Pembuatan Aplikasi Pengamanan Data dengan Metode MARS

Elizabeth Nathania W[1], Gregorius Satia B[2], Kristo Radion P[3]
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jln. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031)-2983455, Fax. (031)-8417658
liza_1805@hotmail.com[1], greg@petra.ac.id[2], kristo@petra.ac.id[3]

ABSTRAK

Di era modern ini teknologi berkembang pesat dan semakin canggih, terlebih teknologi komunikasi. Informasi dapat disebarkan tanpa harus bertatap muka satu sama lain. Tidak dipungkiri bahwa informasi bersifat rahasia pun dikirim melalui internet. Melalui beragam jenis informasi inilah berkembang teknik-teknik untuk merusak atau pun mempertahankan keamanan dari penyebaran informasi di internet.

Pengamanan data dengan menggunakan kriptografi sangat dibutuhkan untuk pengiriman data yang bersifat rahasia agar tidak semua orang bisa mengaksesnya. Proses yang dilakukan dalam penelitian ini adalah enkripsi pesan dengan metode MARS. *Input* dalam aplikasi yang dirancang ini berupa *plaintext* atau *file* pesan yang hendak di enkripsi, kemudian *key* untuk mengenkripsi pesan. Aplikasi ini dibuat dengan bahasa pemrograman Java menggunakan NetBeans IDE 7.4.

Hasil pengujian menunjukkan bahwa enkripsi dengan metode MARS dapat dilakukan pada *file* jenis apapun (contoh: *file text*, *file* lagu atau audio, *file* video atau *movie*, dan lain-lain). *File* dapat didekripsi kembali seperti semula jika memasukkan *password* yang sama dengan *password* untuk enkripsi.

Kata Kunci: Pengamanan Data, Kriptografi, MARS.

ABSTRACT

Nowadays, technology is growing rapidly and becoming more sophisticated, especially communications technology. Information can be deployed without having to meet each other. No doubt that any confidential information sent over the Internet. Through various types of information is developing techniques for damage or maintaining the security of the dissemination of information on the internet.

Securing data by using cryptography is needed for sending confidential data so that not everyone can access it. Process undertaken in this study is encrypted messages with MARS method. Input in an application that is designed in the form of plaintext messages or files to be encrypted, then the key to encrypt the message. This application is made with the Java programming language using NetBeans IDE 7.4.

The test results indicate that the encryption with MARS method can be performed on any type of file (example: text file, song or audio file, video file or movie, etc.). Files can be decrypted back to normal if the password is the same as the password for the encryption.

Keyword: *Data Security, Cryptography, MARS.*

1. PENDAHULUAN

Di era modern ini teknologi berkembang pesat dan semakin canggih, terlebih teknologi komunikasi. Hanya dalam hitungan detik suatu informasi dapat tersebar di seluruh dunia melalui internet. Informasi dapat disebarkan tanpa harus bertatap muka satu sama lain. Tidak dipungkiri bahwa informasi bersifat rahasia pun dikirim melalui internet. Melalui beragam jenis informasi inilah berkembang teknik-teknik untuk merusak atau pun mempertahankan keamanan dari penyebaran informasi di internet [5].

Adanya teknik kriptografi atau enkripsi pesan berguna untuk keamanan pesan agar tidak dengan mudah dibaca oleh orang-orang yang tidak berkepentingan. Pada penelitian ini, peneliti akan mengembangkan aplikasi untuk mengamankan data dengan metode enkripsi MARS.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi awalnya diartikan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun sekarang, kriptografi diartikan sebagai ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi atau data. Tidak saja hanya tentang penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan data [6].

Di bawah ini adalah layanan keamanan data berdasarkan dokumen X.800 [7]:

a. ***Authentication***. Pengirim pesan harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak dapat berpura-pura menjadi orang lain.

b. ***Access Control***. Menghindari agar pesan tidak dapat dibaca oleh orang-orang yang tidak berkepentingan.

c. ***Data Confidentiality***. Memproteksi transmisi data dari gangguan yang tidak diinginkan.

d. ***Data Integrity***. Penerima pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi ketika sedang dalam proses transmisi data.

