PERANCANGAN DAN ANALISIS MODIFIKASI KUNCI KRIPTOGRAFI ALGORITMA TWOFISH PADA DATA TEKS

(DESIGN AND ANALYSIS OF TWOFISH ALGORITHM CRYPTOGRAPHY KEY MODIFICATION ON TEXT DATA)

Dwi Anggreni Novitasari [1], R. Rumani, Ir., Drs., MSEE.[2], Rita Magdalena, Ir., MT. [3]
Prodi S1Sistem Komputer,FakultasTeknik, Universitas Telkom
dwianggrenin@ymail.com[1] , rumani@telkomuniversity.ac.id[2], rmy@telkomuniversity.ac.id[3]

Abstrak

Untuk menjaga keamanan suatu data atau informasi yang tersimpan dalam bentuk file dokumen terdapat metode tertentu, salah satunya adalah kriptografi. Kriptografi adalah ilmu yang mempelajari cara untuk menjaga keamanan data agar tetap aman saat dikirimkan, tanpa mengalami gangguan dari pihak ketiga. Data yang dikirimkan bisa berupa informasi umum atau rahasia.

Dalam tugas akhir ini dilakukan suatu perancangan untuk memodifikasi kunci algoritma kriptografi Twofish. Teks dienkripsi dan didekripsi dengan menggunakan algoritma Twofish standar. Setelah proses enkripsi berhasil teks masukan akan dienkripsi dan didekripsi dengan menggunakan algoritma Twofish yang kuncinya telah dimodif. Bagian yang dimodif dari algoritma Twofish adalah pada bagian kuncinya yang difungsikan dengan Blum Blum Shub.

Algoritma Twofish yang digunakan memiliki performansi yang baik, terlihat dari nilai *Avalanche Effect* yang diberikan berkisar antara 0,41 – 0,63. Waktu rata-rata yang digunakan untuk proses enkripsi algoritma Twofish standar adalah 19,375107 detik, sedangkan waktu rata-rata yang diperlukan untuk proses enkripsi pada algoritma Twofish dengan kunci modifikasi adalah 13,835254 detik. Jadi dapat dilihat dari hasil tersebut bahwa waktu yang digunakan pada proses enkripsi dengan menggunakan Algoritma Twofish kunci modifikasi lebih cepat jika dibandingkan dengan algoritma Twofish standar. Memori rata-rata yang digunakan pada algoritma Twofish standar adalah 17,06667 MB, sedangkan memori rata-rata yang digunakan untuk Algoritma Twofish kunci modifikasi adalah 25,63333 MB. Dapat dilihat dari hasil di samping bahwa pada saat penghitungan memori komputasi terlihat bahwa algoritma Twofish kunci modifikasi membutuhkan memori lebih banyak.

***Kata kunci**:* File teks, Kriptografi, Algoritma Twofish, Blum Blum Shub

Abstract

One of the methods to securing data or information inside a document file is cryptography. Cryptography is the study of technique to securing data communication, without unauthorized access from third party. The data sent might be general or secret information.

This final project is focused on designing a modification on Twofish cryptography algorithm. The plaintext is encrypted and decrypted with normal Twofish algorithm. Then the inputted plaintext will be encrypted and decrypted using key modified of Twofish algorithm. The modification done is on the key. The key inputted by user will act as the input for a formula, with output from Blum Blum Shub.

From the testing results, the algorithm is having good performance. The Avalanche Effect is between 0,41 – 0,63. The average encryption time for standard Twofish is 19,375107 second, while The average encryption time for key modified Twofish is 13,835254 second. From the result it can be known that the time used for encryption with key modified Twofish algorithm is faster than standard Twofish algorithm. The average memory used for standard Twofish is 17.06667 MB, while the average memory used for key modified Twofish is 25.63333 MB. From the result it can be known that while measuring the computation memory, the key modified Twofish algorithm is using more memory.

Keyword : Text file, Cryptography, Twofish Algorithm, Blum Blum Shub

1. Pendahuluan

Kriptografi adalah ilmu yang mempelajari cara untuk menjaga keamanan data agar tetap aman saat dikirimkan, tanpa mengalami gangguan dari pihak ketiga. Data yang dikirimkan bisa berupa informasi umum atau rahasia. Yaitu dengan cara mengenkripsi informasi yang akan disampaikan dengan menggunakan suatu kunci. Tetapi sekarang ini sudah banyak orang yang mempelajari metode kriptogafri untuk meyembunyikan suatu informasi.

