PERANCANGAN DAN ANALISIS MODIFIKASI KUNCI KRIPTOGRAFI ALGORITMA RC6 PADA DATA TEKS

(DESIGN AND ANALYSIS OF RC6 ALGORITHM CRYPTOGRAPHY KEY MODIFICATION ON TEXT DATA)

Dewi Siskawati [1], R. Rumani, Ir., Drs., MSEE.[2], Rita Magdalena, Ir., MT. [3]
Prodi S1Sistem Komputer,FakultasTeknik, Universitas Telkom
siska_dewi2701@yahoo.com[1] , rumani@telkomuniversity.ac.id[2], rmy@telkomuniversity.ac.id[3]

Abstrak

Keamanan suatu data atau informasi berupa file dokumen sangat penting, salah satu cara pengamanan data adalah dengan menggunakan metode kriptografi. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Data yang dikirimkan bisa berupa informasi umum atau rahasia.

Dalam tugas akhir ini dibuat suatu perancangan algoritma kriptografi RC6 yang dimodifikasi kuncinya dalam bentuk aplikasi Java. Masukan dari aplikasi tersebut adalah teks, kemudian teks dienkripsi dan didekripsi menggunakan algoritma RC6 dengan kunci biasa. Terakhir, dilakukan proses enkripsi dan dekripsi menggunakan algoritma RC6 dengan kunci yang telah dimodifikasi. Kunci yang dimodifikasi yakni kunci yang difungsikan dengan *Blum Blum Shub*.

Algoritma RC6 yang digunakan memiliki performansi yang baik, terlihat dari nilai *Avalanche Effect* kunci biasa RC6 yang diberikan berkisar antara 46.875% sampai 65.625% dan nilai *Avalanche Effect* kunci modifikasi RC6 yang diberikan berkisar antara 43.75% sampai 62.5%. Rata-rata waktu enkripsi kunci biasa RC6 yang dihasilkan yaitu 3.94939 detik dan rata-rata waktu enkripsi kunci modifikasi RC6 yang dihasilkan yaitu 3.72655 detik. Rata-rata memori kunci biasa RC6 yang digunakan yaitu 20 MB dan rata-rata memori kunci modifikasi RC6 yang digunakan yaitu 23 MB. Dapat disimpulkan bahwa waktu enkripsi kunci modifikasi RC6 lebih cepat daripada waktu enkripsi kunci biasa RC6 dan memori yang digunakan kunci modifikasi RC6 lebih banyak daripada memori yang digunakan kunci biasa RC6.

***Kata kunci**:* File teks, Kriptografi, Algoritma RC6, *Blum Blum Shub*

Abstract

The information and data security is very important in data transfer process, one way of securing data is by using cryptographic methods. Cryptography is the science that studies mathematical techniques related to aspects of information security, such as data confidentiality, data authenticity, data integrity, and data authentication. Transmitted data can be public or confidential information.

In this final project, a design of cryptographic key modified algorithms RC6 in the form of Java applications is created. The input of the application is a text, then the text is encrypted and decrypted using the algorithm RC6 with a regular key. And then, the encryption and decryption process is carried out using the RC6 algorithm with a key that has been modified. The modified key is enabled with Blum-Blum Shub.

The used of RC6 algorithm has a good performance, seen from the regular key RC6 Avalanche Effect in the range of from 46 875% to 65 625% and the value of the modification key RC6 Avalanche Effect in the range of 43.75% to 62.5%. Average time for the ordinary RC6 encryption key is 3.94939 seconds and the average time for the modification RC6 encryption key is 3.72655 seconds. The average memory used for regular key RC6 is 20 MB and the average memory used for modified RC6 key is 23 MB. It can be concluded that the encryption time of modified RC6 encryption key is faster than the ordinary RC6 encryption key and memory used for modified key RC6 is bigger than the memory used in regular RC6 key.

Keyword : Text file, Cryptography, RC6 Algorithm, Blum Blum Shub

1. Pendahuluan

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Perkembangan teknologi yang semakin pesat saat ini tidak hanya berdampak baik dalam memudahkan bertukar data dan mendapatkan informasi saja, namun hal ini juga bisa menyebabkan kerugian bagi pihak yang melakukan komunikasi karena semakin banyak cara yang bisa dilakukan oleh pihak-pihak yang tidak bertanggung jawab

yang ingin mengetahui bahkan menghilangkan informasi tersebut agar tidak sampai pada pihak penerima. Hal ini menyebabkan keamanan informasi tidak terjamin lagi dan merupakan ancaman terhadap keamanan data yang semakin pesat.

Semakin banyak serangan yang mungkin terjadi dalam proses pertukaran data maupun mendapatkan informasi, perlu adanya suatu teknik atau metode agar meningkatkan keamanan informasi yaitu dengan teknik kriptografi. Kriptografi yaitu informasi yang ingin disampaikan dienkripsi terlebih dahulu menggunakan suatu kunci agar tidak dapat langsung diketahui maknanya. Namun, sekarang ini sudah banyak digunakan algoritma kriptografi untuk menyembunyikan suatu informasi.

