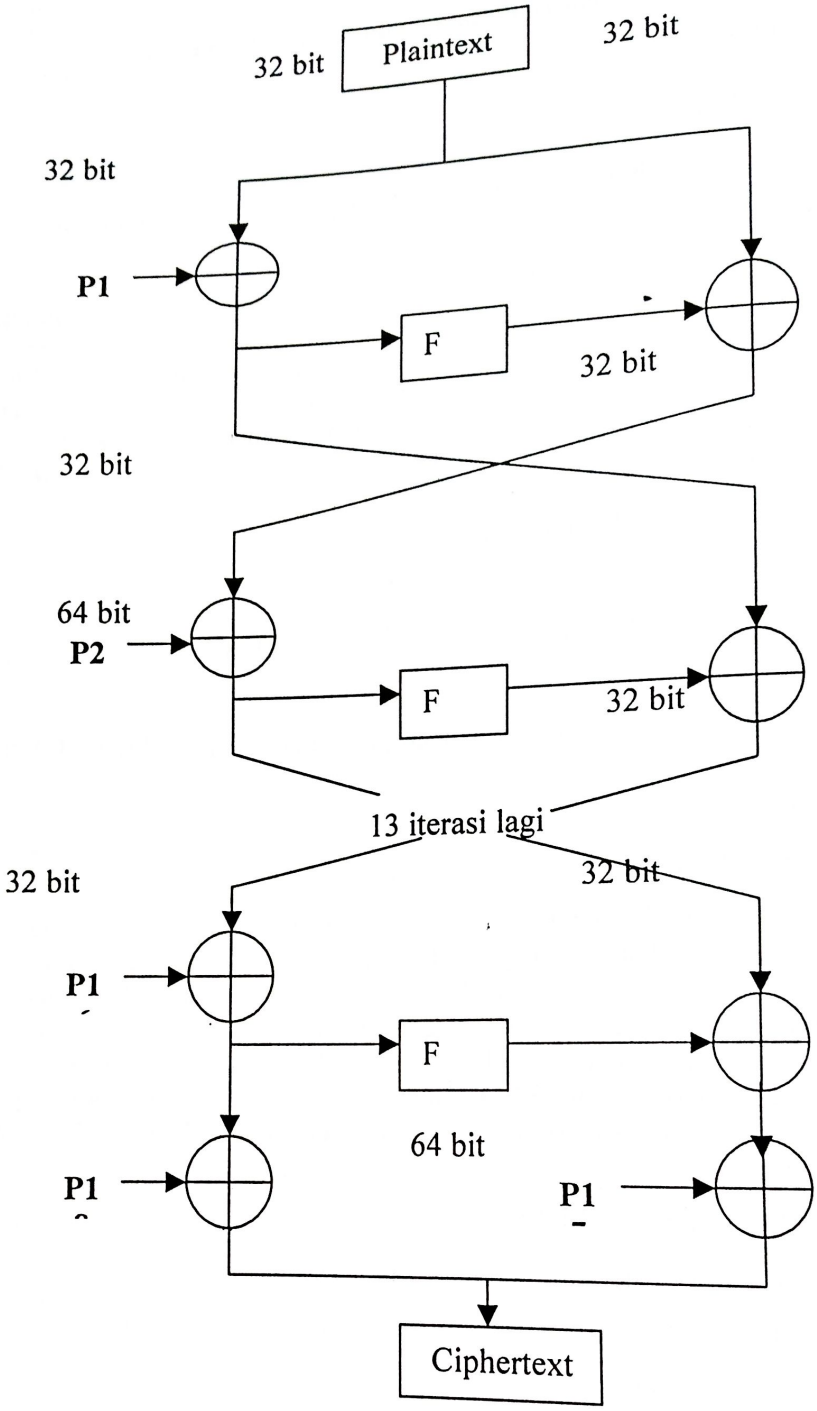
Empat S-box 32- bit mempunyai masing-masing 256 entry [1, 7] yaitu :

$S_{1,0}, S_{1,1}, S_{1,2}, S_{1,3}, \ldots\ldots\ldots\ldots S_{1,255}$

$S_{2,0}, S_{2,1}, S_{2,2}, S_{2,3}, \ldots\ldots\ldots\ldots S_{2,255}$

$S_{3,0}, S_{3,1}, S_{3,2}, S_{3,3}, \ldots\ldots\ldots\ldots S_{3,255}$

$S_{4,0}, S_{4,1}, S_{4,2}, S_{4,3}, \ldots\ldots\ldots\ldots S_{4,255}$



Gambar 2 Blok diagram Algoritma Enkripsi Blowfish

Blowfish adalah sebuah jaringan Feistel yang mempunyai 16 round. Inputnya adalah (x) element data 64-bit. Untuk mengenkripsi (x) yaitu :

Bagi (x) dalam dua bagian 32-bit menghasil (x_L) dan (x_R).