Oleh karena itu, akan dilakukan suatu perancangan untuk memodifikasi algoritma yang sudah ada. Algoritma yang akan digunakan adalah algoritma Twofish. Algoritma Twofish yang telah ada akan dimodifikasi sehingga walaupun ketahuan algoritma apa yang dipakai dalam teknik kriptografi ini, orang lain tetap akan kesulitan dalam membaca informasi yang akan dikirim karena algoritma tersebut telah dimodifikasi. Algoritma tersebut akan digunakan untuk mengenkripsi dan dekripsi *file* teks.

2. Kriptografi[4]

Kriptografi (*Cryptografi*) berasal dari bahasa Yunani yaitu *cryptos* yang artinya rahasia dan *graphein* yang artinya tulisan. Jadi, kriptografi adalah tulisan rahasia. Kriptografi merupakan ilmu yang mempelajari bagaimana membuat suatu pesan yang dikirim pengirim dapat disampaikan kepada penerima dengan aman. Namun kriptografi tidak saja memberikan keamanan informasi, tetapi lebih kearah teknik-tekniknya.

Ada empat tujuan mendasar dari ilmu kriptografi, yaitu :

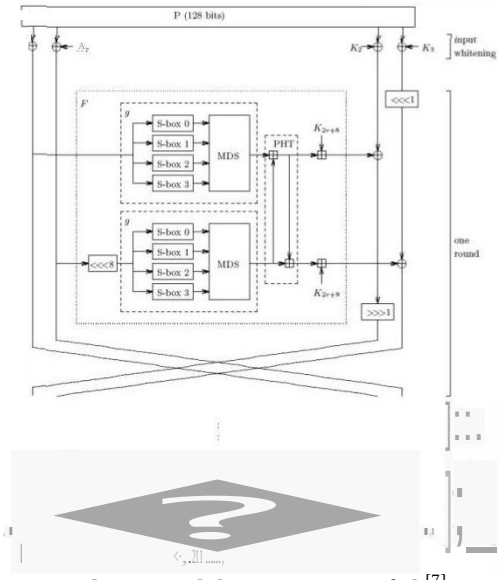
- Kerahasiaan (*confidentiality*), adalah layanan yang digunakan untuk menjaga isi dan informasi suatu data dari siapapun kecuali yang memiliki otoritas untuk membuka isi dan informasi data tersebut.
- Integritas data (*data integrity*), adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data dari pihak-pihak yang tidak berhak, antara lain menyangkut penyisipan, penghapusan, dan pensubtitusian data lain kedalam data yang sebenarnya.
- Autentikasi data (*autthentication*), adalah berhubungan dengan identifikasi, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengirimannya, dan lain-lain.
- Anti penyangkalan (*non-repudiation*), adalah layanan yang dapat mencegah suatu pihak untuk menyangkal aksi pengiriman suatu informasi yang dilakukan sebelumnya.

3. Algoritma Twofish[7]

Algoritma Twofish merupakan algoritma kuat yang sampai saat ini dinyatakan aman karena masih belum ada serangan kriptanalisis yang benar – benar dapat mematahkan algoritma ini. Algoritma ini juga tidak dipatenkan sehingga penggunaannya pada alat enkripsi tidak perlu mengeluarkan biaya.

Beberapa keunggulan algoritma kriptografi Twofish yaitu:

- Memiliki varian dengan sebuah nomor variabel dari setiap *round*.
- Memiliki *key schedule* yang dapat diprakomputasikan untuk kecepatan maksimum dan penggunaan memori minimum.
- Cocok sebagai *stream chipper*, fungsi hash satu arah, MAC dan *pseudo random number generator*, dengan menggunakan metode konstruksi yang dapat dimengerti.
- Memiliki varian *family-key* untuk memungkinkan versi *chipper* yang berbeda dan *non interrupterable*.



Gambar 3.1 Blok Diagram Twofish[7]

Pada gambar 2.3 menunjukkan blok diagram algoritma Twofish. Twofish menggunakan 16 putaran jaringan Feistel dengan proses *whitening* tambahan pada input dan output.

Plaintext dibagi menjadi empat bagian, dengan masing-masing kata memiliki ukuran 32 bit. Pada langkah *whitening* dilakukan fungsi XOR terhadap 4 kata kunci ini. Setelah dilakukan 16 putaran feistel. Pada setiap putaran, dua kata-kata pada sisi kiri digunakan sebagai masukan kepada fungsi g (salah satu darinya diputar pada 8 bit pertama). Fungsi g terdiri dari empat *byte-wide S-Box key-dependent*, yang diikuti oleh suatu langkah pencampuran linier berdasarkan suatu matriks MDS.