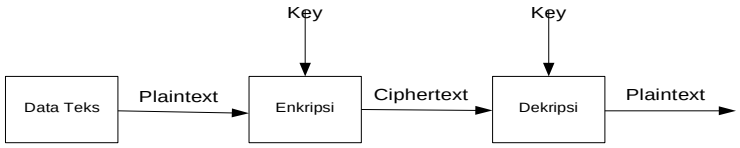
Oleh karena itu, penulis membuat sebuah implementasi algoritma kriptografi yang sudah ada dan algoritma tersebut akan dimodifikasi bagian kuncinya. Kriptografi yang digunakan yakni kriptografi algoritma RC6. Dilakukan modifikasi kunci pada algoritma RC6 agar pihak yang tidak berhak tetap kesulitan dalam membaca informasi yang akan dikirim ke pihak yang berhak walaupun sudah mengetahui algoritma yang dipakai. Dan informasi yang akan dienkripsi yakni berupa file teks.

2. Perancangan

2.1 Perancangan Sistem

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

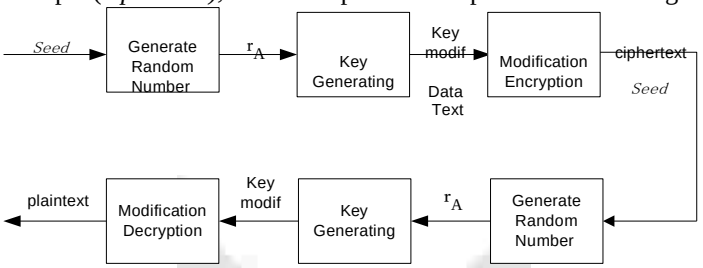
2.1.1 Rancangan Pemodelan Sistem



Gambar 2.1 Rancangan Pemodelan Sistem Enkripsi Dekripsi Standar (Sebelum Modifikasi)

Proses pemodelan sistem Enkripsi Dekripsi sebelum modifikasi antara lain:

1. Input data teks.
2. Input *key*.
3. Dilakukan proses enkripsi. Kemudian menghasilkan *ciphertext.*
4. Dari hasil enkripsi (*ciphertext)*, dilakukan proses dekripsi kemudian menghasilkan *plaintext* awal.

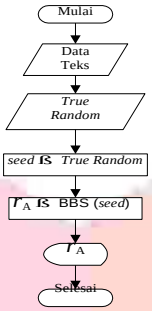


Gambar 2.2 Rancangan Pemodelan Sistem.Enkripsi Dekripsi Modifikasi Kunci (Setelah Modifikasi)

Proses pemodelan sistem Enkripsi Dekripsi Modifikasi antara lain:

1. Input *Seed*.
2. Kemudian *Seed* dilakukan *Generate Random Number.*
3. Kemudian hasil dari *Generate Random* di-*Generate key.*
4. Maka dihasilkan *key* modif, data teks kemudian dienkripsi.
5. Hasil dari enkripsi adalah *ciphertext.*
6. Kemudian *ciphertext* didekripsi dengan melakukan proses *key* yang sama dengan sebelumnya.

2.1.2 *Flowchart Generate Random Number*

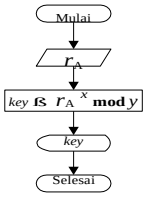


Gambar 2.3 *Flowchart Generate Random Number*

Proses *Generate Random Number* sebagai berikut :

1. Input data teks.
2. Input *true random.*
3. Kemudian didapatkan *seed* dari *true random*.
4. Kemudian s*eed* difungsikan dengan BBS *(Blum-Blum Shub),* maka akan dihasilkan *Random Number*.

2.1.3 *Flowchart Generating Key*

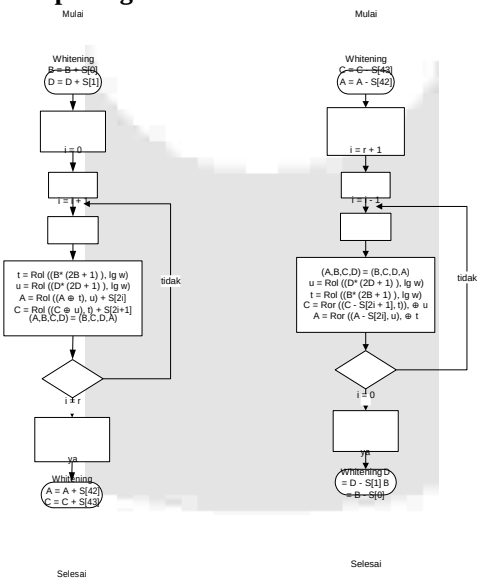


Gambar 2.4 *Flowchart Generating Key*

Proses *Generating Key* sebagai berikut :

1. *Random Number* yang didapatkan dari proses *Generate Random Number*, dimana *x* dan *y* saling relatif prima yakni GCD(*x*, *y*) = 1.
2. Dari hasil *Random Number* kemudian dipangkatkan dengan *x* lalu dioperasikan secara modulus dengan *y*, maka dihasilkan *key* yang akan digunakan dalam operasi enkripsi dengan metode RC6 yang sudah dimodifikasi.

