ISSN 2355 - 5920
Volume II Nomor 1
Februari 2015

jurnal PSEUDOCODE

TEKNIK INFORMATIKA
UNIVERSITAS BENGKULU
www.ejournal.unib.ac.id

Jurnal Pseudocode

JURNAL PSEUDOCODE
TEKNIK
INFORMATIKA
Volume II Nomor 1 Februari 2015

DAFTAR ISI

PENGANTAR REDAKSI

Seiring dengan perkembangan sains komputer dan teknologi informasi yang demikian pesat, dan sebagai pewujudan berkontribusi bagi Indonesia yang lebih baik. Jurnal *Pseudocode* terbitan Program Studi Teknik Informatika Universitas Bengkulu hadir sebagai salah satu media penulisan karya ilmiah yang diharapkan tidak hanya dapat menjadi sebuah referensi akademik, melainkan pula menghadirkan ide strategi pemecahan masalah bagi para pembuat kebijakan, masyarakat, dan kaum akademisi tentunya dengan menggunakan perspektif komputer sains.

Pada Februari 2015 ini, Jurnal *Pseudocode* telah menginjak usia 1 tahun. Telah banyak yang kami coba upayakan diantaranya: seleksi konten yang lebih selektif dari artikel – artikel yang masuk dari berbagai provinsi di Indonesia, mendaftarkan semua *paper* Jurnal *Pseudocode* mulai dari edisi tahun pertama agar terindeks di *Google Scholar*, hingga yang tengah dalam upaya kami adalah mendaftarkan *paper* di portal Garuda Dikti.

Pada edisi Volume II Nomor 1 ini, *Pseudocode* menyajikan delapan naskah baik yang sifatnya terapan maupun teoretikal yaitu di bidang *Wireless / Mobile Computing*, Pengolahan Citra Digital, jaringan, Sistem Cerdas, Kriptografi, dan Sistem Informasi. Bersama ini tak lupa kami ingin mengucapkan terima kasih kepada para pengirim naskah, citra bestari, serta semua pihak yang telah berperan dalam penyusunan jurnal ilmiah edisi ini.

Terakhir, redaksi kembali mengundang dan memberi kesempatan kepada para peneliti dibidang rumpun ilmu informatika untuk mempublikasikan hasil-hasil penelitiannya melalui jurnal ilmiah ini. Semoga keberadaan jurnal ilmiah *Pseudocode* mampu memberikan kontribusi pemikiran, yaitu berupa solusi optimal terhadap permasalahan ditinjau dari bidang TI.

Bengkulu, Februari 2015

Redaksi

ANALISIS PERBANDINGAN KINERJA ALGORITMA *BLOWFISH* DAN ALGORITMA *TWOFISH* PADA PROSES ENKRIPSI DAN DEKRIPSI

Dimas Aulia Trianggana[1], Herlina Latipa Sari[2]

[1,2]Program Studi Teknik Informatika – S1, Fakultas Ilmu Komputer
Universitas Dehasen Bengkulu

[1]ezdroez@gmail.com
[2]herlinalatipasari@ymail.com

Abstrak: Algoritma *Blowfish* dan algoritma *Twofish* merupakan dua buah algoritma kriptografi yang bersifat simetris dan beroperasi dalam bentuk blok cipher, jaringan fiestel, dan s-box. Berdasarkan hal tersebut, timbul suatu permasalahan antara algoritma *Blowfish* dan *Twofish* terhadap estimasi waktu yang diperlukan pada saat proses enkripsi dan dekripsi suatu file. Penelitian ini bertujuan untuk menganalisis kedua algoritma tersebut, dan mengujinya dengan cara membuat sebuah aplikasi berbasis windows yang dapat digunakan untuk mengukur estimasi waktu proses dan besar file setelah proses enkripsi dan dekripsi, sehingga dapat dilakukan perbandingan antara kedua algoritma tersebut. Pengujian dilakukan menggunakan file dokumen dengan extension *.doc, *.xls, *.psd, *.ppt baik untuk proses enkripsi maupun dekripsi. Dari hasil pengujian yang telah dilakukan diperoleh hasil bahwa jika ditinjau dari estimasi waktu proses enkripsi dan dekripsi, algoritma *Blowfish* lebih cepat waktu eksekusinya dibandingkan dengan algoritma *Twofish*, dan jika ditinjau dari besar ukuran file sebelum dan sesudah proses enkripsi dan dekripsi, algoritma *Blowfish* dan algoritma *Twofish* memiliki besar ukuran yang sama.
Kata Kunci : *Kriptografi, Algoritma Blowfish, Algoritma Twofish.*