Untuk i = 1 sampai 16 maka :

$x_L = x_L \oplus P_i$

$x_R = F(x_L) \oplus x_R$

Swap (tukar) x_L dan x_R

Swap (tukar) x_L dan x_R (mengulang swap yang lalu)

$x_R = x_R \oplus P_{17}$

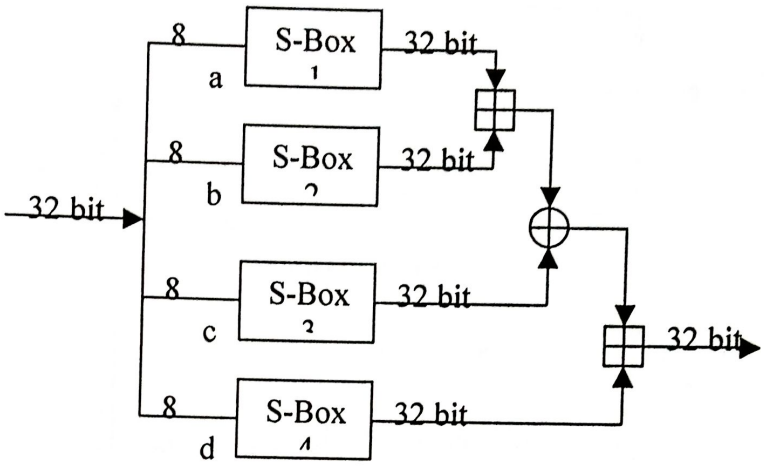
$x_L = x_L \oplus P_{18}$

Gabungkan kembali x_L dan x_R [1, 7]

Fungsi F adalah sebagai berikut [1]:

Bagi x_L dalam empat kuarter 8-bit yaitu a, b, c dan d seperti gambar 3 maka :

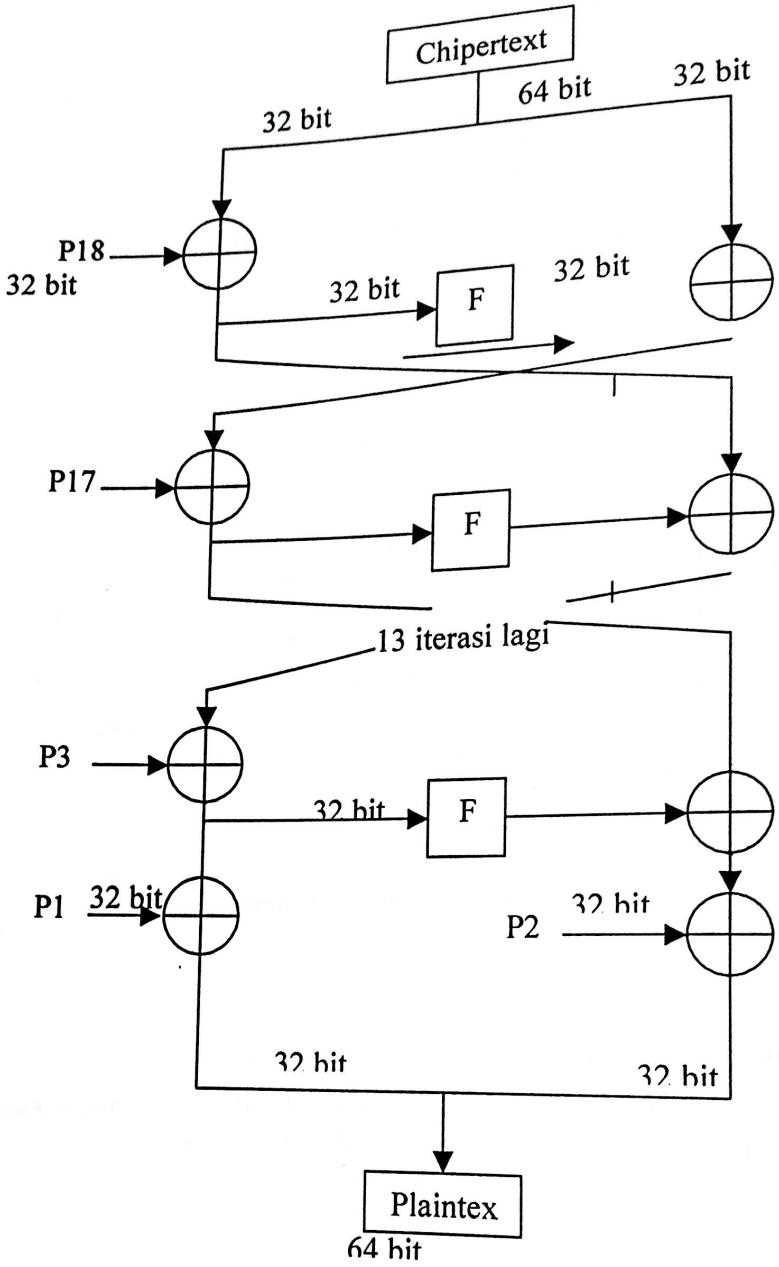
$$F(x_L) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \oplus S_{3,c}) + S_{4,d} \bmod 2^{32}$$



