PENGAMANAN DATA DENGAN PENGGABUNGAN METODE GOST DAN RC6

Didi Juardi
Politeknik Tri Mitra Karya Mandiri (TMKM)
Jl. By Pass Jomin Dusun Rawasari Desa Jomin Barat Kotabaru Karawang
didi.juardi@gmail.com

ABSTRAK

Secara umum data dikategorikan menjadi dua, yaitu data yang bersifat rahasia dan data yang tidak bersifat rahasia. Data yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Yang sangat perlu diperhatikan adalah data yang bersifat rahasia, dimana setiap informasi yang ada didalamnya akan sangat berharga bagi pihak yang membutuhkan karena data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasanya dilakukan berbagai cara yang tidak sah. Sistem keamanan data dan kerahasiaan data merupakan salah satu aspek penting dalam perkembangan dunia telekomunikasi,khususnya komunikasi yang menggunakan komputer dan terhubung ke jaringan internet. Dalam perancangan penelitian ini akan dibahas analisis kriptografi yang ditekankan pada penggabungan metode RC6 dan metode GOST untuk mengenkripsi dan mendekripsi file

***Kata Kunci**: keamanan data, GOST, RC6*

PENDAHULUAN

Perkembangan teknologi informasi, terutama dalam dunia komputer, maka informasi saat ini telah menjadi kebutuhan pokok sehari-hari bagi banyak pihak, baik itu perorangan, instansi pemerintah maupun swasta.Bagi pihak yang membutuhkan informasi saat ini bisa mendapatkan informasi yang dibutuhkan tidak terbatas pada suatu tempat saja, informasi bisa diperoleh dimanapun selama ada fasilitas internet.

Fasilitas internet yang memang telah banyak dilindungi oleh berbagai sarana untuk menjaga keamanan.Namun hal ini tentu masih belum cukup menjamin keamanan bagi kerahasiaan informasi.Oleh karena itu saat ini telah banyak dikembangkan metode tertentu dalam upaya pengamanan data. Adapun metode pengamanan yang akan dibahas saat ini adalah metode pengacakan data dengan menggunakan dua metode yaitu GOST dan RC6.

METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Kepustakaan
   Studi ini dilakukan dengan membaca literatur-literatur khususnya tentang enkripsi dan dekripsi, algoritma pemrograman, bahasa pemrograman PHP, baik yang berupa kepustakaan buku maupun di *internet*.
b. Metode GOST dan RC6

Enkripsi Dengan Metode Gost

GOST merupakan *blok cipher* dari bekas Uni Sovyet, yang merupakan singkatan dari "**Go**sudarstvennyi **St**andard" atau Standar Pemerintah, standar ini bernomor **28147-89** oleh sebab itu metoda ini sering disebut sebagai **GOST 28147-89**[4].

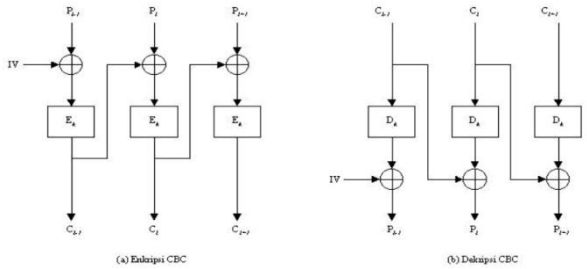
GOST merupakan *blok cipher 64* bit dengan panjang kunci 256 bit. Algoritma ini mengiterasi algoritma enkripsi sederhana sebanyak 32 putaran (*round*). Untuk mengenkripsi pertama-tama *plainteks* 64 bit dipecah menjadi 32 *bit* bagian kiri, L dan 32 bit bagian kanan, R. Subkunci (*subkey*) untuk putaran i adalah K_i. Pada satu putaran ke-i operasinya adalah sebagai berikut:

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

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

Sedangkan pada fungsi f mula-mula bagian kanan data ditambah dengan subkunci ke-i modulus 2^{32}. Hasilnya dipecah menjadi delapan bagian 4 *bit* dan setiap bagian menjadi *input* s-box yang berbeda. Di dalam GOST terdapat 8 buah s-box, 4 *bit* pertama menjadi s-box pertama, 4 *bit* kedua menjadi s-box kedua, dan seterusnya. Output dari 8 s-box kemudian dikombinasikan menjadi bilangan 32 *bit* kemudian bilangan ini dirotasi 11 *bit* kekiri.Akhirnya hasil operasi ini di-xor dengan data bagian kiri yang kemudian menjadi bagian kanan dan bagian kanan menjadi bagian kiri (*swap*). Pada implementasinya nanti rotasi pada fungsi f dilakukan pada awal saat inisialisasi sekaligus membentuk s-box 32 *bit* dan dilakukan satu kali saja sehingga lebih menghemat operasi dan dengan demikian mempercepat proses enkripsi/dekripsi

Pada metode *blok cipher* ada yang dikenal sebagai mode operasi. Mode operasi biasanya mengkombinasikan *cipher* dasar, *feedback* dan beberapa operasi sederhana. Operasi cukup sederhana saja karena keamanan merupakan fungsi dari metoda *cipher* yang mendasarinya bukan pada modenya. Mode pertama adalah ECB (*Electronic Codebook*) dimana setiap blok dienkrip secara independen terhadap blok lainnya. Dengan metode operasi ini dapat saja sebuah pesan disisipkan diantara blok tanpa diketahui untuk tujuan tertentu, misalnya untuk mengubah pesan sehingga menguntungkan si pembobol. Mode lainya adalah CBC (*Cipher Block Chaining*) dimana plainteks dikaitkan oleh operasi xor dengan cipherteks sebelumnya, metoda ini dapat dijelaskan seperti pada Gambar.1. Untuk mode ini diperlukan sebuah *Initialization Vector* (IV) yang akan di-xor dengan plainteks yang paling awal. IV ini tidak perlu dirahasiakan karena bila kita perhatikan jika terdapat n blok maka akan terdapat n-1 IV yang diketahui. Metode lain yang dikenal adalah CFB (*Cipher Feedback*), OFB (*Output Feedback*), *Counter Mode*, dan lain-lain.



Gambar 2. Mode Operasi CBC

Enkripsi Dengan Metode RC6

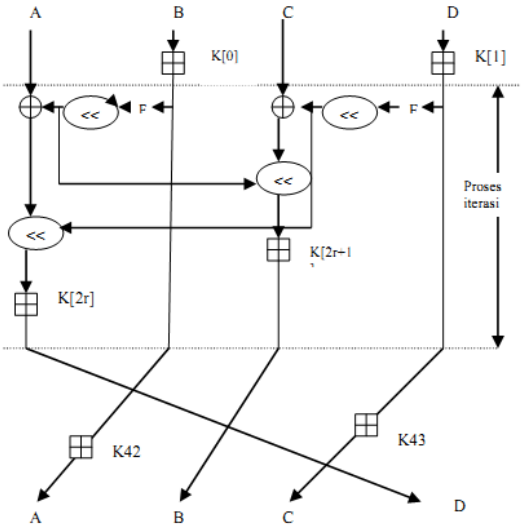
Metode RC6 merupakan pengembangan dari metode RC5, di mana dalam proses enkripsinya menggunakan 6 operasi dasar matematika[5]. Adapun operasi dasar tersebut meliputi sebagai berikut:

1. Operasi penjumlahan.
2. Operasi pengurangan
3. Operasi XOR (*Exclusive* OR)
4. Operasi perkalian
5. Proses pergeseran *bit* ke kanan
6. Proses pergeseran *bit* ke kiri.

Proses enkripsi dengan metode RC6 membagi *plaintext* menjadi blok–blok tertentu, di mana tiap blok berukuran 128 *bit*. Dari 128 *bit* tersebut akan dibagi menjadi 4 blok bagian penting yang sama besar. Katakanlah ke empat blok tersebut adalah A, B, C, dan D. Karena dikatakan bahwa keempat blok tadiberukuran sama besar, maka tiap blok akan berukuran 32 bit. Pada metode RC6 ini melakukan pembentukkan sub kunci yang memiliki panjang 32 *bit*, yang akan diletakkan pada *variabel array* $RC_KEY[0] sampai dengan $RC_KEY[43].