Abstract: ***Blowfish algorithm and Twofish algorithm are two symmetrical cryptographic algorithms. They operate block ciphers, fiestel network, and s-box. Based on this, we need to estimate the time required during the process of encryption and decryption of a file. This research aims to analyze both of the algorithm by making a window based application that can measure the time and size of files after encryption and decryption process. The system was tested using files with the following extensions * .doc, * .xls, * .psd, * .ppt. Experiments results that Blowfish algorithm is faster than Twofish, with similar in size between Twofish and Blowfish both before and after the process of encryption and decryption.***
Keywords: Cryptography, Blowfish Algorithm, Twofish algorithm.

I. PENDAHULUAN

1.1 Latar Belakang

Kriptografi (*Cryptography*) didefinisikan sebagai ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) yang mempunyai pengertian, dengan cara menyamarkannya (mengacak) menjadi bentuk yang tidak dapa dimengerti menggunakan suatu algoritma tertentu.

Algoritma kriptografi terbagi dalam algoritma klasik dan modern. Contoh dari algoritma klasik adalah *Caesar cipher*, sedangkan contoh dari algoritma modern adalah algoritma *Blowfish* dan *Twofish*. Algoritma *Blowfish* dan algoritma *Twofish* telah banyak diterapkan untuk keamanan data, yakni pada enkripsi pesan suara, proses *e-banking*, komunikasi jaringan seluler, dan lain-lain. Kedua algoritma tersebut memiliki beberapa kesamaan, yaitu sama-sama menggunakan kunci simetris, *cipher* blok, jaringan *fiestel*, dan *s-box*.

Berdasarkan hal tersebut, timbul suatu permasalahan antara algoritma *Blowfish* dan *Twofish* terhadap pengukuran performansi waktu serta efektifitas (ukuran *file*) sebagai hasil proses enkripsi dan dekripsi suatu *file*.

1.2 Rumusan Masalah

1. Bagaimana membangun aplikasi kriptografi untuk membandingkan kinerja algoritma *Blowfish* dan algoritma *Twofish* ?
2. Bagaimana langkah-langkah dalam proses enkripsi dan dekripsi pada algoritma *Blowfish* dan algoritma *Twofish* ?
3. Bagaimana mengananalisis perbandingan kinerja dari algoritma *Blowfish* dan algoritma *Twofish* dalam proses enkripsi dan dekripsi ?

1.3 Batasan Masalah

1. Membahas hanya mengenai algoritma *Blowfish* dan algoritma *Twofish* pada proses enkripsi dan dekripsi.
2. Pembangunan aplikasi digunakan untuk menganalisis perbandingan kinerja dalam hal estimasi waktu proses enkripsi dan dekripsi, besar ukuran file sebelum dan sesudah proses enkripsi dan dekripsi dilakukan, serta kecepatan proses enkripsi dan dekripsi.
3. File yang akan diuji untuk keperluan analisis perbandingan adalah file dokumen *extention* *.*doc*, *.*xls*, *.*psd*, *.*ppt*. Sistem ini dibangun menggunakan Visual Basic 6.0 dan berbasis *Windows*.