Gambar 3. Fungsi F (Bruce Schenier: 1996)

2.2.2. Dekripsi Algoritma Blowfish

Dekripsi sama persis dengan enkripsi, kecuali bahwa P_1, P_2,..., P_{18} digunakan pada urutan yang berbalik (reverse) [1, 7]. Blok diagram dekripsi seperti pada gambar 4.



Gambar 4. Blok Diagram dekripsi Blowfish

Dengan membalikkan 18 subkey untuk medekripsi metode algoritma Blowfish. Pertama, masalah ini nampak tidak dapat dipercaya, karena ada dua XOR operasi yang mengikuti pemakaian f-fungsi yang sebelumnya, dan hanya satu yang sebelumnya pemakaian pertama f-fungsi. Meskipun jika kita memodifikasi algoritma tersebut sehingga pemakaian subkey 2 sampai 17 menempatkan sebelum output f-fungsi yang di-XOR-kan ke sebelah kanan blok dan dilakukan ke data yang sama sebelum XOR itu, walaupun itu berarti ia sekarang berada di sebelaha kanan blok, karena XOR subkey tersebut telah dipindahkan sebelum swap (tukar) kedua belah blok tersebut (tukar separuh blok kiri dan separuh blok kanan). Kita tidak merubah suatu apapun karena informasi yang sama di-XOR-kan ke separuh blok kiri antara setiap waktu, informasi ini digunakan sebagai input f-fungsi. Kenyataannya, kita mempunyai kebalikan yang pasti dari barisan dekripsi.

2.3. Jaringan Feistel

Banyak algoritma menggunakan jaringan Feistel. Ide ini muncul pada tahun 1970. Bila kita ambil sebuah blok dengan panjang n dan bagilah blok tersebut dengan dua bagian sepanjang n/2 yaitu bagian L (kiri0 dan bagian R (kanan). Tentu saja panjang n adalah genap. Kita dddapat mendefinisikan sebuah cipher blok iterasi dimana output round ke-I ditentukan dari output round sebelumnya [1,6]:

$$L_i = R_{i-1}$$

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

K_i adalah subkey yang digunakan dalam round ke-I dan f adalah fungsi round arbitrary.

3. PEMBAHASAN

3.1 Analisis Algoritma Blowfish

Blowfish terdiri dari 16 round (putaran). Untuk setiap putaran, pertama XOR-kan separuh kiri blok dengan subkey untuk round itu. Kemudian menerapkan f-fungsi ke separuh blok kiri dan XOR-kan separuh kanan blok dengan hasil tersebut. Terakhir setelah semua dieksekusi tetapi round terakhir saling tukar (*swap*) separuh kiri dan kanan blok. Hanya ada satu subkey untuk setiap round, dan f-fungsi mengkonsumsi tanpa subkey tetapi menggunakan S-box yang mana S-box tersebut adalah keydependent (ketergantungan key).

Setelah round terakhir, XOR-kan separuh kanan blok dengan subkey 17 dan XOR-kan separuh kiri dengan subkey 18. Kemudian gabungkan blok kiri dan kanan, sehingga sekarang kita mendapat fungsi informasi yang terenkripsi sebagai *chipertext*.

Kebanyakan bagian dari Blowfish yang menarik adalah f-fungsi yang tidak membalik. Fungsi ini menggunakan aritmatik modular untuk membangkitkan index-index ke dalam S-box.Ambil fungsi $f(x) = x^2 \bmod 7$, lihat tabel 1 dibawah ini:

Tabel 1 Contoh fungsi $f(x) = x^2 \bmod 7$

X	X^2	X^2 Mod 7
1	1	1
2	4	4
3	9	2
4	16	2
5	25	4
6	36	1
7	49	0

Menghasilkan suatu output yang tidak ada fungsi yang dapat menghasilkan input khusus ke f(x). Sebagai contoh jika kita mengetahui bahwa fungsi kita mempunyai sebuah nilai 4 di beberapa nilai X, maka tidak ada cara untuk mengetahui jika nilai X tersebut adalah 2; 5; atau nilai X yang lain yang mempunyai fungsi f(x) = 4. Blowfish melakukan aritmatinya sebesar mod 2^{32} (2^{32} sama dengan 4 milyar). Ini disebut aritmatik dalam bidang berhingga dan membuat banyak asumsi matematika yang sama yang tidak benar (1+1 tidak sama dengan 2 jika kita berada disebuah bidang ukuran 2 yang berhingga).