Pada proses ini diawali dengan menjumlahkan nilai blok B dengan $RC_KEY[0], dan nilai blok dengan $RC_KEY[1]. Dan pada masing– masing proses iterasi digunakan 2 buah bagian kunci dari proses pembentukkan kunci, yaitu $RC_KEY[2], dan $RC_KEY[3]. Jumlah iterasi yang dipakai secara umum adalah 20 iterasi, yang pada akhir proses iterasi dilakukan penjumlahan blok A dengan $RC_KEY [42], dan C dengan $RC_KEY [43].

Pada proses pembentukkan kunci metode RC6, menggunakan suatu konstanta x = 0xb7e15163 hexsadesimal yang akan diinisialisasikan terhadap $RC_KEY [0] dan $RC_KEY[1], serta nilai $RC_KEY [4] didapatkan dengan menambahkan konstanta 0x9e3779b9 heksadesimal. Misalnya r = 20, maka jumlah iterasi tersebut adalah 20 putaran. Proses di atas akan dijelaskan pada gambar berikut:



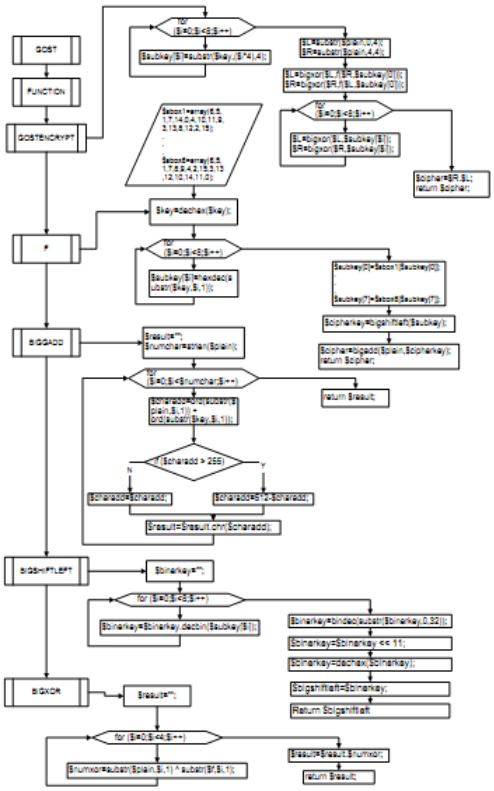
Gambar 3. Diagram Blok Proses Enkripsi Metode RC6

Hasil penjumlahan nilai B dengan $RC_KEY [0], akan dimasukkan ke suatu fungsi yang telah terdefinisi, dan hasilnya akan digeser sepanjang 5 *bit*, simpan hasil ke variabel *$t*. Berikutnya hasil penjumlahan nilai D dengan $RC_KEY[1] dimasukkan ke fungsi, dan hasilnya di geser ke kiri sepanjang 5 *bit*, kemudian hasilnya simpan ke variabel $u. $tdengan niliai A, dan hasilnya digeser ke kiri sejauh $u , serta jumlahkan hasilnya dengan $RC_KEY [2r], kirim hasil operasi ini ke A. Untuk nilai C diXorkan dengan $u dan geser ke kiri sejauh nilai *$t*, serta jumlahkan hasilnya dengan $RC_KEY [2r+1]. Lakukan pertukaran A dengan D, B dengan A, C dengan B, dan D dengan C. Dan proses ini akan dilakukan sebanyak jumlah iterasi r (20).

