Jurnal TIMES , Vol. IV No 2 : 14-19 , 2015
ISSN : 2337 - 3601

ANALISA PENGUJIAN ESTIMASI WAKTU DAN BESAR UKURAN FILE MENGGUNAKAN ALGORITMA TWOFISH PADA PROSES ENKRIPSI DAN DEKRIPSI

Edy Rahman Syahputra

Program Studi Sistem Informasi
Sekolah Tinggi Teknik Harapan
Jl. H.M. Joni No.70C Medan, Indonesia
Email: ydeaja@yahoo.com

Abstrak

Algoritma twofish merupakan algoritma kriptografi yang bersifat simetris dan beroperasi dalam bentuk cipher blok, jaringan fiestel, dan s-box. Berdasarkan hal tersebut, timbul suatu permasalahan algoritma twofish terhadap estimasi waktu yang diperlukan pada saat proses enkripsi dan dekripsi suatu file dan besar ukuran file pada saat proses enkripsi dan dekripsi. Pengujian dilakukan menggunakan file dokumen dengan extension *.doc, *.xls, *.ppt untuk proses enkripsi dan dekripsi. Dari hasil pengujian yang telah dilakukan didapati hasil bahwa estimasi waktu proses enkripsi dan dekripsi yang diperoleh proses dekripsi lebih cepat.

Kata kunci : Algortima Twofish, Waktu Proses, Ukuran File

1. Pendahuluan

Isi data sebuah file yang sangat penting dan tidak harus diketahui oleh pihak-pihak yang tidak diinginkan seringkali dengan mudah diketahui dan didapatkan oleh pihak-pihak yang justru akan membuat kerahasian data tersebut diketahui, Sehingga pengaman suatu data merupakan hal mutlak yang harus dilakukan saat ini. Berbagai cara dilakukan agar kerahasian data tetap dapat terjaga. Isi data dan file yang dibuat dalam bentuk sebuah dokumen, dimana menggunakan program word, excel ataupun powerpoint.

Proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan dan atau alat khusus disebut dengan enkripsi. Sedangkan dekripsi merupakan algoritma atau cara yang dapat digunakan untuk membaca informasi yang telah dienkripsi untuk kembali dapat dibaca [4].

Pada proses enkripsi dan dekripsi seringkali pengguna mengalami kendala yaitu terlalu lamanya waktu proses karena besarnya ukuran file atau tidak efektifnya algoritma yang digunakan untuk mengamankan data tersebut.

Di dalam dunia informatika dikenal algoritma Twofish, algoritma tersebut merupakan algoritma kriptografi kunci simetri kategori *cipher* blok yang banyak digunakan.

Algoritma twofish merupakan algoritma kriptografi yang bersifat simetris dan beroperasi dalam bentuk cipher blok, jaringan fiestel, dan s-box.

Berdasarkan hal tersebut, timbul suatu permasalahan algoritma *twofish* terhadap performansi waktu serta efektifitas pada saat proses enkripsi dan dekripsi suatu file. Untuk mengetahui performansi waktu serta efektifitas dari algoritma tersebut, maka dilakukan analisis kinerja.

Adapun batasan masalah yang membatasi penelitian ini adalah sebagai berikut:

1. Pembahasan hanya pada algoritma *twofish*.
2. Pembangunan aplikasi digunakan untuk menganalisis kinerja dalam hal estimasi waktu proses enkripsi dan dekripsi dan besar ukuran file sebelum dan sesudah proses enkripsi dan dekripsi dilakukan.
3. File yang akan diuji untuk keperluan analisis perbandingan adalah file dokumen *extention *.doc, *.xls, *ppt*.

Selain itu juga pada penelitian ini terdapat beberapa tujuan sebagai berikut:

1. Mengetahui estimasi waktu proses dan besar ukuran file enkripsi dan dekripsi suatu file pada algoritma *twofish*.
2. Merancang program yang dapat digunakan untuk keperluan analisis kinerja dari algoritma *twofish*.
3. Membangun program kriptografi yang mengadopsi algoritma *twofish*.