2.1.4 *Flowchart* Enkripsi dan Dekripsi Algoritma RC6[6]



Gambar 2.5 *Flowchart* Enkripsi dan Dekripsi Algoritma RC6 [7]

Proses Enkripsi RC6 sebagai berikut :

1. Dilakukan proses *whitening* awal, nilai B akan dijumlahkan dengan S[0] dan nilai D dijumlahkan dengan S[*i*].
2. Dilakukan iterasi dari $i = 0$ sampai iterasi $i = i + 1$ (sampai iterasi selanjutnya.
3. Kemudian setiap iterasi mengikuti aturan sebagai berikut, nilai B dimasukan ke dalam fungsi *f*, yang didefinisikan sebagai $f(x) = x(2x+1)$, kemudian diputar kekiri sejauh lg-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C. Nilai t juga digunakan sebagai acuan bagi C untuk memutar nilainya kekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan proses pemutaran kekiri.
4. Dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42] dan nilai C dijumlahkan dengan S[43].
5. Kemudian sub kunci S[2*i*] pada iterasi dijumlahkan dengan A, dan sub kunci S[2*i*+1] dijumlahkan dengan C. keempat bagian dari blok kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempatkan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai (asli) D ditempatkan pad C. Demikian iterasi tersebut akan terus berlangsung hingga *r* kali.

Proses Dekripsi RC6

Proses dekripsi *ciphertext* pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses *whitening*, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses *whitening* setelah iterasi terakhir diterapkan sebelum iteasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses *whitening* sebelum iterasi pertama digunakan pada *whitening* setelah iterasi terkahir. Sehingga, untuk melakukan dekripsi yaitu dengan menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.

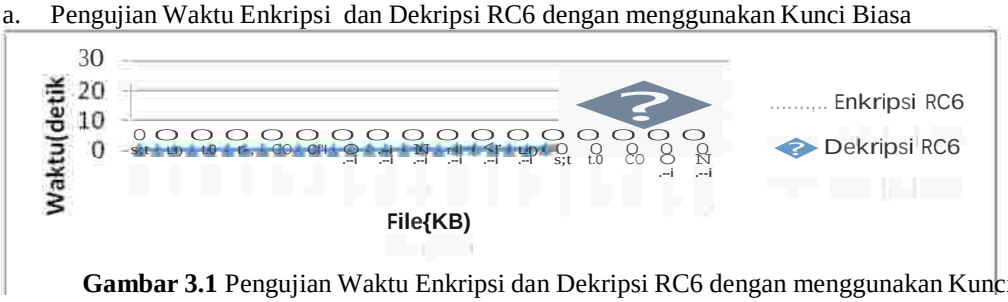
3. Pengujian Dan Analisis

3.1 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 7 karakter dengan besar *file* yang berbeda dan besar *file* yang sama yaitu 550KB dengan panjang kunci yang berbeda. Teknik pengukuran waktu proses enkripsi dan dekripsi dengan menggunakan satuan *nanosecond* yaitu dengan dibagi 1000000000 dan hasilnya dalam satuan *second*.

3.1.1 Perngujian Waktu Enkripsi Dekripsi dengan Kunci Biasa dan Waktu Enkripsi Dekripsi dengan Kunci Modifikasi dengan menggunakan Panjang Kunci Sama dan Besar File 40 KB – 1300 KB

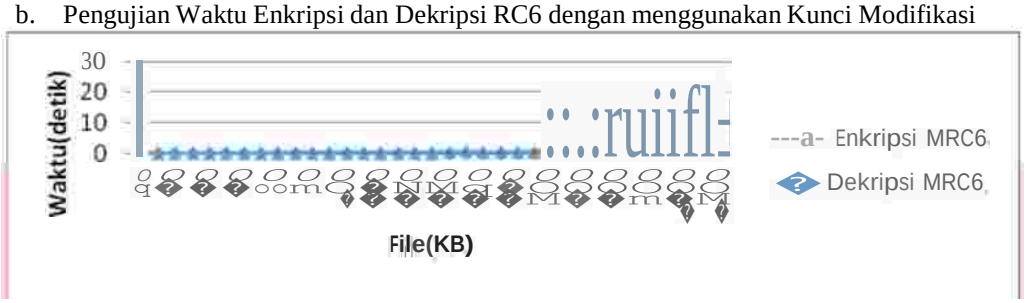
a. Pengujian Waktu Enkripsi dan Dekripsi RC6 dengan menggunakan Kunci Biasa



Gambar 3.1 Pengujian Waktu Enkripsi dan Dekripsi RC6 dengan menggunakan Kunci Biasa

Dari grafik waktu di atas dapat lihat bahwa, semakin besar memori dengan menggunakan kunci biasa yang digunakan maka waktu enkripsi dan dekripsi RC6 dengan menggunakan kunci biasa yang digunakan juga semakin lama.