II. LANDASAN TEORI

2.1. Algoritma *Blowfish*

Blowfish merupakan enkripsi yang menggunakan algoritma simetris yang tergolong ke dalam algoritma cipher blok. *Blowfish* dirancang untuk memenuhi kriteria sebagai berikut :

1. Cepat, pada implementasi yang optimal *Blowfish* dapat mencapai kecepatan *26 clock cycle/byte*
2. Kompak, *Blowfish* dapat berjalan pada memori kurang dari 5 KB
3. Sederhana, *Blowfish* hanya menggunakan operasi yang sederhana, yaitu penambahan (*addition*), *XOR*, dan penelusuran tabel (*table lookup*) pada operand 32 *bit*
4. Keamanan yang variabel, panjang kunci *Blowfish* dapat bervariasi dan dapat mencapai 448 *bit* (56 *Byte*)

Algoritma *Blowfish* menggunakan kunci yang sama untuk proses enkripsi dan dekripsi data dengan membagi pesan ke dalam blok-blok dengan ukuran yang sama panjang [5].

Blowfish termasuk dalam enkripsi *block cipher* 64 *bit* dengan panjang kunci antara 32 *bit* sampai 448 *bit*. Algoritma *Blowfish* terdiri atas dua bagian, yaitu [6]:

1. *Key-Expansion*

   Berfungsi merubah kunci (Minimum 32-*bit*, Maksimum 448-*bit*) menjadi beberapa *array* subkunci (*subkey*) dengan total 4168 *byte*.
2. Enkripsi Data

   Terdiri dari iterasi fungsi sederhana (*feistel Network*) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci-*dependent* dan substitusi kunci- dan data-*dependent*. Semua operasi adalah penambahan (*addition*) dan *XOR* pada variabel 32-*bit*. Operasi tambahan lainnya hanyalah empat penelusuran tabel (*table lookup*) *array* berindeks untuk setiap putaran

Pada algoritma *Blowfish*, digunakan banyak *subkey*. Kunci-kunci ini harus dihitung atau dibangkitkan terlebih dahulu sebelum dilakukan enkripsi atau dekripsi data. Pada jaringan *feistel*, *Blowfish* memiliki 16 iterasi, masukannya adalah 64 *bit* elemen data atau sebut saja "*X*"[7].

Adapun alur algoritma enkripsi dengan metoda *Blowfish* dijelaskan sebagai berikut [5] :

1. Bentuk inisial *P-array* sebanyak 18 buah $(P_0, P_1, \ldots\ldots\ldots\ldots P_{17})$ masing-masing bernilai

32-*bit*. *Array P* terdiri dari delapan belas kunci 32-*bit* subkunci $P_0, P_1, \ldots\ldots, P_{17}$

2. Bentuk *S-box* sebanyak 4 buah masing-masing bernilai 32-*bit* yang memiliki masukan 256. Empat 32-*bit S-box* masing-masing mempunyai 256 masukan:

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

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

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

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

3. *Plaintext* yang akan dienkripsi diasumsikan sebagai masukan, *Plaintext* tersebut diambil sebanyak 64-*bit*, dan apabila kurang dari 64-*bit* maka kita tambahkan *bit*-nya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-*bit* pertama disebut *XL*, 32-*bit* yang kedua disebut *XR*.
5. Selanjutnya lakukan operasi $XL = XL \; xor \; Pi$ dan $XR = F(XL) \; xor \; XR$
6. Hasil dari operrasi diatas ditukar *XL* menjadi *XR* dan *XR* menjadi *XL*.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran *XL* dan *XR*.
8. Pada proses ke-17 lakukan operasi untuk $XR = XR \; xor \; P_{16}$ dan $XL = XL \; xor \; P_{17}$.
9. Proses terakhir, satukan kembali *XL* dan *XR* sehingga menjadi 64-*bit* kembali.

2.2. Algoritma *Twofish*

Twofish adalah algoritma kriptografi yang beroperasi dalam mode *block cipher*. *Twofish* menjadi salah satu finalis dalam kompetisi *Advanced Encryption* Standar (AES) yang diadakan oleh *National Institute of* Standars *and Technology* (NIST). *Twofish* adalah *block cipher* yang berukuran 128 *bit* yang dapat menerima kunci dengan panjang mencapai 256 *bit*.