2. Tinjauan Pustaka

Algoritma Twofish merupakan 128-bit block sandi/cipher yang bisa menerima panjang varibel kunci/key sebesar 256 bit. Cipher tersebut berasal 16-round jaringan Feitsel dengan fungsi bijektif F yang dilanjutkan dengan empat key-dependent 8-by-b-bit S-boxes, satu fixed 4-by-4 maximum distance separable matrix over GF(2^8), satu pseudo-Hadamard transform, satu rotasi bitwise dan satu desain key schedule. Suatu implementasi Twofish yang dioptimalkan mengenksripsi pada Pentium Pro dengan 17,8 siklus clock per byte, dan pada smartcard akan mengenksripsi pada 1660 siklus clock per byte. Twofish dapat diimplemetsikan pada pada perangkat keras dengan 14000 gerbang. Design round function dan penjadwalan kunci mengakibatkan adanya trade

off antara kecepatan, ukuran software, waktu setup key, jumlah gerbang dan memory [2].

Twofish dapat melakukan:

1. Melakukan enkripsi data pada 285 siklus per block di atas Pentium Pro setelah menjalankan key setup 12700 siklus clock.
2. Melakukan enkripsi data pada 860 siklus per blok sdi atas Pentium Pro setelah menjalankan key setup 1250 siklus clock.
3. Melakukan enkripsi data pada 26500 siklus per block di atas sebuah 6805 smart card setelah mejalankan key setup 1750 siklus clock.

Algoritma *twofish* memiliki beberapa blok pembangun, yaitu [5] :

1. Jaringan *Feistel*, merupakan model komputasi berulang yang digunakan oleh banyak *cipher* blok. Fungsi dari model jaringan *feistel* adalah untuk memetakan suatu fungsi enkripsi yang sederhana menjadi fungsi enkripsi yang rumit dan kuat.
2. *S-Box*, merupakan matriks substitusi yang digunakan untuk memetakan *bit-bit*. *S-box* dapat bervariasi bentuk dan ukuran *input output*-nya. *Twofish* menggunakan empat *S-box* yang berbeda, dengan ukuran *8x8 bit*.
3. *Matriks MDS* (*Maximum Distance Separable*) adalah matriks transformasi dari sebuah kode linear.

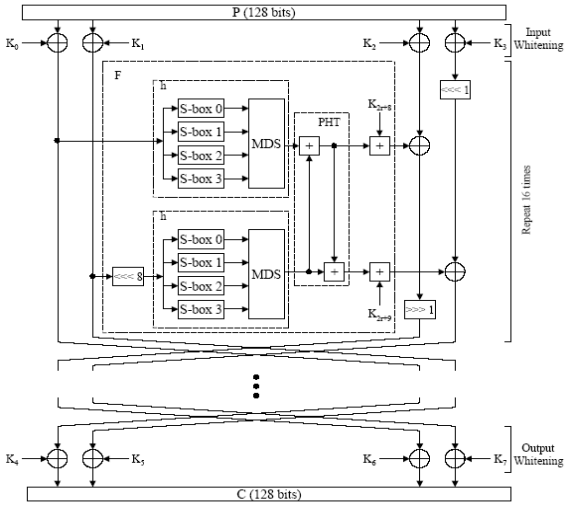
Selain blok-blok pembangunan, algoritma *twofish* terdiri dari beberapa proses, yaitu [6] :

1. Perubahan *Pseudo-Hadamard*, merupakan transformasi dua arah yang menghasilkan difusi. Difusi yang dimaksudkan disini adalah properti dari operasi *cipher* yang dikatakan aman. *Bit* masukan dari *Pseudo-Hadamard* harus memiliki panjang yang genap, karena akan dibagi menjadi dua bagian yang sama panjang, masing-masing sepanjang *n/2* yang dilambangkan dengan *a* dan *b*. Persamaan *pseudo-hadamard* adalah sebagai berikut :
   $a` = a + b \pmod{2^n}$ $b` = a + 2b \pmod{2^n}$
   untuk membalikkan persamaan di atas, persamaanya adalah :
   $b = b` - a` \pmod{2^n}$ $a = 2a` - b` \pmod{2^n}$
   dimana *n* adalah jumlah *bit* yang digunakan.
2. *Whitening*, merupakan teknik untuk meningkatkan keamanan dari *cipher* blok yang menggunakan iterasi, tujuannya adalah agar *input* dan *output* dari fungsi *F* tidak diketahui. *Whitening* dilakukan dengan cara mengubah data dengan meng-*XOR* data dengan sebagian dari kunci sebelum iterasi pertama dan setelah iterasi terakhir dari enkripsi.
3. Penjadwalan kunci adalah proses mengubah kunci menjadi beberapa subkunci yang akan digunakan pada iterasi-iterasi