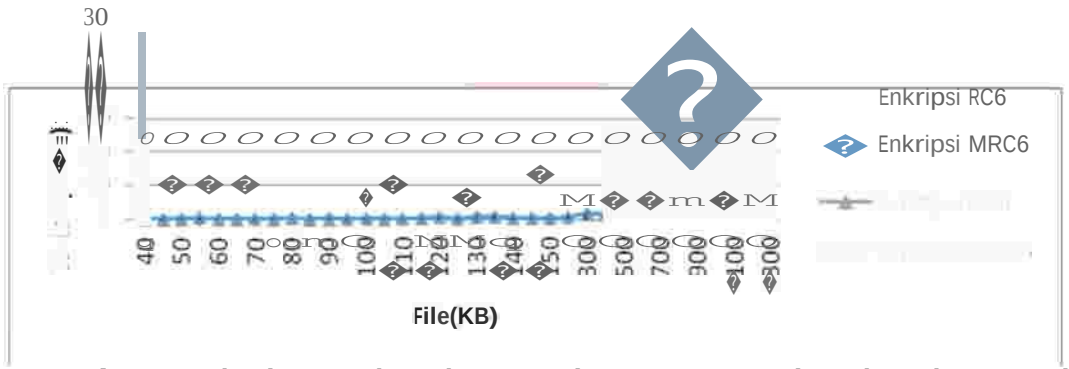
b. Pengujian Waktu Enkripsi dan Dekripsi RC6 dengan menggunakan Kunci Modifikasi



Gambar 3.2 Pengujian Waktu Enkripsi dan Dekripsi RC6 dengan menggunakan Kunci Modifikasi

Dari grafik waktu diatas dapat lihat bahwa, semakin besar memori dengan menggunakan kunci modifikasi yang digunakan maka waktu enkripsi dan dekripsi RC6 dengan menggunakan kunci modififkasi yang digunakan juga semakin lama.

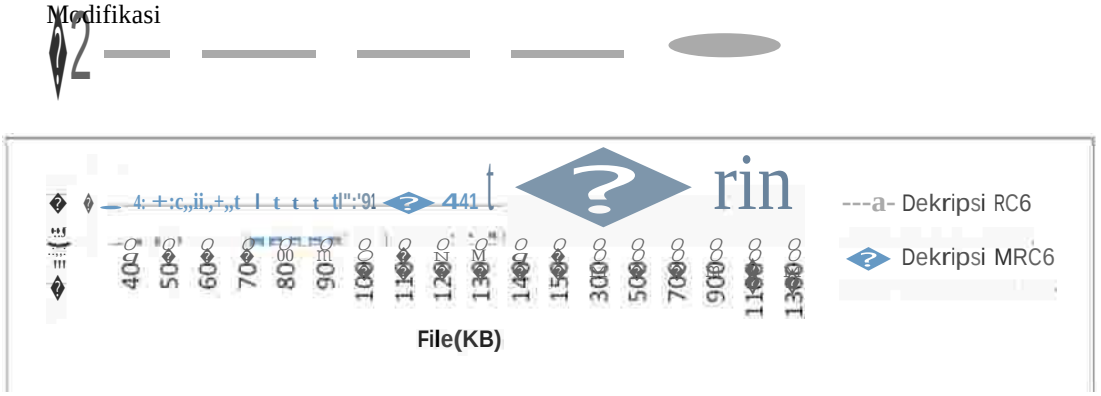
c. Peebandingan Waktu Enkripsi RC6 dengan Kunci Biasa dan Waktu Enkripsi RC6 dengan Kunci Modifikasi



Gambar 3.3 Perbandingan Waktu Enkripsi RC6 dengan Kunci Biasa danWaktu Enkripsi RC6 dengan Kunci Modifikasi

Dari grafik waktu di atas dapat dilihat bahwa, ketika besar file 1300 KB, waktu enkripsi RC6 dengan kunci biasa yakni 26,681 detik dan waktu enkripsi RC6 dengan kunci modifikasi yakni 22.84 detik. Sehingga dapat disimpulkan bahwa waktu enkripsi RC6 dengan kunci biasa lebih lama dibandingkan waktu enkripsi RC6 dengan kunci modifikasi.

d. Perbandingan Waktu Dekripsi RC6 dengan Kunci Biasa dan Dekripsi RC6 dengan Kunci Modifikasi

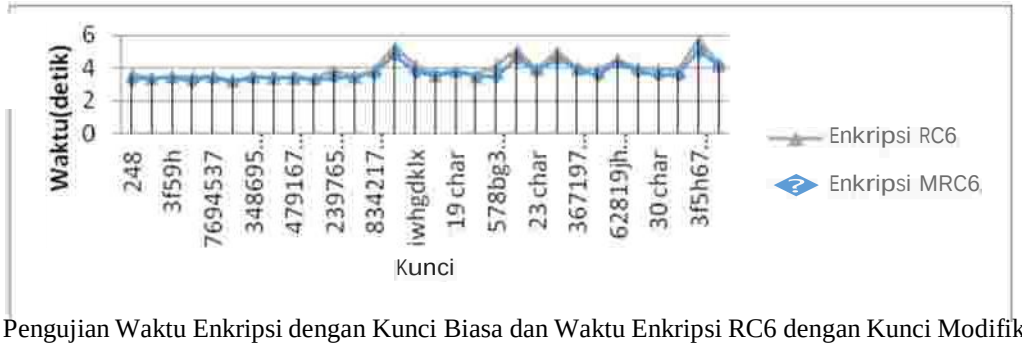


Gambar 3.4 Perbandingan Waktu Dekripsi RC6 dengan Kunci Biasa dan Waktu Dekripsi RC6 dengan Kunci Modifikasi

Dari grafik waktu di atas dapat dilihat bahwa, ketika besar file 1200 KB, waktu dekripsi RC6 dengan kunci biasa yakni 1.274 detik dan waktu dekripsi RC6 dengan kunci modifikasi yakni 1.187 detik. Sehingga dapat disimpulkan bahwa waktu dekripsi RC6 dengan kunci biasa lebih lama dibandingkan waktu dekripsi RC6 dengan kunci modifikasi.