Hasil kedua fungsi g dikombinasikan menggunakan suatu *Pseudo Hadamard Transform* (PHT), dan ditambahkan dua kata kunci. Kedua hasil ini kemudian di-XOR ke dalam kata-kata pada sisi kanan (salah satunya diputar ke kanan 1 bit pertama, yang lainnya diputar ke kanan setelahnya).

Pada sisi kiri dan kanan dibagi menjadi dua kemudian ditukar untuk putaran yang berikutnya, kemudian pada putaran terakhir dibalik, dan pada ke empat kata di-XOR dengan lebih dari empat kata kunci untuk menghasilkan *chipertext*.

Secara formal, 16 byte *plaintext* $p_0,\ldots p_{15}$ yang pertama dipecah menjadi 4 kata $P_0,\ldots P_3$ dari 32 bit masing-masing menggunakan konvemsi *little-endian*.

$$P_i = \sum_{j=0}^{3} P_{(4i+j)}.2^{8j} \qquad i = 0,1,2,3 \qquad (2.1)$$

Dalam langkah *whitening*, kata-kata ini di-XOR dengan 4 kata dari kunci yang diperluas.

$$R_{0,i} = P_i \oplus K_i \qquad i = 0,1,2,3 \qquad (2.2)$$

Pada setiap 16 putaran, dua kata pertama digunakan sebagai masukan kepada fungsi F, yang juga mengambil angka bulat tersebut sebagai masukan. Kata yang ketiga dilakukan fungsi XOR dengan keluaran pertama F dan kemudian diputar ke kanan satu bit. Kata ke-empat diputar ke kiri satu bit kemudian dilakukan fungsi XOR dengan kata keluaran F yang kedua. Kemudian keduanya saling ditukar menghasilkan persamaan :

$$(F_{r,0},F_{r,1}) = F(F_{r,0},F_{r,1,r}) \qquad (2.3)$$

$$R_{r+1,0} = ROR((R_{r,2} \oplus F_{r,0},1) \qquad (2.3)$$

$$R_{r+1,1} = ROL(R_{r,3},1) \oplus F_{r,1} \qquad (2.3)$$

$$R_{r+1,2} = R_{r,0} \qquad (2.3)$$

$$R_{r+1,3} = R_{r,1} \qquad (2.3)$$

Dimana R : kata masukan

r : *round* , untuk r = 0,…15

ROR : rotasi bit ke kanan

ROL : rotasi bit ke kiri

Langkah *whitening* keluaran membatalkan pertukaran terakhir dan melakukan fungsi XOR kata-kata dengan 4 kata dari kunci yang diperluas.

$C_i = R_{16,(i+2) \bmod 4} \oplus K_{i+4}$ i = 0,1,2,3 (2.4)

Empat kata dari *chipertext* kemudian ditulis seperti 16 byte $c_0,\ldots,c_{15}$ sama seperti menggunakan konversi *little-endian* untuk *plaintext.*

$c_i = \lfloor \frac{\quad / \quad}{\quad} \rfloor \bmod 2^8$ i = 0,…,15 (2.5)

4. Blum Blum Shub[2]

Blum blum shub adalah sebuah pembangkit bilangan acak semu (*pseudo random genertor*) yang aman secara kriptografi, diusulkan tahun 1986 oleh Lenore Blum, Manuel Blum dan Michael Shub. Secara sederhana algoritmanya adalah sebagai berikut :

1. Pilih dua buah bilangan prima rahasia, p dan q, yang masing-masing kongruen dengan 3 modulo 4.
2. Kalikan keduanya menjadi n = pq. Bilangan m ini disebut bilangan bulat Blum.
3. Pilih bilangan bulat acak lain, s, sebagai umpan sedemikian sehingga:

(i) $2 < s < n$

(ii) s dan n relatif prima kemudian hitung $x_0 = s^2 \bmod n$.

4. Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan:

(i) Hitung $x_i = x_{i-1}^{\ 2} \bmod n$

(ii) z_i = bit LSB (*Least Significant Bit*) dari x_i Barisan bit acak yang dihasilkan adalah adalah z_1, z_2, z_3, …

Keamanan BBS terletak pada sulitnya memfaktorkan n. Nilai n tidak perlu rahasia dan dapat diumumkan kepada publik. BBS tidak dapat diprediksi dari arah kiri (*unpredictable to the left*) dan tidak dapat diprediksi dari arah kanan. Artinya jika diberikan barisan bit yang dihasilkan oleh BBS, kriptanalis tidak dapat memprediksi barisan bit sebelumnya dan barisan bit sesudahnya.

