STUDI DAN PERBANDINGAN ALGORITMA IDEA DAN ALGORITMA BLOWFISH

Tri Andriyanto[1)]
D. L. Crispina Pardede[2)]

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Universitas Gunadarma
[1)] triandriyanto@gmail.com, [2)]pardede@staff.gunadarma.ac.id

ABSTRAK

Algoritma International Data Encryption Algorithm (IDEA) dan Algoritma Blowfish adalah algoritma kriptografi simetris dengan kategori cipher blok. Kedua algoritma ini beroperasi dalam bentuk blok bit, dengan ukuran blok sebesar 64 bit. Kedua algoritma ini juga dikenal sangat tangguh dalam mengamankan informasi. Studi dan perbandingan antara algoritma IDEA dan algoritma Blowfish dilakukan untuk membandingkan kinerja algoritma IDEA dan Blowfish dalam hal kecepatan proses dan penggunaan memori pada saat proses enkripsi dan dekripsi suatu file. Untuk dapat membandingkan kinerja algoritma IDEA dan algoritma Blowfish, maka penulis membangun sebuah program enkripsi dan dekripsi file dengan menggunakan bahasa pemrograman JAVA. Dari hasil uji coba program terhadap sampel file teks, file dokumen, file image, file audio, dan file video, terlihat bahwa algoritma IDEA lebih cepat dari algoritma Blowfish dan pemakaian memori kedua algoritma relatif sama.

***Kata Kunci** : Kriptografi, block cipher, IDEA, Blowfish*

1. PENDAHULUAN

Algoritma kriptografi dapat dibagi ke dalam kelompok algoritma simetris dan algoritma asimetris. Algoritma simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama baik untuk proses enkripsi maupun dekripsi. Algoritma simetris dapat dikelompokkan menjadi dua kategori, yaitu *cipher* aliran (*stream chiper*) dan *cipher* blok (*block chiper*). *Cipher* aliran merupakan algoritma kriptografi yang beroperasi dalam bentuk bit tunggal. Sedangkan algoritma kriptografi kategori *cipher* blok beroperasi dalam bentuk blok bit.

Saat ini sudah banyak berkembang algoritma kriptografi simetris baik untuk kategori *cipher* aliran maupun *cipher* blok. Di dalam dunia informatika dikenal *International Data Encryption Algorithm* (IDEA) dan *Blowfish*. Kedua algoritma ini beroperasi dalam bentuk blok bit (*cipher* blok), dengan ukuran blok sebesar 64 bit. Kunci yang digunakan dalam algoritma *Blowfish* sepanjang 32 sampai 488 bit. Sedangkan algoritma IDEA dapat bekerja dengan menerima panjang kunci 128 bit. Kedua algoritma ini dikenal cukup tangguh dalam mengamankan informasi karena sampai saat ini belum ada yang berhasil menembus keamanan kedua algoritma ini (Andi, 2003). Oleh karena itu penulis tertarik untuk mengimplementasikan algoritma IDEA dan *Blowfish* dalam suatu

program aplikasi enkripsi dan dekripsi dan kemudian membandingkan kinerja kedua algoritma pada program aplikasi tersebut.

Pada makalah ini dibahas mengenai cara kerja algoritma IDEA dan algoritma *Blowfish*, dan penerapannya dalam program aplikasi enkripsi dan dekripsi, di mana panjang kunci yang digunakan masing-masing algoritma pada aplikasi ini berukuran 128 bit dan mode operasi enkripsi yang digunakan adalah *Electronic Code Book* (ECB).

Makalah ini bertujuan untuk membandingkan kinerja algoritma IDEA dan Blowfish dalam hal kecepatan proses dan penggunaan memori pada saat proses enkripsi dan dekripsi suatu file.