Selain unsur pembangunan di atas, ada beberapa proses penunjang lain pada implementasi algoritma *twofish*, yaitu [2] :

1. *Bit* masukan sebanyak 128 *bit* akan dibagi menjadi empat bagian masing-masing sebesar 32 *bit* menggunakan konvensi *little endlan*. Dua bagian bit akan menjadi bagian kanan, dua bagian bit lainnya akan menjadi bagian kiri.
2. *Bit input* akan di *XOR* terlebih dahulu dengan empat bagian kunci, atau dengan kata lain mengalami proses *whitening*.
   $R_{0,i} = P_i \; XOR \; K_i \qquad i = 0, ..., 3$
   Dimana *K* adalah kunci, K_i berarti subkunci yang ke-*i*.
3. Fungsi *f* dari *twofish* terdiri dari beberapa tahap, yaitu :
   a. Fungsi *g*, yang terdiri dari empat *s-box* dan matriks *MDS*
   b. PHT (*Pseudo-Hadamard Transform*)
   c. Penambahan hasil PHT dengan kunci

   Blok-blok pembangunan beserta karakteristik di atas, akan digunakan pada impementasi algoritma *twofish* dengan tahapan seperti gambar dibawah ini [3].



Gambar 1. Struktur Algoritma *Twofish*

Tahapan-tahapan pada algoritma *twofish* lebih jelasnya adalah sebagai berikut [3] :

1. Bit masukan disebut sebagai P_0, P_1, P_2, dan P_3. P_0 dan P_1 akan menjadi bagian kiri, dua lainnya akan menjadi masukan pada bagian kanan.
2. Kemudian akan melalui proses *whitening*.
3. Bagian kiri akan menjadi masukan untuk fungsi *f*, P_0 akan langsung menjadi masukan bagi fungsi *g*, sementara P_1 akan di-*rotate* 8 *bit* sebelum diproses oleh fungsi *g*.
4. Didalam fungsi *g*, *bit-bit* tersebut akan melalui *S-box* dan matriks MDS, kemudian kedua keluaran akan digabungkan oleh PHT.
5. Setelah melalui PHT, kedua bagian tersebut akan ditambah dengan bagian dari kunci sesuai dengan iterasi yang telah dilewati. Untuk keluaran dari fungsi *f* dengan *input* P_1 akan ditambah dengan K_{2r+8}. Untuk keluaran dari fungsi *f* dengan *input* P_1 akan ditambah dengan $K_{2r+9,}$ dimana *r* adalah jumlah iterasi yang telah dilewati. Masing-masing

ditambah delapan dan sembilan karena delapan urutan awal sudah digunakan untuk *whitening input* dan *output*.

6. Keluaran dari fungsi *f* dengan input P_0 akan di-*XOR* dengan P_2, kemudian hasil *XOR* tersebut akan di-*rotate* 1 *bit*.
7. Keluaran dari fungsi *f* dengan input P_1 akan di-*XOR* dengan P_3, namun P_3 sebelumnya di-*rotate* 1 *bit* terlebih dahulu.
8. Setelah perhitungan *bit* selesai, bagian kanan yang telah dihitung tadi akan menjadi bagian kiri dan bagian kiri yang belum dihitung akan menjadi bagian kanan.
9. Kemudian setelah 16 iterasi, akan dilakukan *whitening* terhadap keluarannya. *Whitening* pada *output* akan meng-*undo* pertukaran bagian kanan dan bagian kiri pada iterasi terakhir, dan melakukan *XOR* data dengan 4 bagian kunci,
   $Ci = R16_{,(i+2) \bmod 4} \oplus K_{i+4} \qquad i = 0, ..., 3$
   Bagian kunci yang digunakan disini berbeda dengan bagian kunci yang akan digunakan saat *whitening* pada *input*. Oleh karena itu urutan bagian kunci yang dipakai ditambah empat, karena empat urutan bagian kunci satu sampai empat sudah terlebih dahulu digunakan untuk *whitening* pada *input*.
10. Keempat bagian cipherteks tersebut kemudian ditulis menjadi 16 *byte* $C_0, ..., C_{15}$ menggunakan konversi *little-endian* seperti pada plainteks.
    $Ci = \left\lfloor \frac{C_{\lfloor i/4 \rfloor}}{2^{8(i \bmod 4)}} \right\rfloor \bmod 2^8 \quad i = 0, ..., 15$