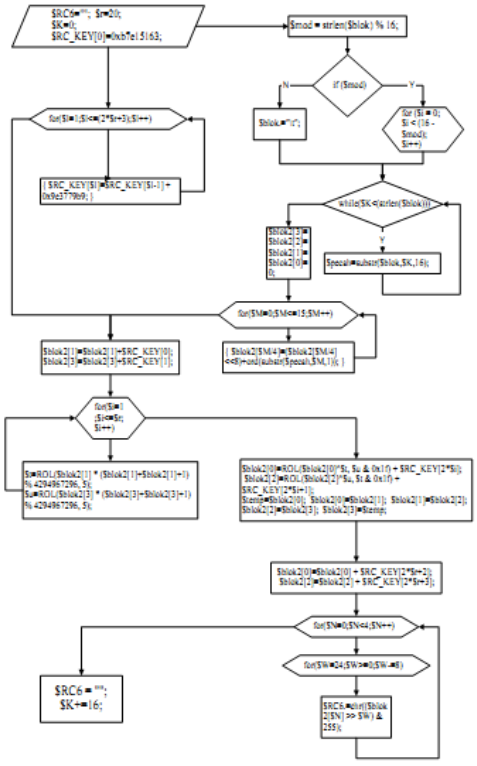
Sedangkan pada proses dekripsinya adalah proses kebalikan dari proses enkripsinya. Yaitu diawali dengan pengurangan nilai C dengan $RC_KEY [43], dan A dengan nilai $RC_KEY[42]. Dan pada tiap iterasi dilakukan dengan urutan kunci lanjutannya. Sampai pada akhir iterasi yaitu pengurangan D dengan $RC_KEY [1], dan B dengan nilai $RC_KEY[0]. Sehingga pada akhir proses emkripsi metode RC6 akan didapatkan hasil akhir susunan blok B, C, D, A.

PEMBAHASAN

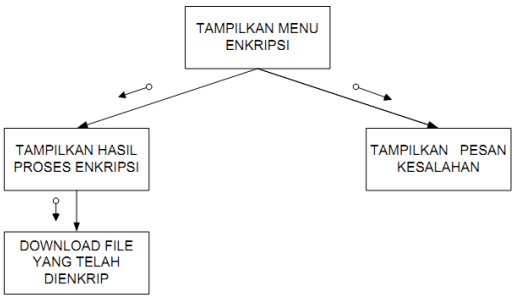
Rancangan Flowchart GOST dan RC6



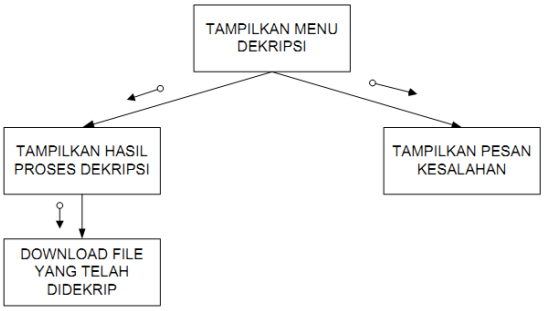
Gambar 4. Rancangan Flowchart GOST



Gambar 5. Rancangan Flowchart RC6



Gambar 6. Structure Chart Enkripsi



Gambar 7. Structure Chart Dekripsi

Proses Algorima Enkripsi

Proses ini menggunakanmetode metode RC6 dan GOST untuk mengubah data sehingga file tersebut tidak dapat dibaca oleh orang lain. Pada proses enkripsi dikerjakan melalui beberapa tahapan yang diantaranya penyimpanan ekstensi sebagai tanda file dan tahapan proses enkripsi *plaintext.* Ekstensi file merupakan tanda file, yang nantinya akan digabungkan dengan kunci dan dienkrip dengan metode sendiri yang akan dijadikan *header* file.

Proses yang pertama dilakukan adalah proses *enkripsi header*, setelah itu baru dilakukan proses enkripsi isi file. Hasil enkrip adalah penggabungan antara enkrip header dan enkrip isi file Sebelum melakukan proses enkripsi user diminta untuk memasukkan kunci yang nantinya akan dijadikan header setelah mengalami proses enkripsi. Header berasal dari proses enkripsi karakter "*NAS" + kunciyang dimasukkan + panjang ekstensi file + nama ekstensi file*. Sebelum proses enkripsi file, header ini akan dienkripsi terlebih dahulu dengan metode sendiri (tidak sama dengan metode enkripsi file).