2. TINJAUAN PUSTAKA

Algoritma IDEA

IDEA (*International Data Encryption Algorithm*) merupakan algoritma simetris yang beroperasi pada sebuah blok pesan terbuka dengan lebar 64 bit dan panjang kunci berukuran 128 bit. Algoritma IDEA menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Dan pesan rahasia yang dihasilkan oleh algoritma ini berupa blok pesan rahasia dengan lebar atau ukuran 64 bit. Algoritma IDEA menggunakan operasi campuran dari tiga operasi aljabar yang berbeda, yaitu: Operasi XOR ($\oplus$), Operasi penjumlahan modulo 2^{16} ($\boxplus$), dan Operasi perkalian modulo ($2^{16}+1$) ($\odot$). Semua operasi ini dilakukan pada subblok 16 bit. Algoritma IDEA melakukan iterasi sebanyak 8 iterasi dan terdapat transformasi keluaran setelah melakukan 8 iterasi.

Proses Enkripsi IDEA

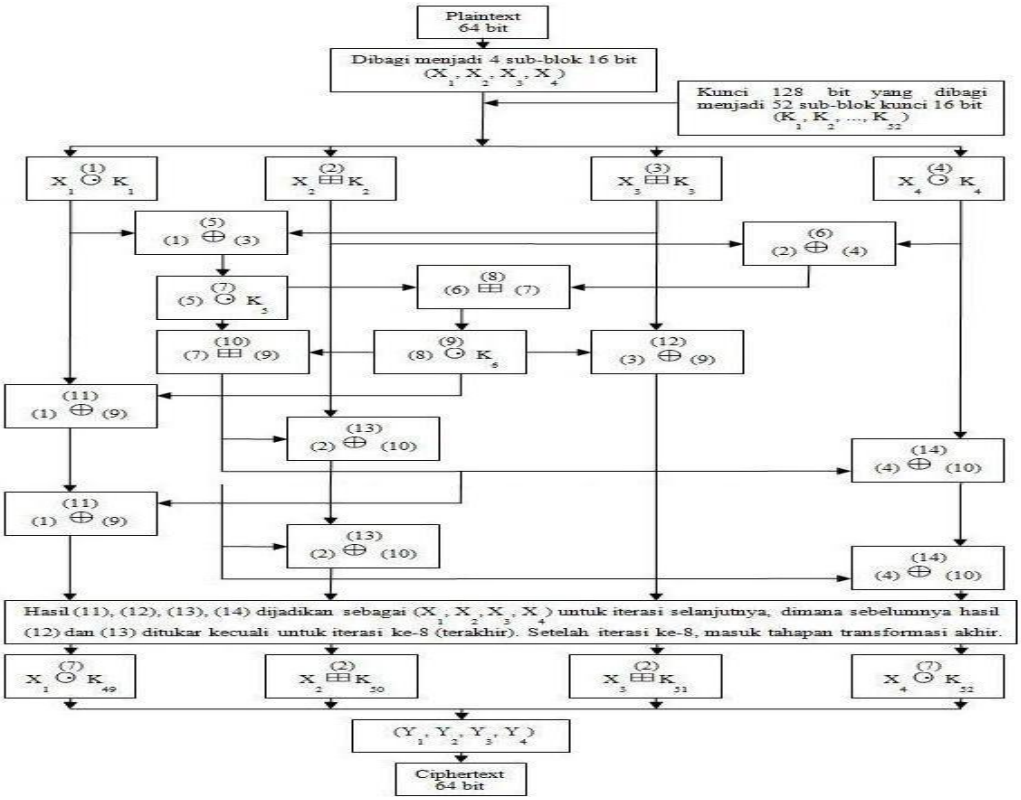
Langkah awal dalam pemrosesan enkripsi dengan algoritma IDEA ini yaitu blok pesan terbuka dengan lebar 64 bit (diperumpamakan dengan X), dibagi menjadi 4 subblok 16 bit, X_1, X_2, X_3, X_4, sehingga X = (X_1, X_2, X_3, X_4). Keempat subblok 16 bit itu ditransformasikan menjadi subblok 16 bit, Y_1, Y_2, Y_3, Y_4, sebagai pesan rahasia 64 bit Y = (Y_1, Y_2, Y_3, Y_4) yang berada di bawah kendali 52 subkunci 16 bit yang dibentuk dari blok kunci 128 bit. Proses enkripsi IDEA dapat dilihat pada Gambar 1.