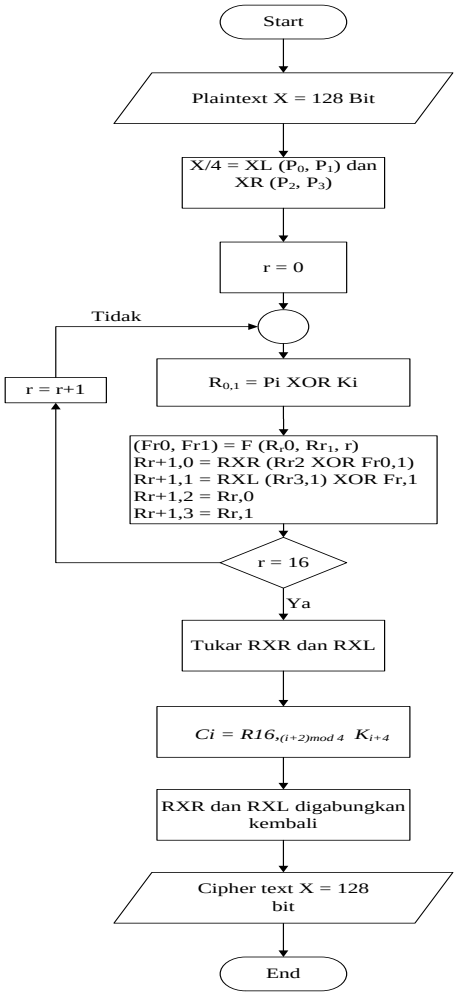
Implementasi algoritma *twofish* harus memperhatikan kecepatan komputasi yang diinginkan. *Twofish* mempunyai karakteristik melakukan persiapan kunci dalam waktu yang lama. Karena itulah untuk menjamin kecepatan, semua proses penjadwalan kunci dapat dilakukan terlebih dahulu dan disimpan. Penggunaan algoritma *twofish* antara lain terdapat pada [1] :

1. Away32 Deluxe dan Away IDS Deluxe oleh BMC Engineering.
   Kedua aplikasi di atas merupakan perangkat lunak untuk enkripsi arsip dan folder pada windows.
2. CleverCrypt oleh Quantum Digital Security.
   Perangkat lunak enkripsi drive virtual on-the-fly untuk windows. Selain menggunakan algoritma *twofish*, aplikasi ini juga menggunakan rijndael, dan *blowfish*.
3. Cryptcat oleh Farm9
   Versi aplikasi netcut buatan L0pht dengan algoritma *twofish*. aplikasi ini memungkinkan pembangunan tunnel sederhana yang terenkripsi antar mesin, melalui jaringan internet, dan dalam kasus-kasus tertentu dapat melewati firewalls.
4. DigiSecret oleh TamoSoft.
   Aplikasi berbasiskan windows untuk membuat archive yang terenkripsi dan arsip self extracting exe, shredding, dan file sharing melalui internet.
5. FoxTrot oleh Roth Systems
   Sebuah server HTTP yang dirancang sebagai server aplikasi profesional. Dengan menggunakan aplikasi ini, pengguna dapat mengeksekusi perintah SQL langsung melalui address line pada browser.