3.1.2 Perbandingan Waktu Enkripsi Dekripsi dengan Kunci Biasa dan Waktu Enkripsi Dekripsi dengan Kunci Modifikasi dengan menggunakan Panjang Kunci yang Berbeda dan Besar File 550 KB

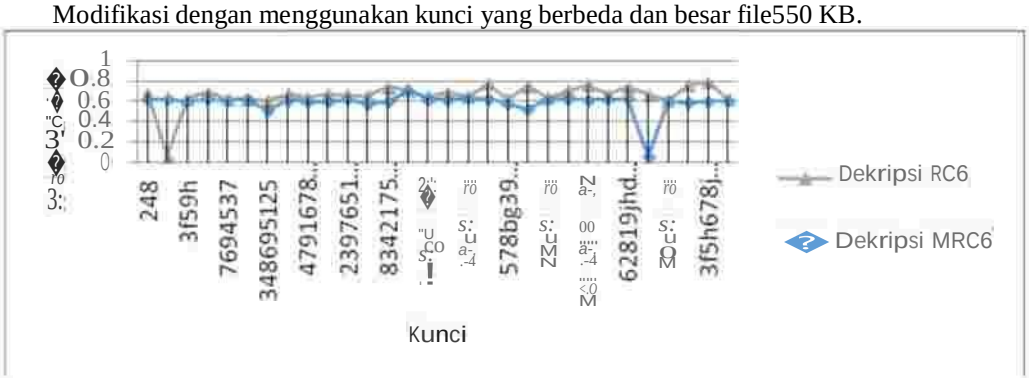
a. Perbandingan Waktu Enkripsi RC6 dengan Kunci Biasa dan Enkripsi RC6 dengan Kunci Modifikasi.



Gambar 3.5 Pengujian Waktu Enkripsi dengan Kunci Biasa dan Waktu Enkripsi RC6 dengan Kunci Modifikasi

Dari grafik di atas dapat dilihat bahwa perubahan waktu untuk melakukan proses enkripsi RC6 dengan kunci biasa dan enkripsi RC6 dengan kunci modifikasi tidak terlalu besar dengan menggunakan panjang kunci yang berbeda. Waktu rata-rata yang diperlukan untuk proses enkripsi RC6dengan kunci biasa adalah 3.94939 detik dan rata-rata waktu yang diperlukan untuk proses enkripsi RC6 dengan kunci modifikasi adalah 3.7265 detik. Sehingga dapat disimpulkan bahwa waktu proses enkripsi RC6 dengan kunci biasa lebih lama daripada waktu proses enkripsi RC6 dengan kunci modifikasi.

b. Perbandingan Waktu dekripsi RC6 dengan kunci Biasa dan Waktu dekripsi RC6 dengan Kunci Modifikasi dengan menggunakan kunci yang berbeda dan besar file550 KB.



Gambar 3.6 Perbandingan Waktu dekripsi RC6 dengan Kunci Biasa dan Waktu dekripsi RC6 dengan Kunci Modifikasi dengan menggunakan kunci yang berbeda dan besar file 550 KB

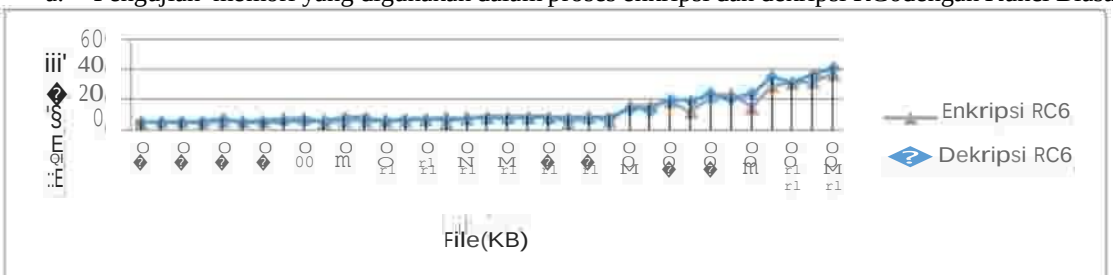
Dari grafik di atas dapat dilihat bahwa perubahan waktu untuk melakukan proses dekripsi RC6 dengan kunci biasa dan dekripsi RC6 dengan kunci modifikasi tidak terlalu besar dengan menggunakan panjang kunci yang berbeda. Waktu rata-rata yang digunakan untuk proses dekripsi RC6 dengan kunci biasa adalah 0.65880 detik dan waktu yang digunakan untuk proses dekripsi RC6 dengan kunci modifikasi adalah 0.590012 detik. Sehingga dapat disimpulkan bahwa waktu proses dekripsi RC6 dengan kunci biasa lebih lama daripada waktu proses dekripsi RC6 dengan kunci modifikasi.