Header ini nantinya akan dijadikan tanda atau ciri dari enkripsi. Pada proses dekripsi nanti akan dilakukan pengecekan header, bila hasil dekrip header tiga karakter awalnya adalah "NAS" dikuti kunci berarti file tersebut benar hasil enkripsi metode RC6 dan GOST by Didi Juardi, bila tidak maka proses enkripsi tidak bisa dilanjutkan.

Algoritma Enkripsi Metode RC6

Pada metode ini akan dilakukan proses pengeblokan plaintext per enambelas (16) byte atau karakter. Jika ditemukan plaintext kurang dari enambelas karakter maka akan ditambahkan karakter "\t" sebanyak kekuranganya. Adapun algoritma untuk melakukan proses penambahan blok plaintext tersebut adalah ;

```
1. $mod = strlen($teks) % 16;
2.     if ($mod) {
3.             for ($i = 0; $i < (16 - $mod); $i++) {
4.             $teks.="\t";    } }
```

Artinya pada tahap pertama plaintext akan diperiksa panjangnya. Jika kurang dari 16 karakter maka akan ditambahkan karakter "\t" sebanyak dari kekurangannya. Jika sudah ditemukan sebanyak 16 karakter maka akan dilakukan proses pengeblokan terhadap blok plaintext tersebut.

Dari enambelas blok yang terbentuk tersebut akan dipecah lagi menjadi empat bagian blok sama besar yaitu 32 bit.

Algoritma Enkripsi Metode GOST

Ciphertext yang dihasilkan oleh metode RC6akan diambil per delapan karakter oleh metode GOST**,** sebagai berikut: Delapan karakter pertama **á‰> @ • ½,** dan delapan karakter berikutnya adalah **zµŠÌÔô9Û.** Dimana masing – masing blok plaintext panjangnya 64 bit atau 8 karakter (1 karakter mewakili 8 bit). Dari tiap 64 bit blok tersebut akan dibagi menjadi 2 blok bagian sama besar yaitu blok kanan **($R**) dan blok kiri (**$L**). masing – masing pecahan blok memiliki panjang 32 bit. Agar lebih jelas maka tabel 3.9, 3.10 akan memperlihatkan bentukan bloknya.

Tabel 1. Pecahan 64 Bit dari 8 Blok Pertama

Blok kiri (**L**)	Blok kanan (**R**)
á‰>	**@½**

Tabel 2. Pecahan 64 Bit dari 8 Blok Kedua

Blok kiri (**L**)	Blok kanan (**R**)
zµŠÌ	**Ôô9Û**

Sebelum melakukan proses plaintext pada metode GOSTdilakukan dahulu proses pembentukan kunci GOST. Pada metode GOST pertama–tama didefinisikan 16 bilangan sebagai kunci acak, dimana 16 bilangan acak tersebut kita tentukan sendiri, dan bilangannya adalah sebagai berikut;

```
$sbox1=array(6,5,1,7,14,0,4,10,11,9,3,13,8,12,2,15);
$sbox2=array(14,13,9,0,8,10,12,4,7,15,6,11,3,1,5,2);
$sbox3=array(6,5,1,7,2,4,10,0,11,13,14,3,8,12,15,9);
$sbox4=array(8,7,3,9,6,4,14,5,5,13,0,12,1,11,10,15);
$sbox5=array(10,9,6,11,5,1,8,4,0,13,7,2,14,3,15,12);
$sbox6=array(5,3,0,6,11,13,4,14,10,7,1,12,2,8,15,9);
$sbox7=array(2,1,12,3,11,13,15,7,10,6,9,14,0,8,4,5);
$sbox8=array(6,5,1,7,8,9,4,2,15,3,13,12,10,14,11,0);
```

Program yang dibuat dalam skripsi ini membatasi user memasukan password sebanyak 9 karakter, tidak bisa lebih.Padahal pada metode GOSTmenggunakan kunci sepanjang 256 bit atau 32 karakter. Untuk menyesuaikan proses itu, kekurangan kunci sepanjang 9 karakter akan ditambahkan karakter "_" sebanyak kekurangannya.