Tahapan-tahapan pada algoritma *Twofish* lebih jelasnya adalah sebagai berikut :

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,

   $Ci = R16_{,(i+2) mod 4} \oplus K_{i+4} \;\; i = 0, \ldots, 3$

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.

$$Ci = \left\lfloor \frac{C_{\lfloor i/4 \rfloor}}{2^{8(i \bmod 4)}} \right\rfloor \bmod 2^8 \quad i = 0, \ldots, 15$$

III. METODE PENELITIAN

3.1. Kerangka Kerja

1. Identifikasi Masalah
2. Studi Pustaka
3. Analisis Sistem

   Pada tahapan ini dilakukan analisis terhadap kebutuhan sistem, serta menganalisis elemen-elemen yang dibutuhkan oleh sistem. Pada tahap ini dilakukan studi terhadap sistem kerja algoritma *Blowfish* dan *Twofish* yang didapatkan.
4. Perancangan

   Pada tahapan ini dilakukan perancangan tampilan program yang akan dibuat penulis. Tampilan program bersifat *user-friendly* atau mudah digunakan oleh pengguna.
5. Implementasi

   Tahapan ini dilakukan untuk mengimplementasikan hasil rancangan dan analisis di atas. Pada tahapan ini dilakukan pembuatan program, pembuatan antarmuka masukan dan keluaran, dan antarmuka proses enkripsi dan dekripsi algoritma *Blowfish* dan algoritma *Twofish*.
6. Pengujian

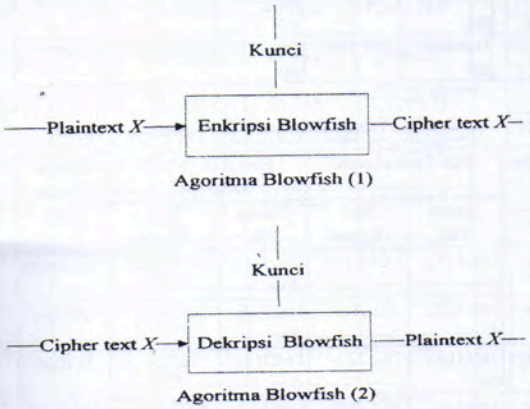
   Pada tahap ini penulis melakukan pengujian terhadap algoritma *Blowfish* dan *Twofish* dengan menggunakan aplikasi kriptografi. Adapun mekanisme pengujiannya yaitu :

   a. Mempersiapkan file-file yang akan diuji, yaitu file dokumen dengan *extention* *.*doc*, *.*xls*, *.*psd*, *.*ppt*.
   b. Menetapkan kunci untuk proses enkripsi dan dekripsi. Karena algoritma *Blowfish* dan *Twofish* merupakan kunci simetris, maka kunci yang digunakan adalah kunci yang sama dalam proses enkripsi dan dekripsi
   c. Melakukan proses enkripsi setiap file asli yang akan diuji dengan kunci yang telah ditetapkan. Kemudian mencatat waktu proses enkripsi, dan kapasitas file sebelum dan sesudah proses enkripsi dilakukan.
   d. Melakukan proses dekripsi setiap file hasil enkripsi dengan kunci yang telah ditetapkan (kunci yang sama dengan kunci yang digunakan dalam proses enkripsi). Kemudian mencatat waktu proses dekripsi, dan kapasitas file sebelum dan sesudah proses dekripsi dilakukan, menghitung kecepatan proses enkripsi dan dekripsi.
   e. Hasil proses enkripsi dan dekripsi yang telah dicatat tadi kemudian di analisis kembali untuk memberikan perbandingan kinerja antara algoritma *Blowfish* dan algoritma *Twofish*, sehingga nantinya diperoleh kesimpulan tentang kelebihan dan kekurangan antara kedua algoritma tersebut.
7. Dokumentasi

Dilakukan dokumentasi laporan mulai dari identifikasi masalah hingga implementasi dan pengujian.