3.2 Pengujian Memori Enkripsi dan Dekripsi

Pengujian ini dilakukan dengan mengukur memori yang digunakan untuk proses enkripsi ketika *file* dikirim ke penerima sedangkan memori proses dekripsi dilakukan ketika pesan diterima. Teknik pengukuran memori proses enkripsi dan dekripsi adalah total memori – memori yang kosong dan hasilnya dalam satuan MB.

3.2.1 Perngujian Memori Enkripsi Dekripsi dengan Kunci Biasa dengan menggunakan Pnjang Kunci Sama dan Besar File 40 KB – 1300 KB

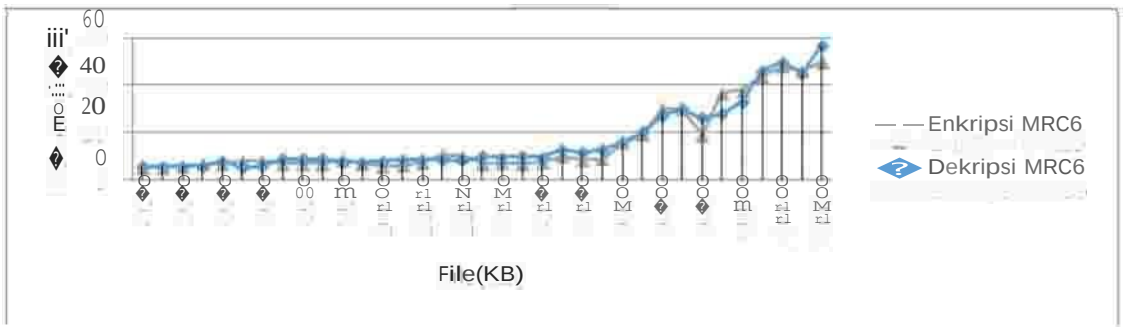
a. Pengujian memori yang digunakan dalam proses enkripsi dan dekripsi RC6dengan Kunci Biasa.



Gambar 3.7 Pengujian memori yang digunakan dalam proses enkripsi dan dekripsi RC6 dengan Kunci Biasa

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang digunakan dengan menggunakan kunci biasa pada algoritma RC6, tetapi masih ada pencilan .

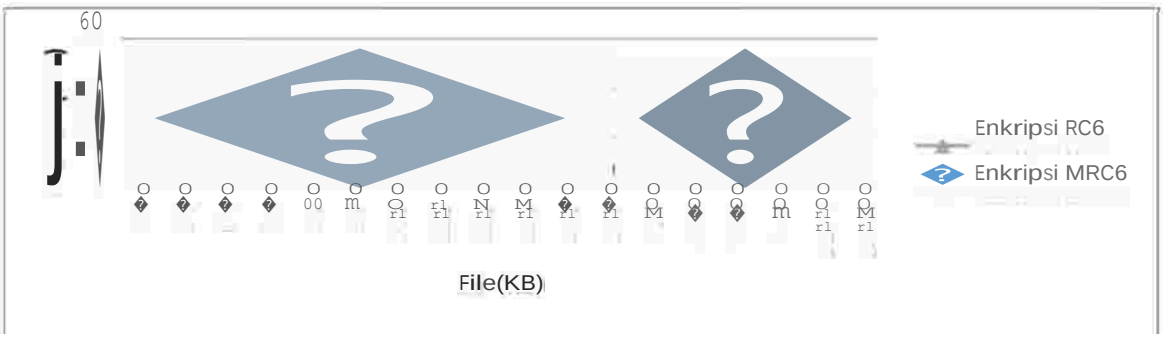
b. Pengujian memori yang digunakan dalam proses enkripsi dan dekripsi RC6 dengan Kunci Modifikasi.



Gambar 3.8 Pengujian memori yang digunakan dalam proses enkripsi dan dekripsi RC6 dengan Kunci Modifikasi

Dari grafik di atas dapat dilihat bahwa semakin besar file yang digunakan maka semakin besar memori yang digunakan dengan menggunakan kunci modifikasi, tetapi masih ada pencilan.

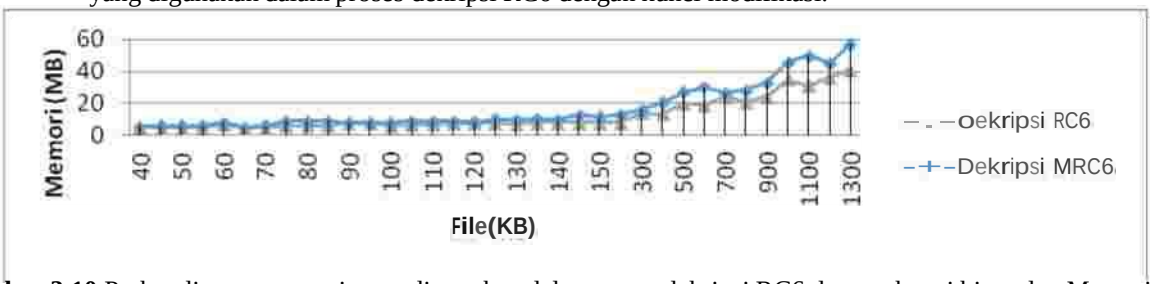
c. Perbandingan memori yang digunakan dalam proses enkripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses enkripsi RC6 dengan Kunci Modifikasi.