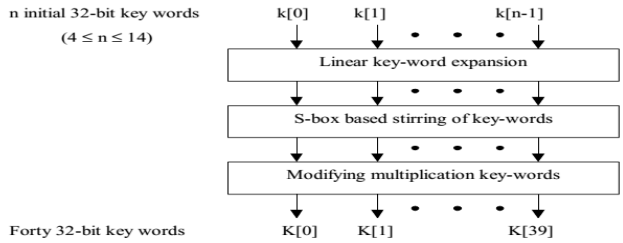
e. ***Non-Repudiation***. Pengirim pesan harus tidak dapat menyangkal pesan yang dia kirimkan.

2.2 MARS

MARS adalah salah satu kandidat *Advanced Encryption Standard* (AES) yang masuk dalam 5 besar pada AES *Competition*. Algoritma ini adalah *shared-key cipher block* dengan ukuran blok 128 bit dan ukuran kunci bervariasi antara 128 sampai 448 bit [4]. Sebelum enkripsi blok dimulai, tiap blok dibagi menjadi empat *word* data yang tiap *word* terdiri dari 32 bit data. Keseluruhan operasi internal terjadi pada tiap satu *word* data. Langkah utama dari metode ini adalah proses ekspansi kunci, proses enkripsi, dan proses dekripsi.

2.2.1 Proses Ekspansi Kunci

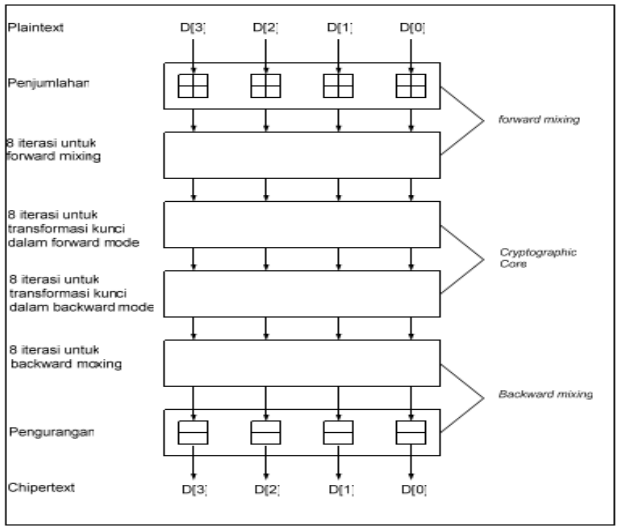
Proses ekspansi kunci ini membuat panjang kunci yang akan digunakan untuk enkripsi atau dekripsi awalnya antara 128 sampai 448 bit atau 4 *word* sampai 14 *word* menjadi 40 *word*. Proses ini terdiri dari tiga langkah seperti pada Gambar 1. Langkah pertama adalah "*linear expansion*". Pada langkah ini dilakukan ekspansi kunci asli yang panjangnya antara 4 sampai 14 *word* menjadi 40 *word* menggunakan transformasi linear. Langkah kedua adalah "*S-box based key stirring*". Dilakukan pengacakan kunci yang sudah diekspansi pada langkah kedua ini dengan menggunakan 7 putaran feistel tipe 1. Langkah kedua ini tujuannya adalah untuk memutuskan relasi linear pada kunci. Langkah ketiga adalah "*multiplication key-word modifying*". Pada langkah ini dilakukan untuk melakukan perkalian (*multiplication*) jika dibutuhkan.[1]



Gambar 1. Proses Ekspansi Kunci [1]

2.2.2 Proses Enkripsi

Proses enkripsi MARS dapat terlihat pada Gambar 2.