3.2. Skema Proses Enkripsi dan Dekripsi Algoritma *Blowfish*

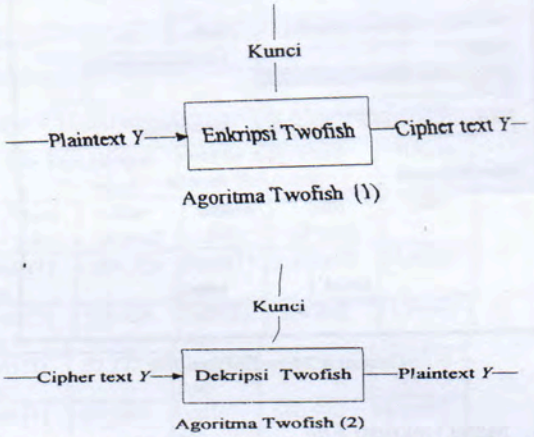
Algoritma *Blowfish* menggunakan kunci *simetris dalam proses enkripsi dan dekripsi.* Misalkan kunci yang digunakan untuk proses enkripsi adalah *123*. Maka untuk melakukan dekripsi harus menggunakan kunci yang sama yaitu *123*, agar didapati hasil yang sama sebelum dienkripsi. Adapun skema proses enkripsi dan dekripsi pada algoritma *Blowfish*, dapat di lihat pada gambar 1.



Gambar 1. Skema Proses Enkripsi dan Dekripsi Algoritma *Blowfish*

3.3. Skema Proses Enkripsi dan Dekripsi Algoritma *Twofish*

Algoritma *Twofish* menggunakan kunci simetris dalam proses enkripsi dan dekripsi. Misalkan kunci yang digunakan untuk proses enkripsi adalah *123*. Maka untuk melakukan dekripsi harus menggunakan kunci yang sama yaitu *123*, agar didapati hasil yang sama sebelum dienkripsi. Adapun skema proses enkripsi dan dekripsi pada algoritma *Twofish*, dapat di lihat pada gambar 2.



Gambar 2. Skema Proses Enkripsi dan Dekripsi Algoritma *Twofish*

IV. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Antarmuka perangkat lunak dibuat menggunakan IDE Visual Basic 6.0, yang memiliki kelas-kelas dan modul untuk membuat *Graphical User Interface* (GUI). Antarmuka perangkat lunak kriptografi terbagi menjadi 4 (empat) menu, yaitu sebagai berikut :

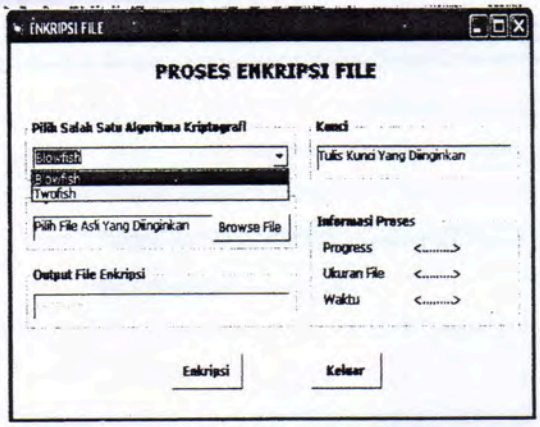
1. Menu Utama

Menu utama merupakan menu yang pertama kali tampil ketika program dijalankan oleh pengguna.

Gambar 3. Menu Utama

2. Menu Enkripsi File

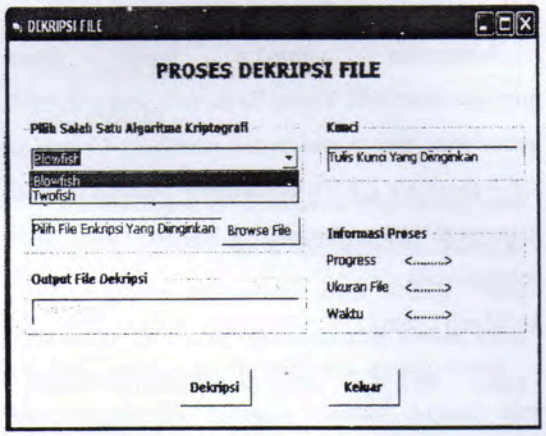
Menu enkripsi file merupakan menu yang akan digunakan untuk melakukan proses enkripsi file pada algoritma *Blowfish* dan algoritma *Twofish*.