Gambar 3.9 Perbandingan memori yang digunakan dalam proses enkripsi RC6 dengan Kunci Biasa dan Memori yang digunakan dalam proses enkripsi RC6 dengan Kunci Modifikasi

Dari grafik di atas dapat dilihat bahwa memori yang digunakan untuk proses enkripsi RC6dengan kunci biasa lebih sedikit daripada memori yang digunakan untuk proses enkripsi RC6 dengan kunci modifikasi.

d. Perbandingan memori yang digunakan dalam proses dekripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses dekripsi RC6 dengan kunci modifikasi.

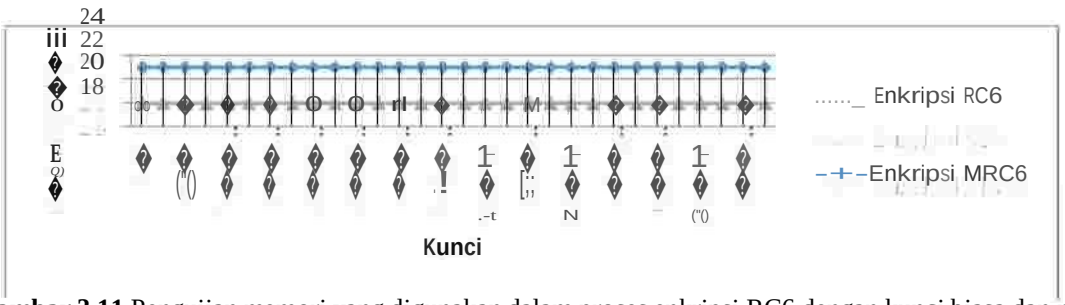


Gambar 3.10 Perbandingan memori yang digunakan dalam proes dekripsi RC6 dengan kunci biasa dan Memori yang digunakan dalam proes dekripsi RC6 dengan kunci modifikasi

Dari grafik di atas dapat dilihat bahwa memori yang digunakan dalam proses dekripsi RC6 dengan kunci biasa lebih sedikit daripada memori yang digunakan dalam proses dekripsi RC6 dengan kunci modifikasi.

3.2.2 Pengujian Memori Enkripsi Dekripsi dengan Kunci Biasa dan Memori Enkripsi Dekripsi dengan Kunci Modifikasi dengan menggunakan Panjang Kunci yang Berbeda dan Besar File 550 KB

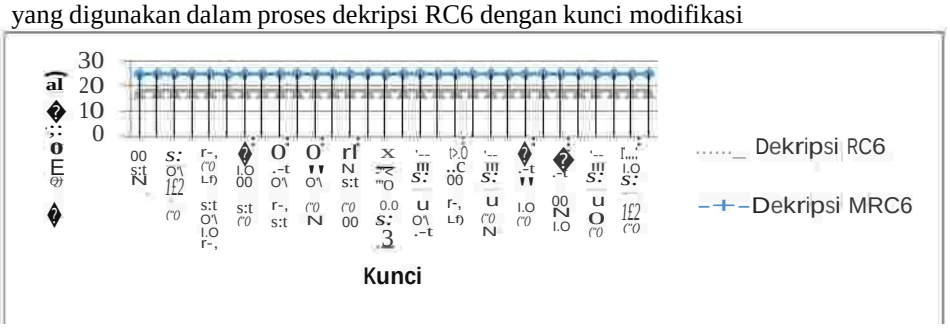
a. Perbandingan memori yang digunakan dalam proses enkripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses enkripsi RC6 dengan kunci modifikasi



Gambar 3.11 Pengujian memori yang digunakan dalam proses enkripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses enkripsi RC6 dengan kunci modifikasi

Dari grafik di atas dapat dilihat bahwa ketika menggunakan kunci yang berbeda dan file sama, proses enkripsi RC6 dengan kunci biasa menggunakan memori yang sama yakni 20 MB dan proses enkripsi RC6 dengan kunci modifikasi menggunakan memori yang sama juga yakni 23 MB. Sehingga dapat disimpulkan bahwa memori yang digunakan dalam proses enkripsi RC6 dengan kunci biasa lebih sedikit daripada memori yang digunakan dalam proses enkripsi RC6 dengan kunci modifikasi.

b. Perbandingan memori yang digunakan dalam proses dekripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses dekripsi RC6 dengan kunci modifikasi



Gambar 3.12 Perbandingan memori yang digunakan dalam proses dekripsi RC6 dengan kunci biasa dan memori yang digunakan dalam proses dekripsi RC6 dengan kunci modifikasi

Dari grafik di atas dapat dilihat bahwa ketika menggunakan kunci yang berbeda dan file sama, proses dekripsi RC6 dengan kunci biasa menggunakan memori yang sama yakni 18 MB dan proses dekripsi RC6 dengan kunci modifikasi menggunaan memori yang sama juga yakni 25 MB. Sehingga dapat disimpulkan bahwa memori yang digunakan dalam proses dekripsi RC6 dengan kunci biasa lebih sedikit daripada memori yang digunakan dalam proses dekripsi RC6 dengan kunci modifikasi.