3. Metode Penelitian

Tahapan dalam analisis pengujian estimasi waktu pada algoritma twofish digambarkan dalam bentuk flowchart sebagai berikut:

1) Flowchart proses enkripsi



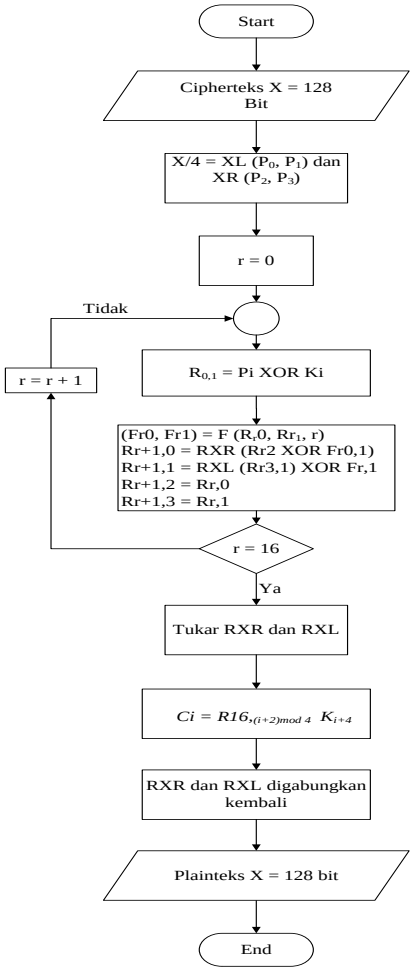
Gambar 2. Flowchart Proses Enkripsi Algoritma Twofish

Pada gambar 2. merupakan *flowchart* proses dekripsi pada algoritma *twofish*. Proses dekripsi algoritma *twofish* yang terjadi, yaitu sebagai berikut

1. Memulai proses enkripsi (*plaintext*) dengan *X* = 128 *bit*
2. *X* dibagi menjadi 4 bagian. *XL* = P_0, P_1 dan *XR* = P_2, P_3
3. *Input whitening* ke-empat bagian tersebut di-*XOR* dengan kunci yang telah diekspansi
4. Melakukan perulangan hingga 16 kali putaran (*r* = *r*+1), pada setiap putaran P_0 dan P_1 sebagai masukan dari fungsi *F*, P_2 dilakukan operasi *XOR* dan dirotasikan ke kanan sebanyak 1 *bit*, P_3

dirotasikan ke kanan 1 *bit* dan dilakukan rotasi *XOR* pada keluaran fungsi *F*
5. Menukarkan hasil *RXR* dan *RXL*
6. *Output whitening* hasil keluaran dan melakukan operasi *XOR* dengan 4 buah kata dari kunci yang diekspansi
7. Menggabungkan hasil *RXR* dan *RXL*.
8. Menghasilkan *cipher text X*
9. Selesai

2) Flowchart Proses Dekripsi



Gambar 3. Flowchart Proses Enkripsi Algoritma Twofish

Pada gambar 3. merupakan *flowchart* proses dekripsi pada algoritma *twofish*. Proses dekripsi algoritma *twofish* yang terjadi, yaitu sebagai berikut :

1. Memulai proses dekripsi (*cipher text*) dengan *X* = 128 *bit*.
2. *X* dibagi menjadi 4 bagian. *XL* = P_0, P_1 dan *XR* = P_2, P_3
3. *Input whitening* ke-empat bagian tersebut di-*XOR* dengan kunci yang telah diekspansi
4. Melakukan perulangan hingga 16 kali putaran di mulai dengan *i* = 0, pada setiap putaran P_0 dan P_1 sebagai masukan dari fungsi *F*, P_2 dilakukan operasi *XOR* dan dirotasikan ke kanan sebanyak 1 *bit*, P_3 dirotasikan ke kanan 1 *bit* dan dilakukan rotasi *XOR* pada keluaran fungsi *F*
5. Menukarkan hasil *RXR* dan *RXL*
6. *Output whitening* hasil keluaran dan melakukan operasi *XOR* dengan 4 buah kata dari kunci yang diekspansi
7. Menggabungkan hasil *RXR* dan *RXL*.
8. Menghasilkan *plaintext X*
9. Selesai

4. Implementasi dan Hasil

Pelaksanaan pengujian dilakukan berdasarkan skenario pengujian yang telah ditetapkan sebelumnya. Adapun proses enkripsi dan dekripsi file yang dilakukan, yaitu sebagai berikut :