Proses Algoritma Dekripsi

Pada hakekatnya proses dekripsi adalah proses kebalikan dari proses enkripsi. Jika pada awal proses enkripsi diawali dengan metode RC6 dan dilanjutkan dengan metode GOST, maka pada proses kebalikannya ditandai dengan proses dekripsi dengan metode GOST dan dilanjutkan dengan metode RC6.

Dekripsi Algoritma Metode GOST

Seperti pada sedia kala proses GOST selalu melakukan proses pengambilan pada tiap 8 blok. Agar lebih jelas tabel 3 dan 4 akan memperlihatkan:

Tabel 3. Pecahan 64 Bit dari 8 Blok Pertama Dekripsi GOST

Blok kiri (**L**)	Blok kanan (**R**)
'~ÈU	**‹IF™**

Tabel 4. Pecahan 64 Bit dari 8 Blok Kedua Dekripsi GOST

Blok kiri (**L**)	Blok kanan (**R**)
Đà0	**\ • Ú¦**

Seperti pada awal proses enkripsi, pada proses dekripsi juga digunakan pembentukan kunci GOST**.** Intinya sama persis dalam penerapanya hanya saja pada saat sebelum proses dekripsi dilakukan terlebih dahulu proses pemeriksaan kunci dilakukan yaitu apakah kunci yang dinput sama dengan kunci pada saat melakukan proses enkripsi. Jika ternyata password atau kunci yang dimasukkan adalah sama maka proses pembentukan kunci akan dilakukan. Setelah selesai proses pembentukan kunci dilakukan, maka proses berikutnya adalah proses ***bigxor* .** Sama persis pada waktu proses enkripsi dilakukan proses fungsi ini akan membaca blok plaintext untuk diXorkan dengan blok kuncinya. Dan pada proses ini dihasilkan karakter seperti diperlihatkan pada tabel 5 dan 6.

Tabel 5. Pecahan 64 Bit dari 8 Blok Pertama Bigxor

Blok kiri (**L**)	Blok kanan (**R**)
éMB•	**EzÌY**

Tabel 6. Pecahan 64 Bit dari 8 Blok Kedua Bigxor

Blok kiri (**L**)	Blok kanan (**R**)
>‹Þª	**²ä<**

Blok yang terbentuk hasil proses ***bigxor*** ini dilanjutkan dengan proses Xor antara blok dengan ketentuan rumusan sebagai berikut;

$L=bigxor($L,f($R,$subkey[0]))
$R=bigxor($R,f($L,$subkey[0]))

Dari proses tersebut maka dihasilkan blok plaintext seperti diperlihatkan pada tabel 7 dan 8.

Tabel 7. Pecahan 64 Bit dari 8 Blok Pertama Proses Xor

Blok kiri (**L**)	Blok kanan (**R**)
@ • ½	**á‰>**

Tabel 8. Pecahan 64 Bit dari 8 Blok Kedua Proses Xor

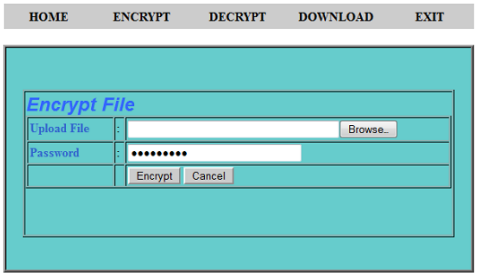
Blok kiri (**L**)	Blok kanan (**R**)
Ôô9Û	**zµŠÌ**

Dan perlu diperhatikan bahwa karakter inilah hasil akhir proses dekripsi dengan metode GOSTby Nasir yaitu berupa karakter "**á‰> @□½zµŠÌÔô9Û**" dan nilai blok ini sudah dilakukan proses pertukaran blok oleh metode GOST**.**