3.3 Pengujian *Avalanche Effect*

Pada kriptografi, hasil yang diberikan sangat unik, berbeda dari data yang menjadi masukan dari proses

tersebut. Sedikit perubahan pada data masukan dapat memberikan perubahan yang signifikan pada hasil

proses kriptografi, dan perubahan tersebut dinamakan *avalanche effect*. Semakin besar perubahan yang terjadi, semakin baik performansi dari algoritma kriptografi tersebut.

a. Pengujian *Avalanche Effect* RC6 ketika *plaintext*-nya diubah 1 karakter:
Plaintext 1 : 00000000000000000000000000
Plaintext 2 : 00000000000000000000000001
Kunci : 00000000000000000000000000
Ciphertext 1 : 11010111111001001010001010011011
Ciphertext 2 : 11100011101001 0 0001100111011100
Avalanche Effect =

Dari hasil pengujian *Avalanche Effect* RC6 ketika *plaintext*-nya diubah satu karakter, hasilnya cukup baik yakni berkisar antara 46.875% sampai 65.625% dan hasil pengujian *Avalanche Effect* MRC6 ketika *plaintext*-nya diubah satu karakter, hasilnya cukup baik yakni berkisar antara 43.75% sampai 62.5% .

b. Pengujian *Avalanche Effect* RC6 ketika kuncinya diubah 1 karakter :
Plaintext : 00000000000000000000000000
Kunci 1 : 00000000000000000000000000
Kunci 2 : 00000000000000000000000001
Ciphertext 1 : 10001110111101101100101011010100
Ciphertext 2 : 01101011100100110010000110101011
Avalanche Effect =

Dari hasil pengujian *Avalanche Effect* RC6 ketika kuncinya diubah satu karakter, hasilnya cukup baik yakni berkisar antara 50% sampai 68.75% dan hasil pengujian *Avalanche Effect* MRC6 ketika kuncinya diubah satu karakter, hasilnya cukup baik yakni berkisar antara 50% sampai 62.5%.

4. Kesimpulan dan Saran

4.1 Kesimpulan

1. Waktu yang diperlukan untuk proses enkripsi dan dekripsi dengan menggunakan kunci biasa pada algoritma RC6 lebih lama jika dibandingkan dengan menggunakan kunci yang sudah modifikasi pada algoritma RC6.
2. Memori yang digunakan dalam proses enkripsi dan dekripsi dengan menggunakan kunci biasa algoritma RC6 lebih kecil jika dibandingkan dengan menggunakan kunci yang sudah dimodifikasi pada algoritma RC6.
3. Panjang kunci yang berbeda dan file yang sama tidak terlalu berpengaruh terhadap waktu dan memori yang digunakan pada proses enkripsi dan dekripsi, baik menggunakan kunci biasa pada algoritma RC6 ataupun dengan menggunakan kunci yang sudah dimodifikasi pada algoritma RC6.
4. Nilai *Avalanche Effect* yang dihasilkan dengan menggunakan kunci biasa maupun dengan kunci modifikasi memiliki performansi yang sangat baik, sehingga akan sulit untuk dipecahkan ketika kunci yang digunakan tidak diketahui dan hasil yang diberikan sangat unik

4.2 Saran

Dari perancangan sistem yang telah dibangun tentunya masih perlu pengembangan agar bisa lebih baik dan dapat diimplementasikan. Saran untuk melakukan pengembangan pada perancangan sistem ini adalah sebagai berikut.

1. Proses modifikasi dilakukan pada algoritma keseluruhan dan bukan hanya kuncinya saja.
2. Proses enkripsi dan dekripsi dengan kunci modifikasi dilakukan menggunakan dua computer yang berbeda.
3. Masukan *file*-nya atau *plaintext*-nya berbeda, misalnya file *.doc dan *.pdf.

5. Referensi

[1] Abdurohman, Maman. 2002. *Analisis Performansi Algoritma Kriptografi RC6*. Teknologi Informasi, Departemen Teknik Elektro, Institut Teknologi Bandung.
[2] Ariyus, Doni. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta: ANDI.
[3] *Blum Blum Shub Cryptosystem and Generator*. [Online]. Tersedia : http://diamond.boisestate.edu/~liljanab/ISAS/course_materials/BBSpresentation.pdf [9 Januari 2015].
[4] Nazernasrisamaaja. (2011). Kriptografi Simetris,Asimetris, dan Hybrid, Jakarta.
[5] Nurwahidin,Arif. 28 November 2011. "Pengembangan Algoritma Kriptografi DES dengan 112 kunci internal". p 1-2.

[6] Prayudi, yudi dan halik, Idham. 2005. "Studi dan Analisis Algoritma *Rivest Code* 6 Dalam Enkripsi/Dekripsi Data".ISBN = 979-756-061-6.
[7] *Random Number Generator.* [Online]. Tersedia : http://elib.unikom.ac.id/files/disk1/301/jbptunikompp-gdl-deviagungs-15023-3-babiil-i.pdf. [10 Januari 2015].
[8] *Random Number Generator and Pseudo-Random Number Generator*. [Online]. Tersedia : http://elib.unikom.ac.id/files/disk1/301/jbptunikompp-gdl-deviagungs-15023-3-babiil-i.pdf. [9 Januari 2015].
[9] Septiarini Anindita, Hamdani. 1 Februari. (2011). "Sistem Kriptografi Untuk *Text Message* Menggunakan Metode Affine ". p 1-2.