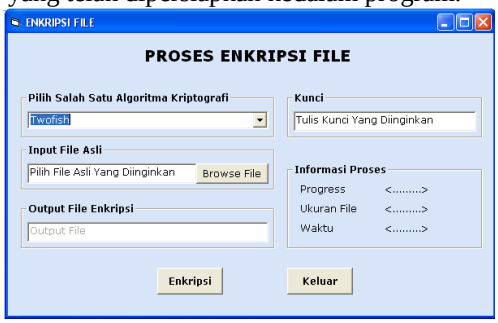
Mempersiapkan file-file yang akan diuji, yaitu file asli (file sebelum dienkripsi) dokumen dengan *extention* **.doc*, **.xls*, **.ppt*, masing-masing terdiri dari 5 file, sehingga jumlah file yang akan diuji adalah 15 file, seperti tampak pada tabel.

Tabel 1. File Dokumen Yang Akan Diuji

No	Nama File	Jenis File	Ukuran File (Bytes)
1.	Test1(1).doc	File Word	1.074.688
2.	Test1(2).doc	File Word	529.408
3.	Test1(3).doc	File Word	32.256
4.	Test1(4).doc	File Word	660.992
5.	Test1(5).doc	File Word	867.840
6.	Test2(1).xls	File Excel	34.304
7.	Test2(2).xls	File Excel	45.056
8.	Test2(3).xls	File Excel	39.936
9.	Test2(4).xls	File Excel	36.352
10.	Test2(5).xls	File Excel	41.472
11.	Test3(1).ppt	File Power Point	1.413.120
12.	Test3(2).ppt	File Power Point	861.696
13.	Test3(3).ppt	File Power Point	808.960
14.	Test3(4).ppt	File Power Point	751.616
15.	Test3(5).ppt	File Power Point	582.144

1) Proses Enkripsi

Membuka menu enkripsi file dan input file-file yang telah dipersiapkan kedalam program.



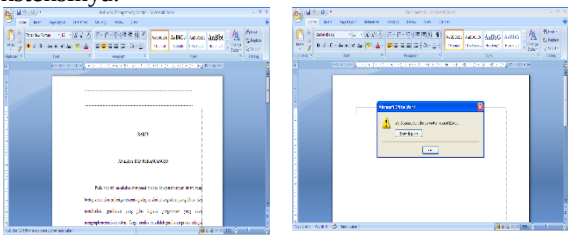
Gambar 4. Menu Enkripsi File

Setelah file diinputkan dan proses enkripsi selesai didapatkan hasil sebagai berikut:

Tabel 2. Hasil Proses Enkripsi

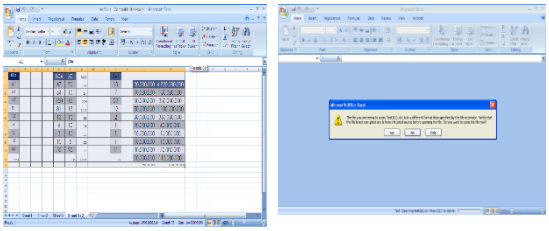
No	File Asli		File Terenkripsi		Waktu Proses (ms)
	Nama File	Size (Bytes)	Nama File	Size (Bytes)	
1.	Test1(1).doc	1,074,688	Test1(1).doc	1.074.704	31,31162
2.	Test1(2).doc	529.408	Test1(2).doc	529.424	13,62425
3.	Test1(3).doc	32.256	Test1(3).doc	32.272	0,952625
4.	Test1(4).doc	660.992	Test1(4).doc	661.008	17,60913
5.	Test1(5).doc	867.840	Test1(5).doc	867.856	22,48388
6.	Test2(1).xls	34.304	Test2(1).xls	34.320	1,06225
7.	Test2(2).xls	45.056	Test2(2).xls	45.072	1,281125
8.	Test2(3).xls	39.936	Test2(3).xls	39.952	1,156125
9.	Test2(4).xls	36.352	Test2(4).xls	36.368	1,124875
10	Test2(5).xls	41.472	Test2(5).xls	41.488	1,04675
11	Test3(1).ppt	1.413.120	Test4(1).ppt	1.413.136	36,437
12	Test3(2).ppt	861.696	Test4(2).ppt	861.712	22,311 75
13	Test3(3).ppt	808.960	Test4(3).ppt	808.976	20,921 25
14	Test3(4).ppt	751.616	Test4(4).ppt	751.632	19,421 63
15	Test3(5).ppt	582.144	Test4(5).ppt	582.160	15,155 88

Berikut adalah bentuk file yang telah terenkripsi berdasarkan jenis file yang berbeda ekstensinya.