Gambar 2. Proses Enkripsi MARS [4]

Proses enkripsi MARS dibagi menjadi 3 tahap yaitu:

- *Forward Mixing*. Tahap ini berfungsi untuk mencegah serangan terhadap *chosen plaintext*. Terdiri dari penambahan sub kunci pada setiap *word* data, diikuti dengan delapan iterasi *mixing* tipe-3 feitsel (dalam *forward* mode) dengan berbasis S-box.
- *Cryptographic core* dan *cipher*, terdiri dari enam belas iterasi tranformasi kunci tipe-3 feistel. Untuk menjamin bahwa proses enkripsi dan dekripsi mempunyai kekuatan yang sama, delapan iterasi pertama ditunjukkan dalam "*forward* mode" dan delapan iterasi terakhir ditunjukkan dalam "*backward* mode".
- *Backward mixing*, berfungsi untuk melindungi serangan kembali terhadap *chosen chipertext*. Tahap ini merupakan *invers* dari tahap pertama, terdiri dari delapan iterasi *mixing* tipe-3 feistel (dalam *backward* mode) dengan berbasis S-box, diikuti dengan pengurangan sub kunci dari *word* data. Hasil pengurangan inilah yang disebut dengan *ciphertext*.

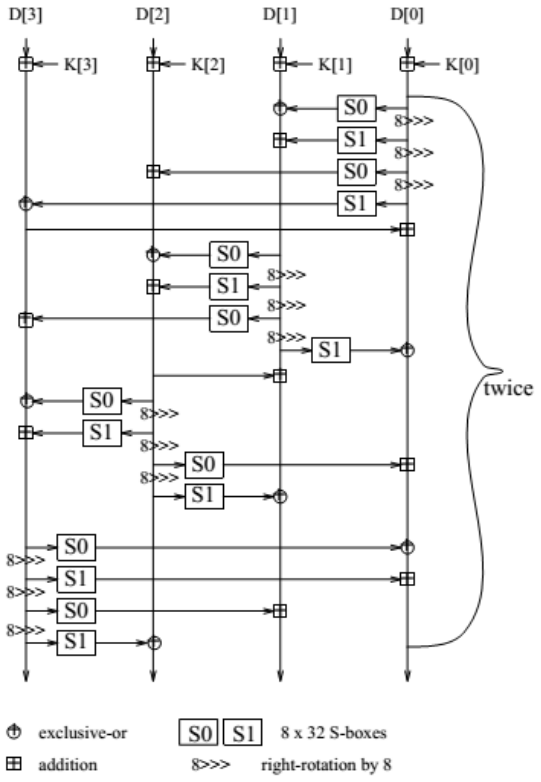
2.2.2.1 Forward Mixing dan Backward Mixing

Langkah *forward* dan *backward mixing* (atau *wrapper layers*) sebenarnya adalah kebalikan satu dengan yang lain. Langkah *forward mixing* dimulai dengan menambah *key-word* pada data-*word*, lalu diputar sebanyak delapan kali berdasarkan S-box, kemudian dilakukan *unkeyed mixing*. Langkah *backward mixing* mempunyai delapan putaran yang berkebalikan dengan *forward mixing*, diikuti dengan *key-subtraction*.[1]