Gambar 4. Menu Enkripsi *File*

3. Menu Dekripsi *File*

Menu dekripsi *file* merupakan menu yang akan digunakan untuk melakukan proses dekripsi file pada algoritma *Blowfish* dan algoritma *Twofish.*



Gambar 5. Menu Dekripsi *File*

4.2. Pengujian Sistem

Pada tahap ini, penulis melakukan pengujian perangkat lunak menggunakan algoritma *Blowfish* dalam proses enkripsi dan dekripsi *file* yang telah disiapkan pada skenario pengujian.

Tabel 1. Hasil Proses Enkripsi File Algoritma *Blowfish*

No	File Asli		File Hasil Enkripsi		Waktu Proses (ms)
	Nama File	Size (Bytes)	Nama File	Size (Bytes)	
1.	Test1(1).doc	1,074,688	Test1(1).doc	1.074.704	24,03087
2.	Test1(2).doc	529.408	Test1(2).doc	529.424	11,45275
3.	Test1(3).doc	32.256	Test1(3).doc	32.272	0,82725
4.	Test1(4).doc	660.992	Test1(4).doc	661.008	13,84363
5.	Test1(5).doc	867.840	Test1(5).doc	867.856	17,8745
6.	Test2(1).xls	34.304	Test2(1).xls	34.320	0,655875
7.	Test2(2).xls	45.056	Test2(2).xls	45.072	0,88975
8.	Test2(3).xls	39.936	Test2(3).xls	39.952	0,952375
9.	Test2(4).xls	36.352	Test2(4).xls	36.368	0,749875
10	Test2(5).xls	41.472	Test2(5).xls	41.488	1,03075
11	Test3(1).psd	464.484	Test3(1).psd	464.496	9,140625
12	Test3(2).psd	456.135	Test3(2).psd	456.152	8,811625
13	Test3(3).psd	611.491	Test3(3).psd	611.504	11,9375
14	Test3(4).psd	874.467	Test3(4).psd	874.480	17,1555
15	Test3(5).psd	2.964.970	Test3(5).psd	2.964.984	58,95275
16	Test4(1).ppt	1.413.120	Test4(1).ppt	1.413.136	29,65537
17	Test4(2).ppt	861.696	Test4(2).ppt	861.712	16,76513
18	Test4(3).ppt	808.960	Test4(3).ppt	808.976	15,65588
19	Test4(4).ppt	751.616	Test4(4).ppt	751.632	14,76563
20	Test4(5).ppt	582.144	Test4(5).ppt	582.160	11,4835

Tabel 2. Hasil Proses Dekripsi File Algoritma *Blowfish*

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	22,48388
2.	Test1(2).doc	529.424	Test1(2).doc	529.408	10,484
3.	Test1(3).doc	32.272	Test1(3).doc	32.256	0,7655
4.	Test1(4).doc	661.008	Test1(4).doc	660.992	12,9835
5.	Test1(5).doc	867.856	Test1(5).doc	867.840	16,89
6.	Test2(1).xls	34.320	Test2(1).xls	34.304	0,79625
7.	Test2(2).xls	45.072	Test2(2).xls	45.056	1,0155
8.	Test2(3).xls	39.952	Test2(3).xls	39.936	0,905625
9.	Test2(4).xls	36.368	Test2(4).xls	36.352	0,859125
10	Test2(5).xls	41.488	Test2(5).xls	41.472	0,95225
11	Test3(1).psd	464.496	Test3(1).psd	464.484	9,202375
12	Test3(2).psd	456.152	Test3(2).psd	456.135	8,921375
13	Test3(3).psd	611.504	Test3(3).psd	611.491	12,0935
14	Test3(4).psd	874.480	Test3(4).psd	874.467	17,26563
15	Test3(5).psd	2.964.984	Test3(5).psd	2.964.970	57,82763

16	Test4(1).ppt	1.413.136	Test4(1).ppt	1.413.120	28,06225
17	Test4(2).ppt	861.712	Test4(2).ppt	861.696	17,04688
18	Test4(3).ppt	808.976	Test4(3).ppt	808.960	15,93713
19	Test4(4).ppt	751.632	Test4(4).ppt	751.616	14,68713
20	Test4(5).ppt	582.160	Test4(5).ppt	582.144	11,37412