Proses Dekripsi IDEA

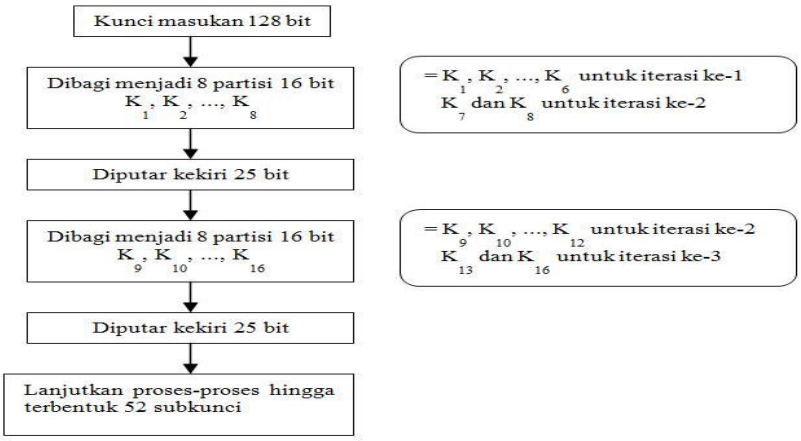
Proses dekripsi menggunakan algoritma yang sama dengan proses enkripsi. Perbedaannya hanya pada 52 buah subkunci yang digunakan masing-masing merupakan hasil turunan 52 buah subkunci enkripsi.

Pembentukan Subkunci IDEA

Sebanyak 52 subkunci 16 bit untuk proses enkripsi diperoleh dari sebuah kunci 128 bit. Pembentukan subkunci untuk proses enkripsi dapat dilihat pada Gambar 2. Subkunci yang digunakan untuk proses enkripsi pada algoritma IDEA dapat dilihat pada Tabel 1. Pembentukan kunci dekripsi didasarkan pada kunci enkripsi yang telah dibentuk sebelumnya. Dari Tabel 2. dapat dilihat perbedaan subkunci yang digunakan untuk enkripsi dengan subkunci untuk dekripsi. Perbedaan yang pertama, yaitu pada urutan penggunaan subkunci. Perbedaan yang kedua, pembentukan subkunci dekripsi menggunakan operasi invers perkalian modulo $2^{16} + 1$ pada subkunci K_{49}, K_{52}, K_{43}, K_{46}, K_{37}, K_{40}, K_{31}, K_{34}, K_{25}, K_{28}, K_{19}, K_{22}, K_{13}, K_{16}, K_7, K_{10}, K_1, K_4 dan menggunakan operasi invers penjumlahan modulo 2^{16} untuk subkunci K_{50}, K_{51}, K_{45}, K_{44}, K_{39}, K_{38}, K_{33}, K_{32}, K_{27}, K_{26}, K_{21}, K_{20}, K_{15}, K_{14}, K_9, K_8, K_2, K_3.



Gambar 1. Proses Enkripsi IDEA



Gambar 2. Pembentukan Subkunci IDEA

Tabel 1. Subkunci Enkripsi IDEA

Putaran Ke-1	$K_1\ K_2\ K_3\ K_4\ K_5\ K_6$
Putaran Ke-2	$K_7\ K_8\ K_9\ K_{10}\ K_{11}\ K_{12}$
Putaran Ke-3	$K_{13} K_{14}\ K_{15}\ K_{16}\ K_{17}\ K_{18}$
Putaran Ke-4	$K_{19}\ K_{20}\ K_{21}\ K_{22}\ K_{23}\ K_{24}$
Putaran Ke-5	$K_{25}\ K_{26}\ K_{27}\ K_{28}\ K_{29}\ K_{30}$
Putaran Ke-6	$K_{31}\ K_{32}\ K_{33}\ K_{34}\ K_{35}\ K_{36}$
Putaran Ke-7	$K_{37}\ K_{38}\ K_{39}\ K_{40}\ K_{41}\ K_{42}$
Putaran Ke-8	$K_{43}\ K_{44}\ K_{45}\ K_{46}\ K_{47}\ K_{48}$
Transformasi Output	$K_{49}\ K_{50}\ K_{51}\ K_{52}$