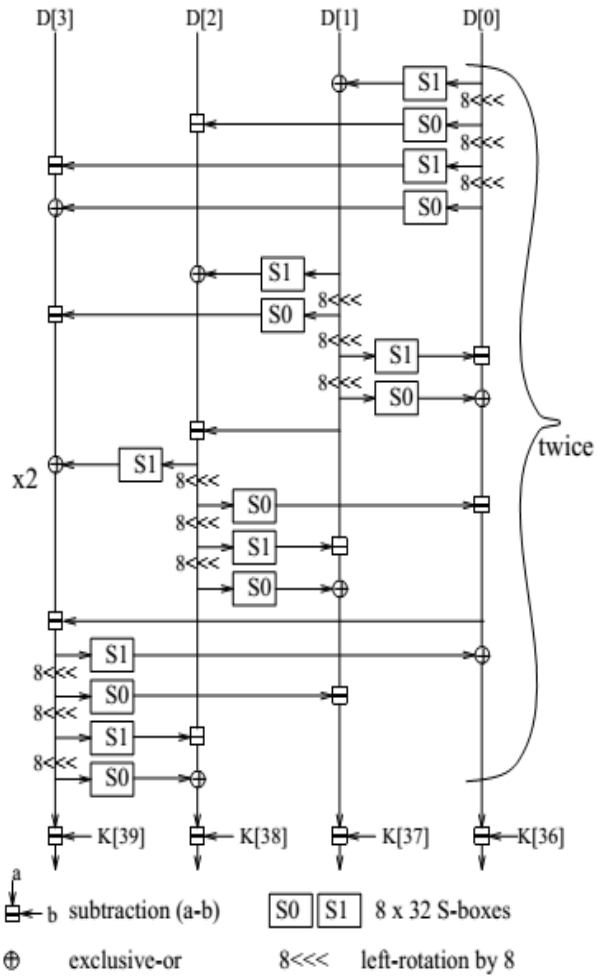
Pada setiap putaran di *forward* dan *backward mixing*, satu data-*word* (*source word*) digunakan untuk memodifikasi 3 data *word* yang lain (*target words*). Empat *bytes* dari *source word* dilihat sebagai index dari dua S-boxes, S0 dan S1, tiap S-boxes terdiri dari 256 32 bit *words*, dan *input* S-box yang sesuai akan dilakukan operasi xor atau penjumlahan atau pengurangan pada 3 data *word* yang lain.[1]

Dalam *forward* dan *backward mixing*, 4 *words* akan dirotasi tiap putarannya, jadi yang sebelumnya menjadi *target word* yang pertama akan menjadi *source word* pada putaran berikutnya, *target word* yang kedua akan menjadi *target word* yang pertama, *target word* yang ketiga akan menjadi *target word* yang kedua, dan *source word* akan menjadi *target word* yang ketiga. [1]

Sebagai tambahan, pada langkah *forwad mixing* dilakukan penjumlahan antara *target word* yang ketiga dengan *source word* setelah putaran pertama dan kelima, dan penjumlahan antara *target word* yang pertama dengan *source word* setelah putaran kedua dan keenam. Pada langkah *backward mixing* dilakukan pengurangan antara *target word* yang pertama dengan *source word* sebelum putaran keempat dan kedepalan, dan pengurangan antara *target word* yang ketiga dengan *source word* sebelum putaran ketiga dan ketujuh. Alasannya adalah untuk menambah ekstra *mixing* untuk mengurangi kemungkinan terjadinya *differential attacks* pada langkah *mixing*. Untuk lebih jelas dapat dilihat pada Gambar 3 dan Gambar 4.[1]



Gambar 3. Struktur *Forward Mixing*[1]



Gambar 4. Struktur *Backward Mixing*[1]

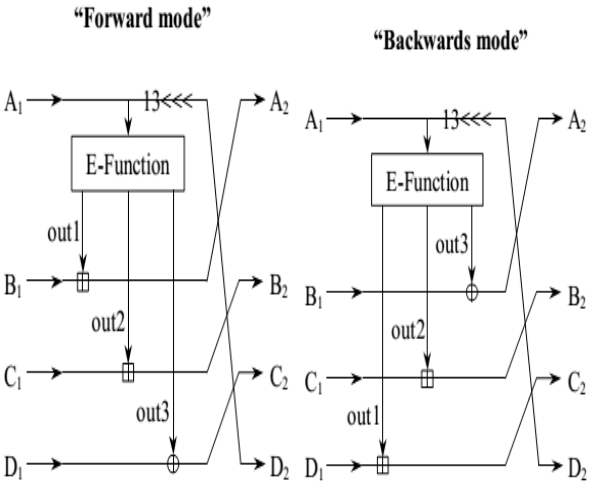
2.2.2.2 Cryptographic Core

Cryptographic core dari MARS menggunakan feistel tipe 3 yang terdiri dari enam belas putaran. Setiap putaran digunakan "E-*function*" (E untuk *expansion*) yang menggunakan *input* satu data *word* dan menghasilkan output tiga data *word*. Struktur dari Feistel dapat dilihat pada Gambar 5 dan struktur E-*function* dapat dilihat pada Gambar 6. *Input*an pada E-*function* di setiap putaran berupa satu data *word* dan *output*nya berupa tiga data *word* yang dilakukan operasi penjumlahan atau xor. Sebagai tambahan, *source word* dirotasi sebanyak 13 posisi ke kiri.