Dekripsi Algoritma Metode RC6

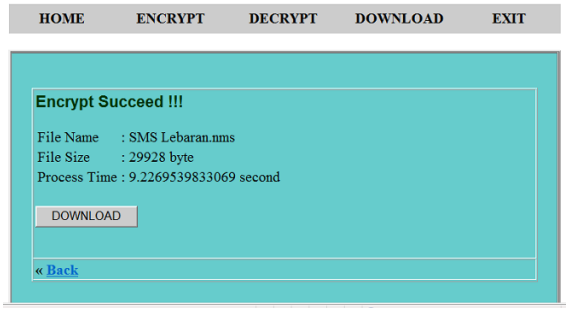
Pada metode RC6 juga berprinsip pada hal yang relatif sama, artinya pada proses dekripsi ini juga menggunakan proses balik dari enkripsinya. *Ciphertext* hasil dekrip dari metode Gost akan diambil oleh metode RC6. Blok *ciphertext* dari **gost** oleh RC6 dipecah menjadi 4 bagian blok yang berukuran 32 bit dan langsung dikonversi dalam bentuk bilangan bertipe long. Adapun tehnik membuat 4 karakter menjadi 1 buah blok bilangan sudah dibahas sebelumnya, yaitu dengan cara menggunakan pergeseran 8 bit ke kiri serta dengan menjumlahkan karakter berikutnya.

Proses upload file untuk proses enkripsi



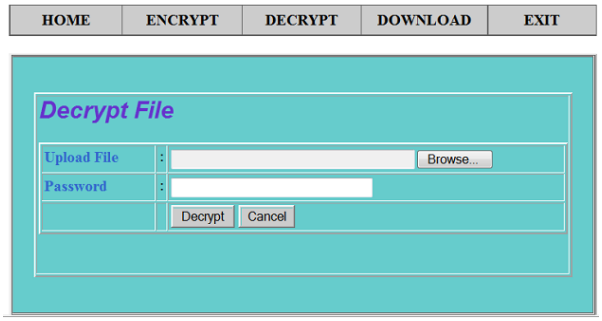
Gambar 8. File Tujuan yang di Enkripsi

Hasil Proses Enkripsi



Gambar 9. Hasil dari Enkripsi

Proses Upload File Dekripsi



Gambar 10. File Tujuan yang di Dekripsi

KESIMPULAN

Setelah dilakukan analisis permasalahan yang ada dalam pengamanan dan penanganan data yang sedang digunakan dan teknik yang diambil dari berbagai macam teknik kriptografi yang ada, maka dapat diambil kesimpulan sebagai berikut:

Kriptografi ini dapat melakukan proses pengacakan data pada semua jenis file. Penggabungan metode RC6 dengan GOST untuk pengamanan data dalam skripsi ini masih sangat sederhana, sehingga masih perlu lagi ditingkatkan dengan pembuatan algoritma yang lebih baik agar tingkat keamanan data lebih baik pula.

Enkripsi dengan metode ini dapat dijalankan dimanapun kapanpun selama tersedia sarana untuk mengakses teknologi *internet* melalui sebuah *browser*.

Program enkripsi ini dapat berjalan pada Sistem Operasi windows, dalam penulisan ini program dibuat pada komputer dengan Sistem Operasi Windows XP Profesional dan Windows 7.

Pada akhir proses enkripsi dan dekripsi akan ditampilkan nama file dan ukuran file hasil proses serta waktu yang dibutuhkan untuk proses file, tak lupa disediakan tombol untuk mendownload file hasil baik setelah proses enkripsi maupun dekripsi.

DAFTAR PUSTAKA

Alan G.Konheim. (2007). *Computer Security and Cryptograhy*. New Jersey. John Wiley & Sons, Inc

Anupam, V. Mayer, A. (1998). Secure Web Scripting, *IEEE Internet Computing*. pp. 46-55

Bruce Schneier. (1996). Section 14.1 GOST. *In Applied Cryptography*. Second Edition. John Wiley and Sons. ISBN 0-471-11709-9

Bruce Schneier. (1995). The GOST encryption algorithm. *Dr. Dobb's Journal*. Vol. 20. No.1. pp. 123-124

Dony Ariyus. (2008). *Pengantar Ilmu Kriptografi, Teori, Analisis, dan Implementasi*. Yogyakarta: Andi Offset