Tabel 2. Subkunci Dekripsi IDEA

Putaran Ke-1	$(K_{49})^{-1}\ -K_{50}\ -K_{51}\ (K_{52})^{-1}\ K_{47}\ K_{48}$
Putaran Ke-2	$(K_{43})^{-1}\ -K_{44}\ -K_{45}\ (K_{46})^{-1}\ K_{41}\ K_{42}$
Putaran Ke-3	$(K_{37})^{-1}\ -K_{38}\ -K_{39}\ (K_{40})^{-1}\ K_{35}\ K_{36}$
Putaran Ke-4	$(K_{31})^{-1}\ -K_{33}\ -K_{32}\ (K_{34})^{-1}\ K_{29}\ K_{30}$
Putaran Ke-5	$(K_{25})^{-1}\ -K_{26}\ -K_{27}\ (K_{28})^{-1}\ K_{23}\ K_{24}$
Putaran Ke-6	$(K_{19})^{-1}\ -K_{21}\ -K_{20}\ (K_{22})^{-1}\ K_{17}\ K_{18}$
Putaran Ke-7	$(K_{13})^{-1}\ -K_{15}\ -K_{14}\ (K_{16})^{-1}\ -K_{11}\ K_{12}$
Putaran Ke-8	$(K_7)^{-1}\ -K_9\ -K_8\ (K_{10})^{-1}\ K_5\ K_6$
Transformasi Output	$(K_1)^{-1}\ -K_2\ -K_3\ (K_4)^{-1}$

Algoritma Blowfish

Blowfish merupakan metoda enkripsi yang mirip dengan DES. *Blowfish* merupakan blok *cipher* 64 bit dengan panjang kunci variabel. Algoritma ini terdiri dari dua bagian, yaitu : *key expansion* dan enkripsi data. *Key expansion* mengubah kunci yang dapat mencapai 448 bit menjadi beberapa *array* subkunci (*subkey*) dengan total 4168 byte.

Enkripsi data terdiri dari iterasi fungsi sederhana sebanyak 16 kali. Operasi-operasi yang digunakan dalam *Blowfish* adalah: Operasi XOR ($\oplus$), Operasi penjumlahan modulo 2^{32} ($\boxplus$), *Tablelookup* terhadap *array* dengan empat indeks yang dilakukan setiap putaran, yaitu penggunaan S-*box*. Empat 32 bit S-*box* masing-masing mempunyai 256 entri: $S_{1,0}$, $S_{1,1}$, ..., $S_{1,255}$; $S_{2,0}$, $S_{2,1}$, ..., $S_{2,255}$; $S_{3,0}$, $S_{3,1}$, ..., $S3_{,255}$; $S_{4,0}$, $S_{4,1}$, ..., $S_{4,255}$.

Proses Enkripsi *Blowfish*

Blowfish adalah sebuah jaringan Feistel yang mempunyai 16 *round*. Inputnya adalah X elemen data 64 bit. Algoritma *Blowfish* untuk proses enkripsi dapat divisualisasikan seperti pada Gambar 3. Fungsi F dapat divisualisasikan seperti pada Gambar 4.