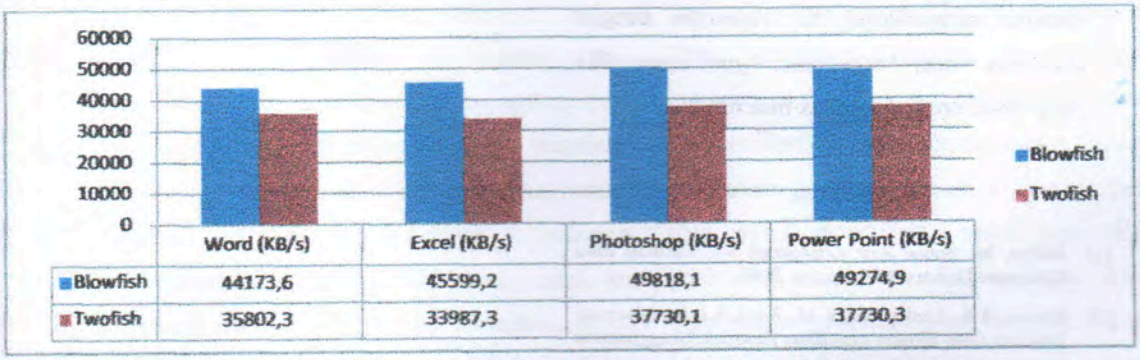
Tabel 3. Hasil Proses Enkripsi File Algoritma *Twofish*

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).psd	464.484	Test3(1).psd	464.496	12,07813
12	Test3(2).psd	456.135	Test3(2).psd	456.152	11,90575
13	Test3(3).psd	611.491	Test3(3).psd	611.504	15,78075
14	Test3(4).psd	874.467	Test3(4).psd	874.480	22,546
15	Test3(5).psd	2.964.970	Test3(5).psd	2.964.984	76,23387
16	Test4(1).ppt	1.413.120	Test4(1).ppt	1.413.136	36,437
17	Test4(2).ppt	861.696	Test4(2).ppt	861.712	22,31175
18	Test4(3).ppt	808.960	Test4(3).ppt	808.976	20,92125
19	Test4(4).ppt	751.616	Test4(4).ppt	751.632	19,42163
20	Test4(5).ppt	582.144	Test4(5).ppt	582.160	15,15588

Tabel 4. Hasil Proses Dekripsi File Algoritma *Twofish*

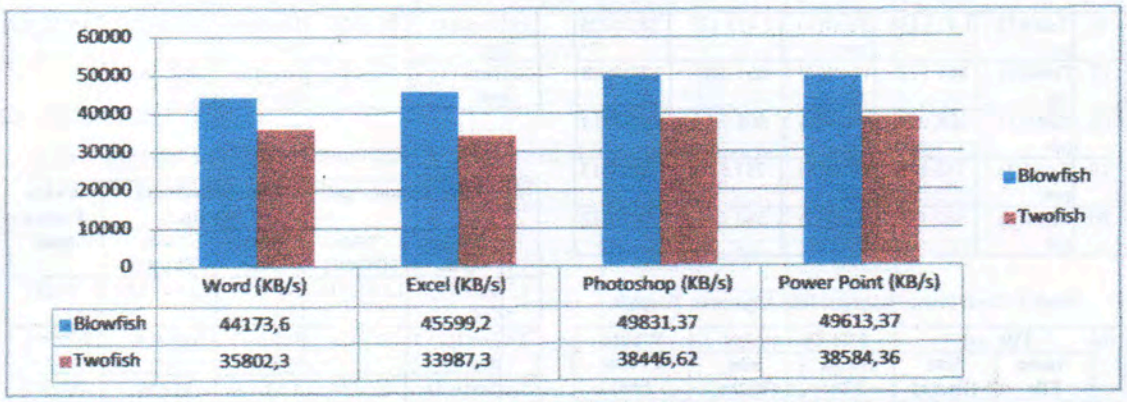
No	File Terenkripsi		File Asli (Hasil Dekripsi)		Waktu Proses (ms)
	Nama File	Size (Bytes)	Nama File	Size (Bytes)	
1.	Test1(1).doc	1.074.704	Test1(1).doc	1,074,688	27,34287
2.	Test1(2).doc	529.424	Test1(2).doc	529.408	13,29663
3.	Test1(3).doc	32.272	Test1(3).doc	32.256	0,796375
4.	Test1(4).doc	661.008	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).psd	464.496	Test3(1).psd	464.484	11,79638
12	Test3(2).psd	456.152	Test3(2).psd	456.135	11,57763
13	Test3(3).psd	611.504	Test3(3).psd	611.491	15,51563
14	Test3(4).psd	874.480	Test3(4).psd	874.467	22,38975
15	Test3(5).psd	2.964.984	Test3(5).psd	2.964.970	74,81162
16	Test4(1).ppt	1.413.136	Test4(1).ppt	1.413.120	36,01537
17	Test4(2).ppt	861.712	Test4(2).ppt	861.696	21,828
18	Test4(3).ppt	808.976	Test4(3).ppt	808.960	20,38975
19	Test4(4).ppt	751.632	Test4(4).ppt	751.616	18,96812
20	Test4(5).ppt	582.160	Test4(5).ppt	582.144	14,71825