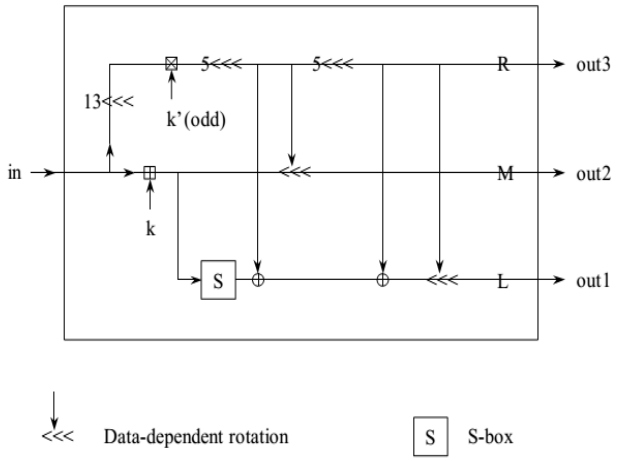
Output dari E-*function* digunakan dengan urutan yang berbeda pada delapan putaran pertama (*forward mode*) dan delapan putaran terakhir (*backwards mode*). Pada delapan putaran pertama dilakukan penjumlahan antara *output* E-*function* yang pertama dan kedua dengan *target words* yang pertama dan kedua, kemudian dilakukan xor pada *output* E-*function* yang ketiga dengan *target word* yang ketiga. Pada delapan putaran terakhir, dilakukan penjumlahan antara *output* E-*function* yang pertama dan kedua dengan *target words* yang ketiga dan kedua, sedangkan *output* E-*function* yang ketiga dilakukan xor dengan *target word* yang pertama.

2.2.3 Proses Dekripsi

Dekripsi adalah kebalikan dari proses enkripsi. Urutan proses yang dilakukan adalah *backward mixing, cryptographic core, forward mixing*. Cara ekspansi kunci sama dengan penjelasan pada poin 2.2.1. Gambar 5 dan Gambar 6 merupakan struktur Feistel dan E-*Function.*



Gambar 5. Struktur Feistel[1]



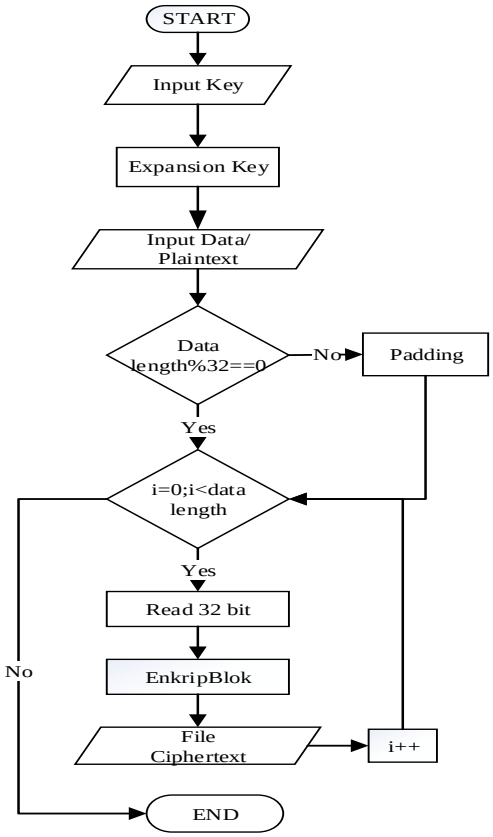
Gambar 6. Struktur E-*Function*[1]