Proses Dekripsi *Blowfish*

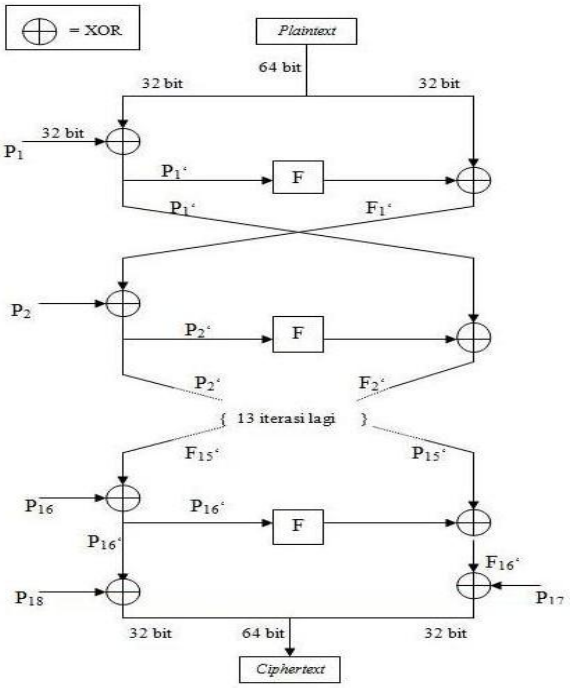
Proses dekripsi *Blowfish* sama persis dengan proses enkripsi *Blowfish*. Perbedaan terletak pada urutan pengggunaan subkunci P_1, P_2,…, P_{18} . Pada proses dekripsi *Blowfish* urutan pengggunaan subkunci P_1, P_2,…, P_{18} dibalik menjadi P_{18}, P_{17},…, P_1.

Pembangkitan Subkunci *Blowfish*

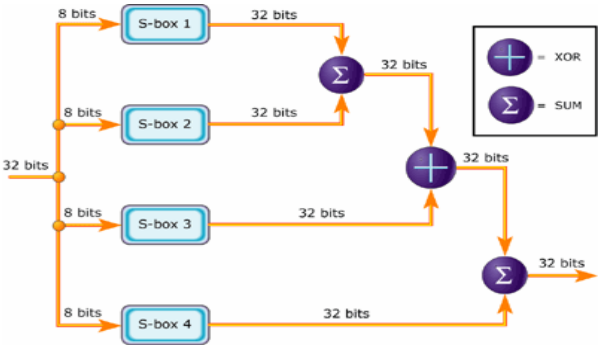
Subkunci dihitung menggunakan algoritma *Blowfish*, di mana secara keseluruhan diperlukan 521 iterasi untuk membangkitkan semua subkunci yang dibutuhkan. Pembangkitan subkunci dapat dilihat pada Gambar 5.