S-box adalah array yang besar dari data yang didefinisikan sebelumnya. Selama proses setup key, key tersebut menggabungkan dengan S-box. Detail key-setup ini relatif tidak menarik tetapi kenyataanya bahwa ia menggabungkan key tersebut dengan S-box yang menguatkan algoritma

tersebut. Key-setup dalam Blowfish dirancang relatif lamban. Hal ini sangat bermanfaat karena seseorang akan melakukan suatu search-key brute-force yang akan menuju proses key-setup yang lamban untuk setiap key yang dicobanya. Meskipun seseorang melakukan enkripsi dan dekripsi harus hanya menuju proses key-setup satu kali, maka proses enkripsi dan dekripsi relatif cepat.

Elemen yang terpenting Blowfish yang lain adalah jaringan Feistel. Menggunakan jaringan Feistel yang menghasilkan cipher dengan dua sifat yang dapat diinginkan yaitu dekripsi menggunakan fungsi (f) yang sama dan kemampuan untuk mengiterasi fungsi tersebut beberapa kali. Beberapa iterasi ini disebut round (putaran). Semakin banyak round maka semakin banyak keamanan algoritma tersebut. Jumlah round yang direkomendasikan tergantung pada aloritma khusus; untuk Blowfish adalah 16 round.

3.1.1. Subkey Generation (Membangkit Subkey)

Mulai dengan menginisial subkey 1 sampai 18 diikuti element zero sampai 255 dari S-box pertama kemudian element zero sampai 255 dari S-box kedua, dan seterusnya sampai S-box keempat, dengan bagian pecahan pi. Bit signifikan yang terbesar dari pecahan pi menjadi bit signifikan terbesar dari subkey yang pertama. Kemudian tentukan key tersebut yang panjangnya sampai 72 byte dan mengulanginya untuk merentangkan semua array 18 subkey, dan XOR-kan key tersebut dengan isi array subkey.

Kemudian mengeksekusikan algoritma Blowfish berulang-ulang dengan sebuah input inisial blok all zero 64-byte sebagai input. Setelah setiap eksekusi gantikan bagian subkey atau S-box dengan output Blowfish secara berurut, dalam urutan yang sama sebagai digit pi dalam bentuk biner atau hexadecimal ditempatkan di dalamnya; setelah iterasi pertama gantikan subkey 1 dan 2; setelah iterasi ke sepuluh gantikan dua entri pertama (0 dan 1) dalam S-box 1; dan seterusnya.

Untuk setiap itersi Blowfish dalam key generation juga menggunakan iterasi sebelumnya sebagai input. Seperti yang dijelaskan oleh jurnal Dr Dobb pada april 1994 yang dapat diterjemahkan untuk mengimplikasi zero itu dan digunakan sebagai input untuk setiap iterasi. Seperti iterasi selanjutnya hanya merubah masing-masing S-box entri, hal ini dapat mengatur sederetan data identik yang besar dalam S-box dan ini adalah salah pembacaan arah, bukan bentuk asli yang berbeda dari algoritma tersebut.