3. DESAIN SISTEM

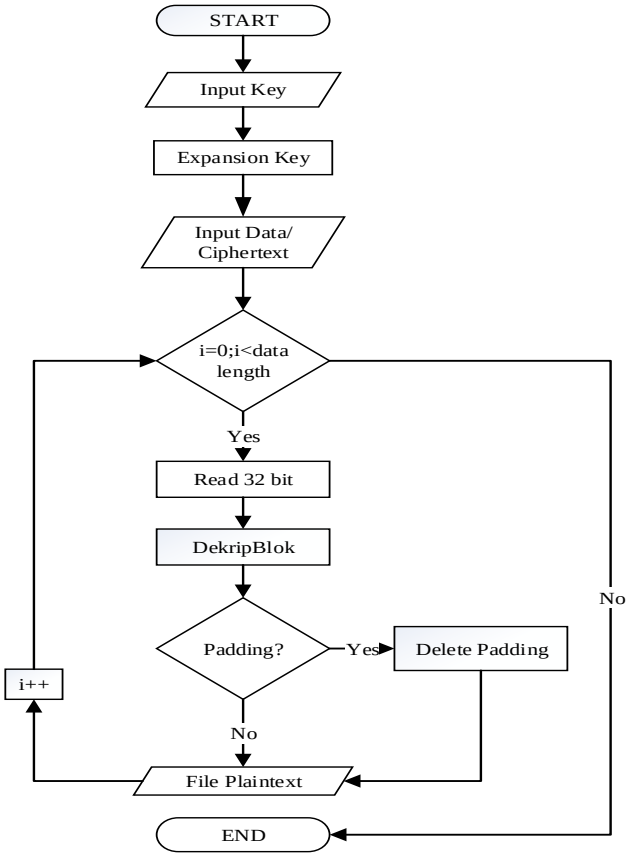
Pada aplikasi ini ada 2 proses yang akan dilakukan, yaitu proses enkripsi dan proses dekripsi. Enkripsi adalah proses untuk mengenkripsi data dari *file plaintext* menjadi *file ciphertext*, sedangkan dekripsi adalah proses untuk mengembalikan data dari *file ciphertext* menjadi *file* asli yaitu *file plaintext*.

3.1 Garis Besar Metode MARS

Untuk melakukan proses enkripsi dan dekripsi menggunakan metode MARS ada beberapa tahap yang akan dilakukan. Proses enkripsi MARS memerlukan inputan kunci dan *file plaintext*. Inputan kunci ini berupa karakter atau angka. MARS dapat menampung panjang kunci dari 128 bit sampai 448 bit. Setelah itu dilakukan ekspansi kunci seperti yang telah dijelaskan pada poin 2.2.1. Inputan *file plaintext* adalah inputan awal berupa data yang ingin dienkripsi. Format data yang ingin dienkripsi bisa berupa *file text*, *file* gambar, *file* suara, ataupun *file* video. Kemudian dilakukan pengecekan apakah panjang data *mod* 32 sama dengan nol, jika tidak maka data akan di*padding*, jika ya maka data akan memasuki proses enkripsi. Proses enkripsi dilakukan setiap 32 bit data. Detail penjelasan proses enkripsi dapat dilihat pada poin 2.2.2. Setelah pembacaan *file* selesai, di akhir proses enkripsi ini akan menghasilkan output berupa *file ciphertext* yang akan digunakan pada proses selanjutnya. Gambar 7 menggambarkan proses enkripsi dan Gambar 8 menggambarkan proses dekripssi.