3. METODE PENELITIAN

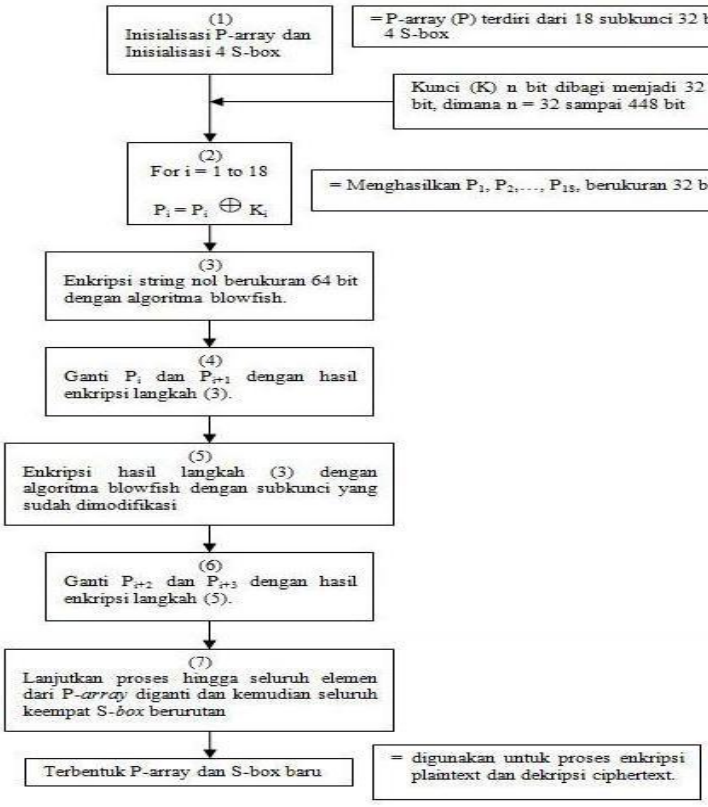
Tahap-tahap yang dilakukan untuk dapat membandingkan kinerja algoritma IDEA dan algoritma Blowfish adalah: tahap pembangunan program enkripsi dan dekripsi file dengan menggunakan algoritma IDEA dan Blowfish dengan panjang kunci berukuran 128 bit. Tahap kedua adalah implementasi di mana penulis melakukan pembangunan program dengan menggunakan bahasa pemrograman JAVA, dengan komponen JAVA *Swing* digunakan untuk membuat tampilan berbasis *Graphical User Interface* (GUI), dan Windows XP sebagai sistem operasi. Dalam pembuatan program ini digunakan paket kriptografi dari provider cryptix. Paket kriptografi ini bersumber dari http://cryptix.org/cryptix-jce-20050328-snap. Tahap akhir adalah uji coba program terhadap file teks (.txt), dokumen (.doc), image (.jpg), audio (.mp3) dan video (.avi). Pada tahap uji coba ini digunakan fasilitas TASK MANAGER yang disediakan sistem operasi windows untuk mengetahui kecepatan proses dan penggunaan memori kedua algoritma pada saat proses enkripsi dan dekripsi file.



Gambar 3. Proses Enkripsi Blowfish



Gambar 4. Fungsi F



Gambar 5. Pembangkitan Subkunci Blowfish

4. HASIL DAN PEMBAHASAN

Konsep Program

Program yang dibuat disini adalah aplikasi enkripsi dan dekripsi IDEA dan *Blowfish*. Program ini digunakan untuk mengenkripsikan dan mendekripsikan sebuah file untuk tipe file teks (.txt), dokumen (.doc), image (.jpg), audio (.mp3) dan video (.avi) dengan menggunakan algoritma IDEA dan *Blowfish*.

Input dan Output

Input program untuk proses enkripsi adalah nama file yang akan dienkripsi. Output yang didapatkan setelah proses enkripsi file yang sudah terenkripsi dan file kunci (*key*). Pada saat melakukan dekripsi file, pengguna memasukkan nama file terenkripsi yang akan didekripsi, serta secara tidak langsung juga memasukkan nama file key.

Uji Coba

Untuk mengetahui perbandingan kinerja algoritma IDEA dan *Blowfish* dalam hal waktu proses dan pemakaian memori pada saat proses enkripsi dan dekripsi file maka dilakukan pengujian terhadap file teks (.txt), dokumen (.doc), image (.jpg), audio (.mp3) dan video (.avi) dan untuk setiap tipe file digunakan lima sampel.

Untuk pengukuran waktu proses dan pemakaian memori, penulis menggunakan fasilitas Task Manager yang terdapat pada sistem operasi Windows dengan satuan detik (*seconds*) untuk kecepatan proses dan satuan kilobytes (KB) untuk pemakaian memori.

Hasil

Pengukuran kecepatan proses berdasarkan pada waktu cpu (*cpu time*) pada Task Manager. Waktu yang diperlukan untuk proses enkripsi dan dekripsi pada file teks dengan menggunakan algoritma IDEA relatif sama dengan waktu proses enkripsi dan dekripsi menggunakan algoritma *Blowfish* (Tabel 3). Waktu proses enkripsi dan dekripsi file dokumen dan file image antara algoritma IDEA dan *Blowfish* hanya menunjukkan sedikit perbedaan. Dari Tabel 6, dapat dilihat waktu enkripsi dan dekripsi algoritma IDEA dan *Blowfish* pada file audio lebih panjang dibanding waktu enkripsi dan dekripsi pada file teks, dokumen dan image. Perbedaan yang cukup signifikan antara waktu enkripsi dan dekripsi algoritma IDEA dengan waktu enkripsi dan dekripsi algoritma *Blowfish* ditemukan pada saat melakukan proses enkripsi dan dekripsi file video.