Fleischmann Ewan, Gorski Michael, Huehne Jan-Hendrik, Lucks Stefan. (2009). Key re-covery attack on full GOST block cipher with zero time and memory. Published as ISO/IEC JTC 1/SC 27 N8229

I. A. Zab otin, G. P. Glazkov, V. B. Isaeva. (1989). Cryptographic Protection for Information Processing Systems, Government Standard of the USSR, GOST 28147-89, Government Committee of the USSR for Standards,. In Russian, translated to English in http://www.autochthonous.org/crypto/gosthash.tar.gz

I Putu Herryawan. (2010). APLIKASI KEAMANAN DATA MENGGUNAKAN METODA KRIPTOGRAFI GOST , *Jurnal TSI*, Vol.1, No.2

Itani, M. Diab, H. (2004). Reconfigurable Computing for RC6 Cryptography, 2004 IEEE/ACS International Conference on Pervasive Services (ICPS'04), pp. 121-127

Nashirudin, Didi Juardi. (2006). *Sistem Perangkat Lunak Enkripsi Data Menggunakan Metode RC6 pada PT Matsuhita Toshiba Picture Display Indonesia*

Nawa l El-F ishawy, Osama M.Abu Zaid. (2007). Quality of Encryp tion Measurement of Bitmap Images with RC 6, MRC 6, and Rijndael Block Cipher Algorithms. *International Journal of Network Security*. Vo l.5, No 3. PP.241–251

Nicolas Courtois, MichaÃl Misztal. (2011). Aggregated Di®erentials and Cryptanalysis of PP-1 and GOST, To app ear in 11th Central Europ ean Conference on Cryptology, will be held in Debrecen, Hungary

Nicolas Courtois. (2011). Security Evaluation of GOST 28147-89 In View Of International Standardisation, do cument o±cially submitted to ISO in May 2011, In Cryptology ePrint Archive, Report 2011/211. 1 May 2011, http://eprint.iacr.org/2011/211

Nicolas Courtois, MichaÃl Misztal. 14 June 2011: Di®erential Cryptanalysis of GOST, In Cryptol-ogy ePrint Archive, Rep ort 2011/312, http://eprint.iacr.org/2011/312

Prayudi, Y.; Halik, I. (2005). *Studi Dan Analisis Algoritma Rivest Code 6 (Rc6) Dalam Enkripsi/Dekripsi Data*. Seminar Nasional Teknologi Informasi 2005 (SNATI 2005). pp. 149 – 158

Rivest, R., L.; Robshaw, M.J.B; Sidney, R.; Yin, Y.L. (1998). *The RC6 Block Cipher, RCA Laboratories*.

Rudianto. *Analisis Keamanan Algoritma Kriptografi RC6*. Jurusan Teknik Informatika ITB, Bandung

Schmeh, Klaus. (2003). *Cryptography and Public Key Infrastructure on the Internet*. Willey & Sons, inc.

Soohyun, et al. (2003). *An Efficient Hybrid Cryptosystem Providing Authentication for Sender's Identity*. Sungkyunkwan University Korea.

Takanori Isobe. (2011). A Single-Key Attack on the Full GOST Blo ck Cipher, In FSE 2011, Fast Software Ecnryption, Springer LNCS

Vasily Dolmatov. (2010). GOST 28147-89 encryption, decryption and MAC algorithms, IETF. ISSN: 2070-1721. Editor, RFC 5830, http://tools.ietf.org/html/rfc5830

Vitaly V. Shorin, Vadim V. Jelezniakov and Ernst M. Gabidulin. (2001). Linear and Dif-ferential Cryptanalysis of Russian GOST, Preprint submitted to Elsevier Preprint

Wiwit Siswoutomo. (2007). *Fundamental of PHP Security*. Jakarta: Elex Media Komputindo

BIODATA PENULIS

Didi Juardi, Ir., M.Kom lahir di Jakarta, tanggal 22 April tahun 1971. Pendidikan terakhir S2 Teknik Informatika STMIK Nusa Mandiri Jakarta.