5. Perancangan Sistem

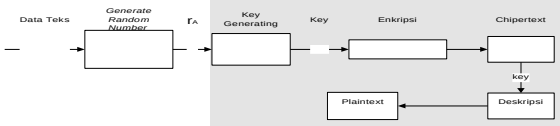
Perancangan sistem mencakup diagram alir sistem, pemodelan sistem, dan perancangan antarmuka.

5.1 Rancangan Pemodelan Sistem

Gambar 5.1 Rancangan Pemodelan Sistem Enkripsi Dekripsi Sebelum Modifikasi (Standar)

Proses pemodelan sistem adalah :

1. Input data teks.
2. Input *key*.
3. Dilakukan proses enkripsi yang menghasilkan *chipertext*.
4. Setelah dihasilkan *chipertext* akan dilakukan proses dekripsi yang kemudian menghasilkan plaintext awal.



Gambar 5.2 Pemodelan Sistem (Setelah Modifikasi)

Proses pemodelan sistem adalah :

1. Input data teks.
2. Kemudian digenerate *random number*.
3. *Random number* kemudian di-generate *key*.
4. Dari *key*, data teks kemudian dienkripsi.

5. Hasil dari enkripsi adalah chiper teks.

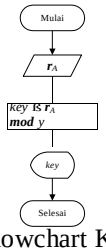
5.1.1 Flowchart Generate Random Number

Gambar 2.3 Flowchart Generate Random Number

Proses Generate Random Number

1. Input data teks.
2. Input *true random*.
3. Kemudian didapatkan *seed* yang berasal dari *true random*.
4. Kemudian seed akan difungsikan dengan BBS (*Blum Blum Shub*).
5. Maka didapatkanlah r_A (*random number*).

5.1.2 Flowchart Key Generating



Gambar2.4 Flowchart Key Generating

Proses Key Generating :

1. Random number yang didapatkan dari Generate Random Number, dimana x dan y saling invers.
2. Dimana hasil Rnadom Number kemudian dipangkatkan dengan *x* lalu dimodulus dengan *y* , maka dihasilkan *key*.

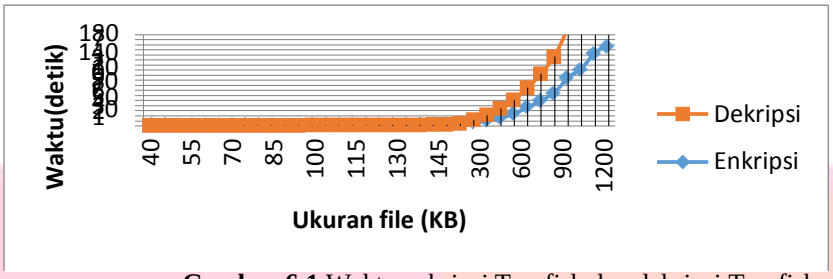
6. Pengujian Dan Analisis

6.1 Pengujian Waktu Enkripsi dan Dekripsi

Pengujian ini dilakukan dengan mengukur waktu proses enkripsi ketika *file* dikirim ke penerima sedangkan waktu proses dekripsi dilakukan ketika pesan diterima. Teknik pengukuran dilakukan dengan panjang kunci yang sama yaitu 6 karakter dengan besar *file* yang berbeda dan dengan besar *file* yang sama yaitu 550KB dengan panjang kunci yang berbeda-beda. Teknik pengukuran waktu proses enkripsi dan dekripsi adalah nanosecond / 1000000000 hasilnya akan menjadi dalam second.

6.1.1 Perbandingan Waktu Enkripsi dan Dekripsi dengan Besar File yang Berbeda dengan Kunci yang Sama

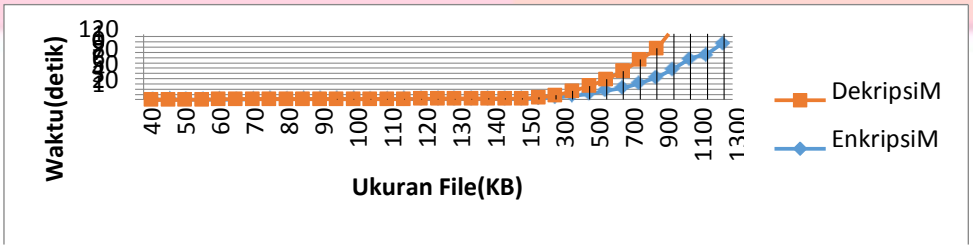
a. Waktu enkripsi Twofish dan dekripsi Twofish.



Gambar 6.1 Waktu enkripsi Twofish dan dekripsi Twofish

Dari grafik waktu di atas bisa kita lihat jika besar memori yang digunakan semakin besar maka waktu enkripsi dan dekripsi yang digunakan juga semakin lama.