4.3. Hasil Pengujian



	Word (KB/s)	Excel (KB/s)	Photoshop (KB/s)	Power Point (KB/s)
Blowfish	44173,6	45599,2	49818,1	49274,9
Twofish	35802,3	33987,3	37730,1	37730,3

Gambar 6. Grafik Kecepatan Proses Enkripsi Algoritma *Blowfish* dan Algoritma *Twofish*



	Word (KB/s)	Excel (KB/s)	Photoshop (KB/s)	Power Point (KB/s)
Blowfish	44173,6	45599,2	49831,37	49613,37
Twofish	35802,3	33987,3	38446,62	38584,36

Gambar 7. Grafik Kecepatan Proses Dekripsi Algoritma *Blowfish* dan Algoritma *Twofish*

Dari Gambar 7 terlihat bahwa algoritma Blowfish lebih cepat dari algoritma Twofish dalam *proses dekripsi*

V. KESIMPULAN

1. Ditinjau dari besar ukuran file sebelum dan sesudah proses enkripsi dan dekripsi dilakukan antara algoritma *Blowfish* dan algoritma *Twofish*, tidak terdapat perbedaan. Ukuran file sebelum dan sesudah proses enkripsi dan dekripsi sama besar.
2. Ditinjau dari kecepatan proses enkripsi antara algoritma *Blowfish* dan algoritma *Twofish*, yang tercepat dalam proses enkripsi adalah algoritma *Blowfish*.
3. Ditinjau dari kecepatan proses dekripsi antara algoritma *Blowfish* dan algoritma *Twofish*, yang tercepat dalam proses dekripsi adalah algoritma *Blowfish*
4. Dilihat dari proses enkripsi, kedua algoritma mampu mengenkripsi file dokumen dengan *extention* *.*doc*, *.*xls*, *.*psd*, *.*ppt*. karena file yang telah terenkripsi tidak bisa dibuka.

REFERENSI

[1] Mulya, M. *Bahan Ajar Kriptografi S-1*. Fakultas Ilmu Komputer Universitas Sriwijaya. 2008.

[2] Pratiwi, A.E., Lhaksmana, K.M., Rizal, S.J. *Implementasi Enkripsi Data dengan Algoritma Blowfish Menggunakan Java Pada Aplikasi Email*. Program Studi Teknik Komputer. Politeknik Telkom Bandung. 2011.

[3] Ratih. *Studi dan Implementasi Enkripsi Pengiriman Pesan Suara Menggunakan Algoritma Twofish*.. Laporan Tugas Akhir. Program Studi Informatika Sekolah Tinggi Teknik Elektro dan Informatika Institut Teknologi Bandung. 2007.

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

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

[6] Syafari, A. *Sekilas Tentang Enkripsi Blowfish*. IlmuKomputer.Com. 2003.