Tabel 3. Waktu Rata-rata Proses Enkripsi dan Dekripsi

Tipe File	Waktu Enkripsi Rata-rata (s)		Waktu Dekripsi Rata-rata (s)	
	IDEA	Blowfish	IDEA	Blowfish
Teks (.txt)	10	10	10	10
Dokumen (.doc)	10,6	10	10,6	10,4
Image (.jpg)	10,6	10	10,6	10,4
Audio (.mp3)	12,6	11,2	13,2	11,6
Video (.avi)	71,8	28,6	80	38,2

Pengukuran pemakaian memori berdasarkan pada *memory usage* pada Task Manager, kemudian dihitung persentase pemakaian memori dengan menggunakan rumus:

$$\text{Persentase pemakaian memori} = \frac{\textit{memory usage}}{\textit{physical memory}} \times 100\%.$$

Dari pengukuran diketahui bahwa pemakaian memori untuk enkripsi dan dekripsi setiap file teks dengan menggunakan algoritma IDEA relatif sama dengan algoritma *Blowfish* (Tabel 4). Pemakaian memori dalam proses enkripsi dan dekripsi file dokumen untuk algoritma IDEA juga relatif sama dengan algoritma *Blowfish*. Jumlah memori yang digunakan saat enkripsi dan dekripsi file image oleh algoritma IDEA relatif sama dengan algoritma *Blowfish*. Pada enkripsi dan dekripsi file audio, pemakaian memori untuk algoritma IDEA juga relatif sama dengan algoritma *Blowfish*. Demikian pula untuk file video.

Tabel 4. Pemakaian Memory Rata-rata untuk Proses Enkripsi dan Dekripsi

Tipe File	**Physical Memory (KB)**	**Pemakaian Memory Enkripsi Rata-rata (%)**		**Pemakaian Memory Dekripsi Rata-rata (%)**	
		IDEA	**Blowfish**	**IDEA**	**Blowfish**
Teks (.txt)	785.136	0,4612	0,4644	0,4622	0,4644
Dokumen (.doc)		0,4638	0,4658	0,4648	0,4634
Image (.jpg)		0,4656	0,4642	0,4654	0,4646
Audio (.mp3)		0,4660	0,4656	0,4650	0,4690
Video (.avi)		0,4608	0,4658	0,4630	0,4668

Analisa

Berdasarkan hasil-hasil pengujian pada kecepatan proses enkripsi dan dekripsi secara keseluruhan, semakin besar ukuran file yang akan dienkripsi atau didekripsi maka semakin panjang waktu yang diperlukan untuk proses enkripsi atau dekripsi file. Untuk file dengan ukuran kecil kecepatan proses antara algoritma IDEA relatif sama dengan algoritma *Blowfish.* Sedangkan untuk file yang berukuran besar, algoritma *Blowfish* lebih cepat dibandingkan dengan algoritma IDEA. Dari hasil uji coba dalam hal pemakaian memori, diperoleh hasil yang relatif sama antara algoritma IDEA dengan algoritma Blowfish untuk semua tipe file dan semua ukuran file.