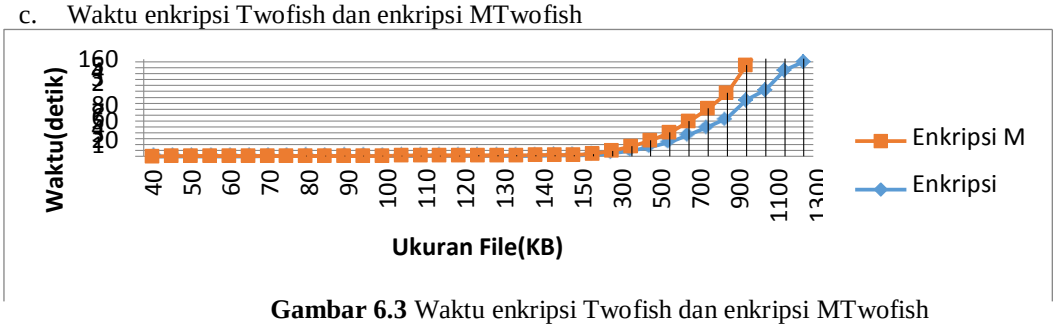
b. Waktu enkripsi MTwofish dan dekripsi MTwofish untuk file ukuran



Gambar 6.2 Waktu enkripsi MTwofish dan dekripsi MTwofish

Dari grafik waktu di atas bisa kita lihat jika besar memori yang digunakan semakin besar maka waktu enkripsi MTwofish dan dekripsi MTwofish yang digunakan juga semakin lama.

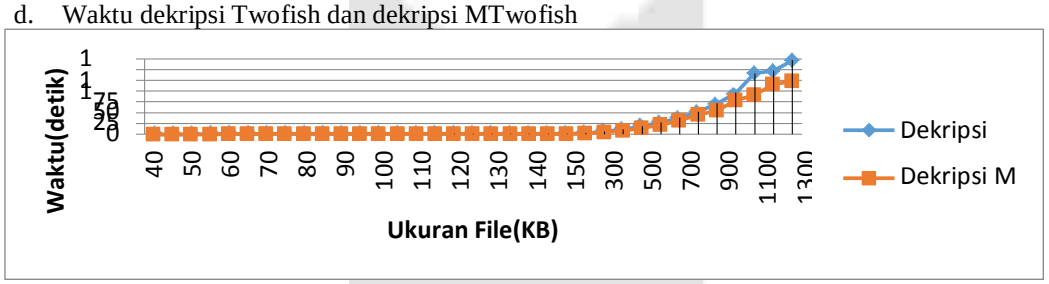
c. Waktu enkripsi Twofish dan enkripsi MTwofish



Gambar 6.3 Waktu enkripsi Twofish dan enkripsi MTwofish

Dari grafik waktu di atas dapat dilihat bahwa perubahan waktu yang diperlukan untuk proses enkripsi Twofish dan MTwofish masih belum stabil hingga file yang dijadikan masukan berukuran 90KB – 2000KB. Perubahan waktu enkripsi Twofish dan enkripsi MTwofish mulai terlihat dari file berkuran 90KB, yaitu waktu yang digunakan untuk enkripsi Twofish lebih besar jika dibandingkan dengan waktu yang digunakan untuk enkripsi MTwofish. Perbedaan waktu yang terlihat dari table di atas cukup signifikan yaitu mencapai 58,04419 detik.

d. Waktu dekripsi Twofish dan dekripsi MTwofish



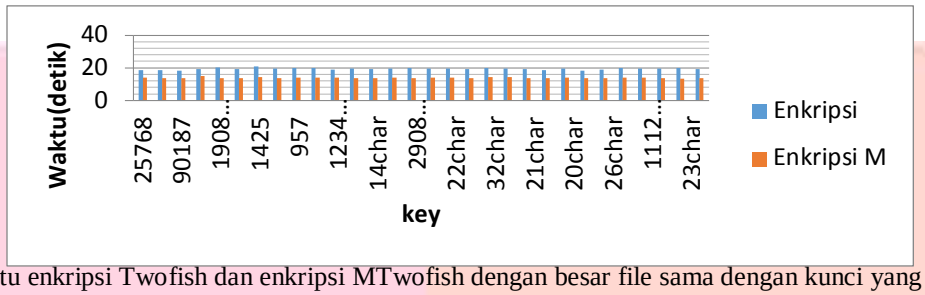
Gambar 6.4 Waktu dekripsi Twofish dan dekripsi MTwofish

Dari grafik waktu di atas dapat dilihat bahwa perubahan waktu yang diperlukan untuk proses dekripsi Twofish dan MTwofish sudah sejak awal. Perubahan waktu dekripsi Twofish dan dekripsi MTwofish sudah terlihat sejak file berkuran 40KB, yaitu waktu yang digunakan untuk dekripsi Twofish lebih besar jika dibandingkan

dengan waktu yang digunakan untuk dekripsi MTwofish. Perbedaan waktu yang terlihat dari table di atas cukup signifikan yaitu mencapai 50.8505 detik.