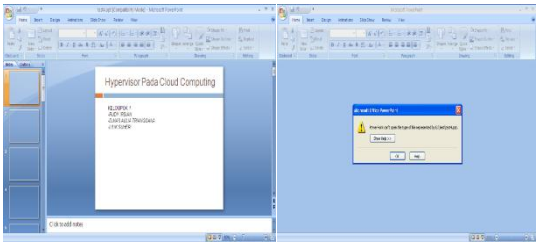
File Asli File Terenkripsi

Gambar 5. Proses Enkripsi Algoritma *Twofish* Pada File Test1(1).doc



File Asli File Terenkripsi

Gambar 6. Proses Enkripsi Algoritma *Twofish* Pada File Test2(1).xls



File Asli File Terenkripsi

Gambar 7. Proses Enkripsi Algoritma *Twofish* Pada File Test3(1).ppt

2) Proses Dekripsi

Membuka menu dekripsi file dan input file-file yang telah dienkripsi sebelumnya.

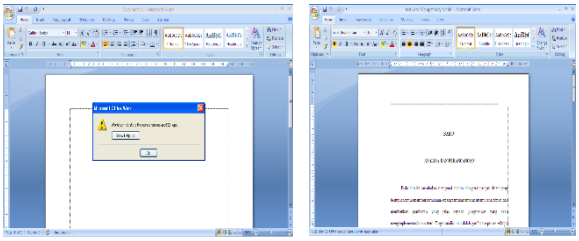
Gambar 8. Menu Dekripsi File

Setelah file diinputkan dan proses enkripsi selesai didapatkan hasil sebagai berikut:

Tabel 3. Hasil Proses Dekripsi

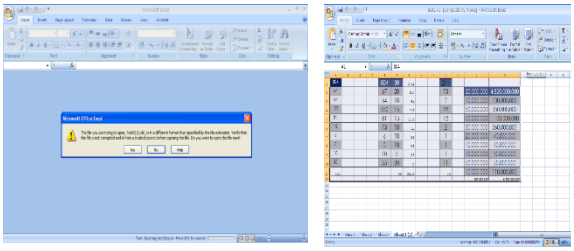
No	File Terenkripsi		File Asli (Hasil Dekripsi)		Waktu Proses (ms)
	Nama File	Size (Bytes)	Nama File	Size (Bytes)	
1.	Test1(1).doc	1.074.704	Test1(1).doc	1,074,688	27,34287
2.	Test1(2).doc	529.424	Test1(2).doc	529.408	13,29663
3.	Test1(3).doc	32.272	Test1(3).doc	32.256	0,796375
4.	Test1(4).doc	661.008	Test1(4).doc	660.992	16,76563
5.	Test1(5).doc	867.856	Test1(5).doc	867.840	22,65575
6.	Test2(1).xls	34.320	Test2(1).xls	34.304	1,014875
7.	Test2(2).xls	45.072	Test2(2).xls	45.056	1,264875
8.	Test2(3).xls	39.952	Test2(3).xls	39.936	0,983625
9.	Test2(4).xls	36.368	Test2(4).xls	36.352	1,062
10	Test2(5).xls	41.488	Test2(5).xls	41.472	1,14
11	Test3(1).ppt	1.413.136	Test4(1).ppt	1.413.120	36,01537
12	Test3(2).ppt	861.712	Test4(2).ppt	861.696	21,828
13	Test3(3).ppt	808.976	Test4(3).ppt	808.960	20,389 75
14	Test3(4).ppt	751.632	Test4(4).ppt	751.616	18,968 12
15	Test3(5).ppt	582.160	Test4(5).ppt	582.144	14,718 25

Berikut adalah bentuk file yang telah terdekripsi berdasarkan jenis file yang berbeda ekstensinya.