Tingkat keamanan suatu algoritma kunci simetris tipe *cipher* blok dapat diukur dari tingkat kerumitan algoritma, panjang blok yang digunakan dan panjang kunci yang digunakan. Untuk kerumitan algoritma, algoritma IDEA sedikit lebih rumit dibandingkan dengan algoritma *Blowfish*. Hal ini dikarenakan operasi-operasi yang dilakukan oleh algoritma *Blowfish* untuk setiap iterasinya, yaitu penjumlahan mod 2^{32}, XOR, dan *tablelookup* lebih sederhana dibandingkan dengan operasi-operasi yang dilakukan oleh algoritma IDEA untuk setiap iterasi, yaitu penjumlahan mod 2^{16}, XOR, dan perkalian mod 2^{16}. Dalam hal panjang blok yang digunakan, kedua algoritma menggunakan panjang blok yang sama dan cukup panjang, yaitu 64 bit. Sedangkan dalam hal panjang kunci yang digunakan, algoritma *Blowfish* menggunakan kunci dengan panjang 32 sampai 448 bit, sedangkan pada algoritma IDEA panjang kunci yang digunakan 128 bit. Jadi, secara keseluruhan kedua algoritma memiliki keunggulan, di mana algoritma IDEA lebih unggul dalam hal kerumitan algoritma, sedangkan algoritma *Blowfish* lebih unggul dalam hal panjang kunci. Sehingga kedua algoritma memiliki tingkat keamanan yang sama-sama tangguh dalam hal mengamankan data atau informasi.

5. KESIMPULAN

Berdasarkan hasil studi dan percobaan dapat disimpulkan bahwa algoritma IDEA dan *Blowfish* beroperasi pada panjang blok yang sama, yaitu 64 bit. Namun panjang kunci yang digunakan algoritma IDEA hanya 128 bit, sedangkan panjang kunci yang digunakan algoritma Blowfish 32 sampai 448 bit. Pada pemrosesannya, algoritma IDEA melakukan iterasi sebanyak 8 iterasi dan terdapat transformasi keluaran akhir setelah melakukan 8 iterasi. Sedangkan iterasi yang dilakukan algoritma *Blowfish* sebanyak 16 iterasi. Dalam pengoperasiannya algoritma IDEA menggunakan operasi campuran yang berbeda, yaitu : operasi XOR, operasi penjumlahan modulo 2^{16}, dan operasi perkalian (2^{16} +1). Sedangkan Algoritma *Blowfish* dalam pengoperasiannya menggunakan operasi-operasi penjumlahan mod 2^{32}, XOR, dan *tablelookup*.

Proses yang dilakukan algoritma *Blowfish* lebih cepat dibanding dengan algoritma IDEA untuk ukuran file yang besar, sedangkan untuk file dengan ukuran yang kecil, kecepatan proses algoritma *Blowfish* sama dengan algoritma IDEA. Sedangkan dalam pemakaian memori, kedua algoritma memakai memori dengan kapasitas yang relatif sama. Dalam hal tingkat keamanan, kedua algoritma memiliki tingkat keamanan yang sama-sama tangguh dalam hal mengamankan data atau informasi.

Program enkripsi dan dekripsi algoritma IDEA dan *Blowfish* ini dapat berjalan dengan baik dan dapat digunakan untuk enkripsi dan dekripsi file dengan berbagai jenis ekstensi, seperti .txt, .doc, .mp3, .bmp, .3gp, .exe, .html, .zip, dan lain-lain.

6. DAFTAR PUSTAKA

Anonim, 2003, *Memahami Model Enkripsi & Security Data*, Andi, Yogyakarta.

Hidayat, Taufik, 2006, *Sistem Kriptografi IDEA*, http://budi.insan.co.id/courses/namadikmenjur/taufik-report.pdf .

http://cryptix.org/cryptix-jce-20050328-snap, 11 April 2006.

Kurniawan, Yusuf, 2004, *Kriptografi Keamanan Internet dan Jaringan Telekomunikasi*, Informatika, Bandung.

Randy, Adhitya, *Studi dan Perbandingan Algoritma Blowfish dan Twofish*, http://informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-038.pdf, 9 Maret 2008.

Ratih, *Studi dan Implementasi Algoritma Blowfish untuk Aplikasi Enkripsi dan Dekripsi File*, 2008, http://informatika.org/~rinaldi/Kriptografi/20062007/Makalah1/ Makalah1-077.pdf.

Stiawan, Deris, 2005, *Sistem Keamanan Komputer*, PT Elex Media Komputindo Kelompok Gramedia, Jakarta.