6.1.2 Perbandingan Waktu Enkripsi dan Dekripsi dengan Besar File yang Sama dengan Panjang Kunci yang Berbeda

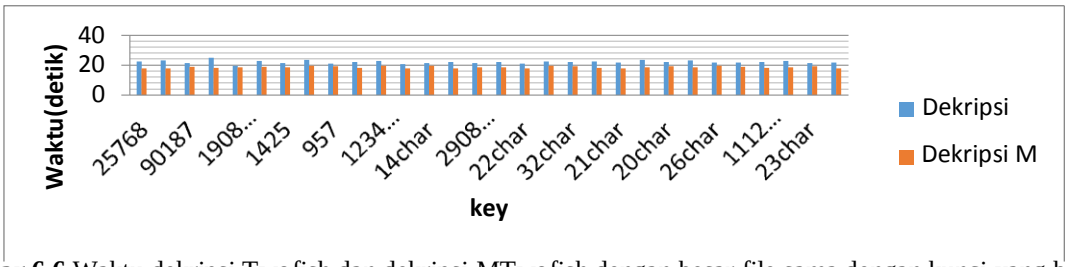
a. Waktu enkripsi Twofish dan enkripsi MTwofish dengan besar file sama dengan kunci yang berbeda.



Gambar 6.5 Waktu enkripsi Twofish dan enkripsi MTwofish dengan besar file sama dengan kunci yang berbeda

Dari grafik di atas dapat dilihat bahwa perubahan waktu yang digunakan untuk melakukan proses enkripsi Twofish dan enkripsi MTwofish tidak terlalu besar walapun menggunakan panjang kunci yang berbeda-beda. Waktu rata-rata yang diperlukan untuk proses enkripsi Twofish adalah 19,375 detik serta waktu yang diperlukan untuk proses enkripsi MTwofish adalah 13,835 detik.

b. Waktu dekripsi Twofish dan dekripsi MTwofish dengan besar file sama dengan kunci yang berbeda.



Gambar 6.6 Waktu dekripsi Twofish dan dekripsi MTwofish dengan besar file sama dengan kunci yang berbeda

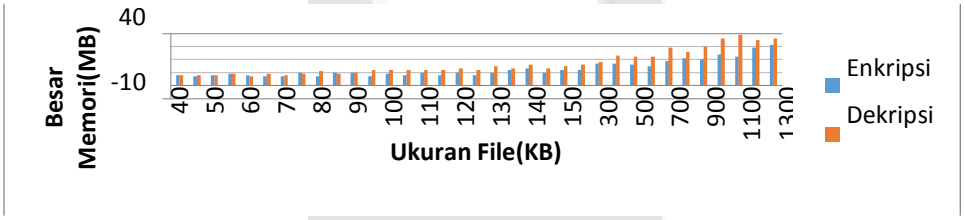
Dari grafik di atas dapat dilihat bahwa perubahan waktu yang digunakan untuk melakukan proses dekripsi Twofish dan dekripsi MTwofish tidak terlalu besar walapun menggunakan panjang kunci yang berbeda-beda. Waktu rata-rata yang diperlukan untuk proses dekripsi Twofish adalah 22,066 detik serta waktu yang diperlukan untuk proses dekripsi MTwofish adalah 18,598 detik.

6.2 Pengujian Memori Enkripsi dan Dekripsi

Pengujian ini dilakukan dengan mengukur memori yang diperlukan untuk proses enkripsi ketika *file* dikirim ke penerima sedangkan memori proses dekripsi dilakukan ketika pesan diterima. Teknik pengukuran memori proses enkripsi dan dekripsi adalah max memori – total memori hasilnya akan menjadi dalam *byte*.