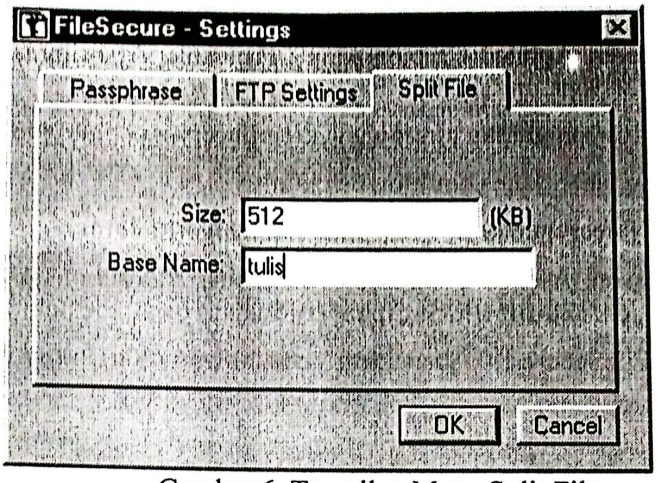
3.1.2. F-Fungsi

Blowfish menggunakan empat S-box. Setiap S-box mempunyai 256 entri dan setiap entri adalah 32 bit panjangnya. Untuk menghitung f-fungsi: gunakan byte pertama 32 bit input untuk mendapatkan sebuah entry dalam S-box pertama, byte kedua untuk mendapatkan sebuah entri S-box kedua dan seterusnya. Nilai f-fungsi adalah $((S1(B1)+S2(B2) \bmod 2^{32})$ XOR $S3(B3)) + S4(B4) \bmod 2^{32}$. [5]

3.2. Simulasi Split File

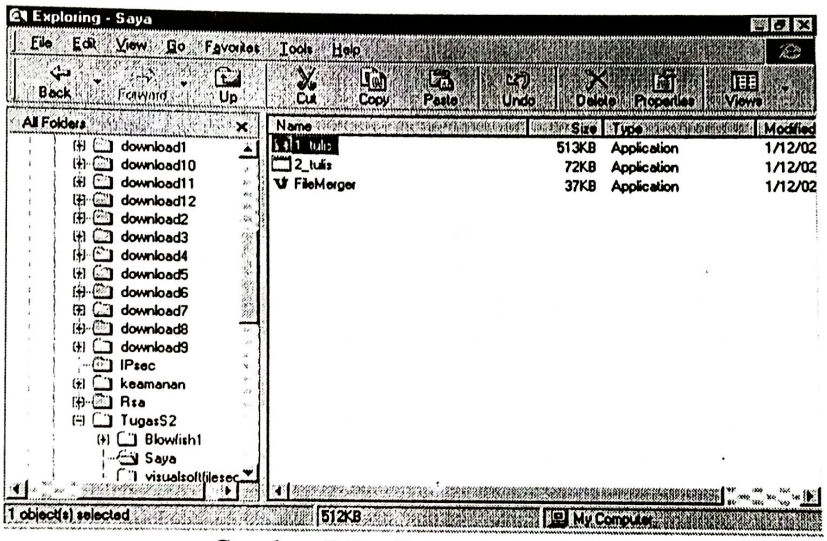
Fasilitas yang tersedia yang ada pada program VisualSoft File Secure yaitu split file. Split file sangat berguna bagi file yang berukuran besar yang ingin kita enkripsi. Setelah program ini dijalankan, tampil seperti gambar 5 dan kita pilih file yang akan kita enkripsi, kemudian pilih Option pada menu bar

selanjutnya pilih Split File dan muncul seperti gambar 11 berikut. Pada gambar 11 adalah mensimulasikan file tulisan1 yang berkapasitas 1,120k byte dan file ini displit dengan ukuran 512 k byte dengan base name tulis. Hasilnya terlihat seperti pada gambar 6, ada tiga file yaitu file 1_tulis,



Gambar 6. Tampilan Menu Split File

2_tulis dan Filemerger.



Gambar 7 Hasil simulasi enkripsi split file

4. KESIMPULAN

Dari analisa algoritma dan simulasi program Blowfish dapat disimpulkan sebagai berikut:

1 Selama proses key set-up algoritma Blowfish, key ini digabungkan dengan S-box sehingga menguatkan algoritmanya.

2 Dengan menggunakan jaringan Feistel maka algoritma Blowfish mempunyai dua sifat: dekripsi menggunakan f-fungsi yang sama (*non-invertible function*) dan kemampuannya mengiterasi fungsi banyak kali (*multiple times*).

3 Blowfish bekerja dengan menggabungkan sebuah f-fungsi *non invertible*, *keydependent* S-box, dan jaringan Feistel.

4 Dalam proses simulasi file/folder data maupun split file enkripsi dalam program algoritma Blowfish ini menggunakan key dengan minimum 6 karekter.

5. DAFTAR PUSTAKA

1. Bruce Schneier, Applied Cryptography : Protocols, Algorithms, and Source Code in C, USA, John Wiley & Sons, Inc., 1996.
2. Concepts of Cryptography http://www.kremlinecrypt.com/crypto
3. http://www.counterpane.com/blowfish.html
4. http://www.finecrypt.net/blowfish encryption algoritm.htm
5. Kent Johansson, A short summary of Blowfish Algorithm : Descryption of a New Variable-Length Key, 64/128-Bit Block Cipher (Blowfish) by Bruce Schneier, 2001
6. VisualSoft FileSecure 1.0, VisualSoft Technologies, 2001 http://www.visualsoft-india.com