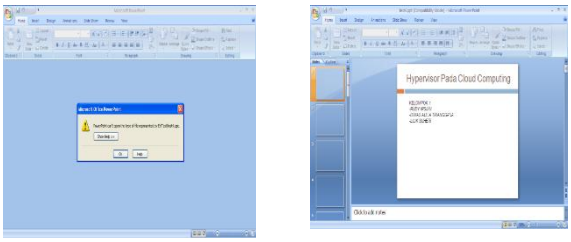
File Terenkripsi File Asli (Hasil Dekripsi)

Gambar 9. Proses Dekripsi Algoritma *Twofish* Pada File Test1.doc



File Terenkripsi File Asli (Hasil Dekripsi)

Gambar 10. Proses Dekripsi Algoritma *Twofish* Pada File Test2.xls

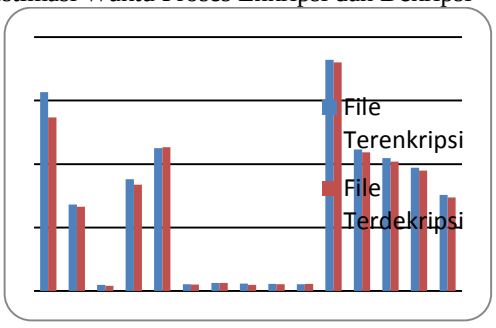


File Terenkripsi File Asli (Hasil Dekripsi)

Gambar 11. Proses Dekripsi Algoritma *Twofish* Pada File Test3.ppt

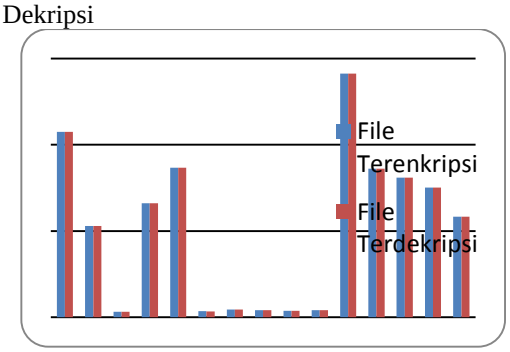
Dari pengujian yang telah dilakukan, didapatkan hasil yang dibentuk dalam grafik berikut ini:

1. Estimasi Waktu Proses Enkripsi dan Dekripsi



Gambar 12. Grafik Estimasi Waktu Proses Enkripsi Dan Dekripsi

2. Besar Ukuran File Pada Saat Enkripsi dan Dekripsi



Gambar 13. Grafik Besar Ukuran File Proses Enkripsi Dan Dekripsi

5. Kesimpulan

Dari hasil pengujian yang telah dilakukan maka didapatkan kesimpulan sebagai berikut:

1. Estimasi waktu proses enkripsi dan dekripsi pada algoritma *twofish*, yang tercepat dalam proses adalah dekripsi lebih cepat.
2. Besar ukuran file dokumen dengan extention *.doc, *.xls, *.ppt sebelum dan sesudah proses enkripsi maupun dekripsi dilakukan pada algoritma *twofish* memiliki besar ukuran yang sama.

6. Daftar Pustaka

[1] Gehlot P, Biradar S.R, Sigh B.P, (2013). "Implemation of Modified Twofish Algorithm Using 128 and 192-bit Keys on VHDL". International Jornal of Computer Applications (0975-8887). Vol.70.No.13.

[2] Hendra A. (2010). "Analisis Perbandingan Kinerja Algoritma Twofish Dan TEA (Tiny Encryption Algorithm) Pada Data Suara". JIMT. Vol.7.No.1.27-34.

[3] Nathasia D.N and Wicaksono E.A. (2011). "Penerapan Teknik Kriptografi Stream Cipher Untuk Pengamanan Basis Data" Jurnal Basis Data, ICT Research Center UNAS. Vol.6 No.1.ISSN:1978-9483.

[4] Primartha R.(2011). "Penerapan Enkripsi dan Deskripsi File Menggunakan Algoritma Data Encryption Standard (DES)". Jurnal Sistem Informasi (JSI) Vol.3 No.2. ISSN: 2085-1588. Hal 371-378.

[5] Setiawan, W. (2010). "Analisa dan Perbandingan Algoritma Twofish dan Rijndael". Makalah IF3058. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika. Institut Teknologi Bandung.

[6] Soleh, M.Y. (2010) "Studi Perbandingan Algoritma Kunci-Simetris Serpent dan Twofish". Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.