6.2.1 Perbandingan Memori Enkripsi dan Dekripsi dengan Besar File yang Berbeda dengan Kunci yang Sama

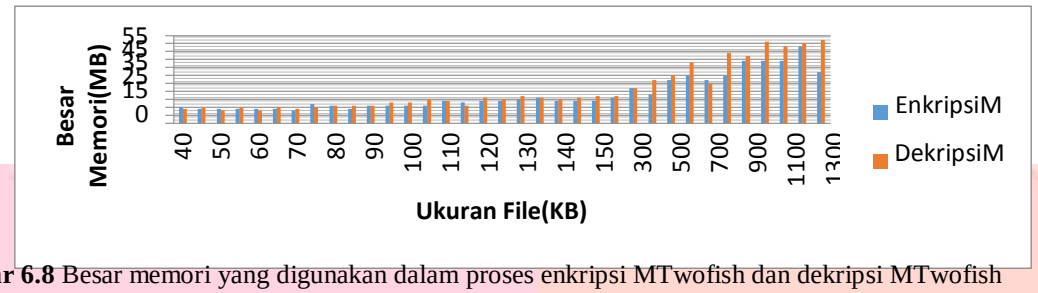
a. Besar memori yang digunakan dalam proses enkripsi dan dekripsi Twofish.



Gambar 6.7 Besar memori yang digunakan dalam proses enkripsi dan dekripsi Twofish

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang terpakai. Sehingga dapat dikatakan bahwa memori yang digunakan untuk proses komputasi semakin besar mengikuti penambahan besar file yang digunakan.

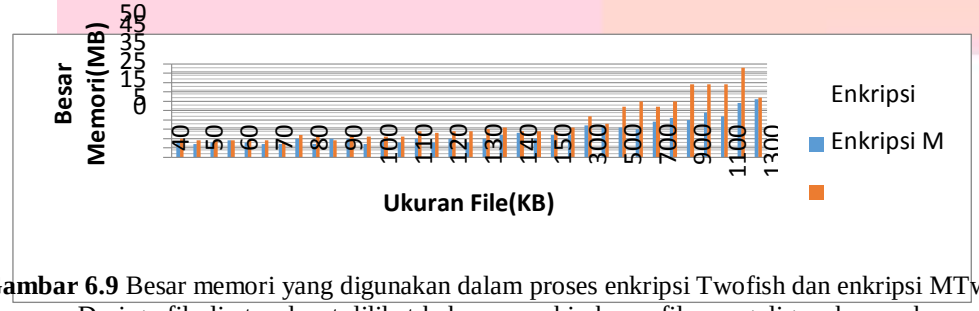
b. Besar memori yang digunakan dalam proses enkripsi MTwofish dan dekripsi MTwofish.



Gambar 6.8 Besar memori yang digunakan dalam proses enkripsi MTwofish dan dekripsi MTwofish

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang terpakai. Sehingga dapat dikatakan bahwa memori yang digunakan untuk proses komputasi semakin besar, mengikuti penambahan besar file yang digunakan.

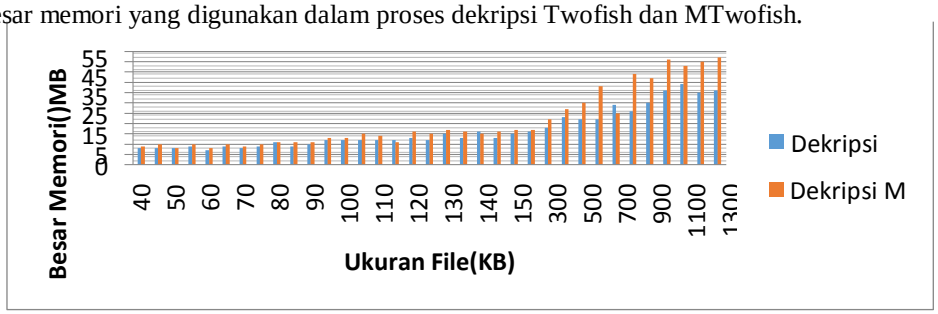
c. Besar memori yang digunakan dalam proses enkripsi Twofish dan enkripsi MTwofish.



Gambar 6.9 Besar memori yang digunakan dalam proses enkripsi Twofish dan enkripsi MTwofish

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang terpakai. Sehingga dapat dikatakan bahwa memori yang digunakan untuk proses komputasi semakin besar, mengikuti penambahan besar file yang digunakan. Dapat dilihat juga bahwa memori komputasi untuk enkripsi Twofish yang dimodif kuncinya lebih besar dibandingkan dengan Twofish standar.

d. Besar memori yang digunakan dalam proses dekripsi Twofish dan MTwofish.



Gambar 3.10 Besar memori yang digunakan dalam proes dekripsi Twofish dan MTwofish

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang terpakai. Sehingga dapat dikatakan bahwa memori yang digunakan untuk proses komputasi semakin besar, mengikuti penambahan besar file yang digunakan. Dapat dilihat juga bahwa memori komputasi untuk dekripsi Twofish yang dimodif kuncinya lebih besar dibandingkan dengan Twofish standar.

4.3 Pengujian *Avalanche Effect*

Pada kriptografi, hasil yang diberikan sangat unik, berbeda dari data yang menjadi masukan proses tersebut. Sedikit perubahan pada data masukan dapat memberikan perubahan yang signifikan pada hasil poses kriptografi, dan perubahan tersebut dinamakan *avalanche effect*. Semakin besar perubahan yang terjadi, semakin baik performansi algoritma kriptografi tersebut.

a. Pengujian *Avalanche Effect* ketika *plaintext* nya diubah 1 karakter :

Plaintext 1 : 00000000000000000000000000
Plaintext 2 : 00000000000000000000000001
Kunci : 00000000000000000000000000
Ciphertext 1 : 11111010011011010011111101101100
Chipertext 2 : 11101001101010110111011100011011

Avalanche Effect = ————————

Untuk pengujian *Avalanche Effect* dengan cara *plaintext*-nya diubah 1 karakter, hasilnya cukup baik yaitu 40,63% - 59,38%. Dapat dilihat pada lampiran B.

b. Pengujian *Avalanche Effect* ketika kuncinya nya diubah 1 karakter :

Plaintext : 00000000000000000000000
Kunci 1 : 00000000000000000000000
Kunci 2 : 00000000000000000000001
Ciphertext 1 : 11111010011011010011111101101100
Ciphertext 2 : 10110011111101111101111010111001
Avalanche Effect = ————————

Untuk pengujian *Avalanche Effect* dengan cara *plaintext*-nya diubah 1 karakter, hasilnya cukup baik yaitu 40,63% - 62,5%. Dapat dilihat pada lampiran B.

7. Kesimpulan dan Saran

7.1 Kesimpulan

Dari pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Waktu yang diperlukan untuk proses enkripsi dan dekripsi dengan menggunakan Algoritma Twofish standar lebih lama jika dibandingkan dengan menggunakan Algoritma Twofish yang sudah dimodifikasi kuncinya.
2. Memori yang digunakan dalam proses enkripsi dan dekripsi dengan menggunakan Algoritma Twofish lebih kecil jika dibandingkan dengan menggunakan Algoritma Twofish yang sudah dimodifikasi kuncinya.
3. Walaupun panjang kunci yang digunakan berbeda-beda tetapi jika file yang digunakan sama, pada saat menjalankan proses enkripsi dan dekripsi tidak akan terlalu berpengaruh terhadap terhadap waktu dan memori yang diperlukan dalam proses komputasi baik menggunakan Algoritma Twofish standar atau Algoritma Twofish yang sudah dimodifikasi kuncinya.
4. Hasil *avalanche effect* yang didapatkan ketika *plaintext* diubah 1 karakter adalah 0,41 – 0,59. Sedangkan hasil *avalanche effect* ketika kunci diubah 1 karakter adalah 0,41 – 0,63.

5.1 Saran

Dari sistem yang telah dibangun tentu masih banyak kekurangan dan masih perlu untuk dikembangkan lagi. Saran untuk pengembangan pada sistem ini adalah sebagai berikut :

1. Proses modifikasi dilakukan pada algoritmanya secara keseluruhan tidak hanya pada kuncinya.
2. Menggunakan masukan yang berbeda misalnya file *.doc, *.pdf, dan gambar.

8. Referensi

[1] Ariyus, Doni. (2008). *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta : ANDI.

[2] *Blum Blum Shub Cryptosystem and Generator*. [Online]. Tersedia : http://diamond.boisestate.edu/~liljanab/ISAS/course_materials/BBSpresentation.pdf [9 Januari 2015].

[3] McLauchlin, Jamie. *Random Number Generation for Cryptography*. [Online]. Tersedia : http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/mclaucwj/Random Number_Generation_for_Cryptography.html [9 Januari 2015].

[4] Munir, Rinaldi. (2006). Kriptografi. Bandung : Informatika

[5] Nazernasrisamaaja. (2011). Kriptografi Simetris,Asimetris, dan Hybrid, Jakarta.

[6] Online tersedia : http://elib.unikom.ac.id/files/disk1/385/jbptunikompp-gdl-novandwian-19207-7-babiii.pdf [3 April 2015]

[7] *Schneier on Security*. [Online]. Tersedia : https://www.schneier.com/twofish.html [7 Januari 2015].

[8] Septiarini Anindita, Hamdani. 1 Februari. (2011). "Sistem Kriptografi Untuk *Text Message* Menggunakan Metode Affine ". p 1-2.