Gambar 7. *Flowchart* Metode Enkripsi MARS



Gambar 8. *Flowchart* Metode Dekripsi MARS

Proses dekripsi sebenarnya adalah kebalikan dari proses enkripsi. *Input* yang diperlukan adalah kunci yang sama yang dipakai saat melakukan enkripsi. Setelah itu dilakukan ekspansi kunci seperti yang telah dijelaskan pada poin 2.2.1. *Input* selanjutnya adalah *file ciphertext* yang telah melalui proses enkripsi sebelumnya. *File* ini akan dibaca sebanyak 128 bit per putaran dengan tiap bloknya sama dengan 32 bit, kemudian dilakukan proses dekripsi. Setelah *file* selesai dibaca maka akan di cek apakah *file* tersebut pada proses enkripsi dilakukan penambahan atau *padding*. Jika terdapat *padding* maka *padding*-nya akan dihapus, jika tidak terdapat *padding* maka itulah hasil akhir dari proses dekripsi.

4. IMPLEMENTASI SISTEM

Pada proses enkripsi data, yang dilakukan adalah meng*input*kan *file* pesan yang akan dienkripsi sekaligus yang akan disembunyikan. Kemudian mengisi *key* untuk *password* enkripsi. Panjang *key* minimal 4 karakter dan maksimal 14 karakter. *File* ini akan di enkripsi dengan metode MARS.

Pada proses dekripsi data yang dilakukan adalah menginputkan *file* yang telah di enkripsi sebelumnya atau disebut *ciphertext*. Kemudian meng*input*kan *key* pertama untuk dilakukan dekripsi *file* menggunakan metode MARS.

5. PENGUJIAN SISTEM

5.1 Pengujian Enkripsi dan Dekripsi MARS

Pengujian enkripsi dan dekripsi MARS dilakukan pada bermacam jenis *file* dapat dilihat pada Tabel 1 di bawah ini. Panjang kunci yang digunakan sama yaitu sebesar 64 bit atau 8 karakter.

Tabel 1. Hasil Pengujian Enkripsi dan Dekripsi MARS

Nama *File*	**Besar *file* sebelum enkripsi (*bytes*)**	**Besar *file* setelah enkripsi (*bytes*)**	**Besar *file* setelah dekripsi (*bytes*)**
abcc.rar	411	416	411
Lorem Ipsum.txt	3.114	3.136	3.114
quotes.jpg	76.695	76.704	76.695
Logogram-02.png	161.065	161.088	161.065
Birthday Song.mp3	1.454.947	1.454.976	1.454.947
Buku TA.docx	2.138.988	2.139.008	2.138.988
MVI_0409.MOV	11.449.492	11.449.504	11.449.492
Tari Kasomber.wav	41.223.242	41.223.264	41.223.242
Falling Plates.mp4	56.682.331	56.682.336	56.682.331

Dari hasil pengujian dapat disimpulkan bahwa metode MARS dapat mengenkripsi bermacam-macam jenis *file* dan dapat dikembalikan seperti asalnya setelah didekripsi.

Tabel 2. Pengujian Waktu Enkripsi dan Dekripsi

Nama *File*	Besar *file* asli(*bytes*)	Waktu enkripsi (detik)	Waktu dekripsi (detik)
abcc.rar	411	0.005	0.001
Lorem Ipsum.txt	3.114	0.016	0.01
quotes.jpg	76.695	0.063	0.047

Nama *File*	Besar *file* asli(*bytes*)	Waktu enkripsi (detik)	Waktu dekripsi (detik)
Logogram-02.png	161.065	0.015	0.078
Birthday Song.mp3	1.454.947	0.047	0.047
Buku TA.docx	2.138.988	0.115	0.107
MVI_0409.MOV	11.449.492	0.284	0.321
Tari Kasomber.wav	41.223.242	0.874	0.77
Falling Plates.mp4	56.682.331	0.983	0.983

Dari hasil pengujian lamanya waktu yang dibutuhkan untuk enkripsi dan dekripsi pada Tabel 2, dapat disimpulkan bahwa semakin besar *file* yang akan diproses maka waktu yang dibutuhkan juga semakin lama.

6. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa algoritma enkripsi MARS dapat mengenkripsi bermacam-macam jenis *file* (contoh: *file text*, *file* lagu atau audio, *file* video atau *movie*, dan lain-lain) dan dapat mendekripsi *file* kembali dengan ukuran yang sama. Selain itu, besar *file* yang akan diproses mempengaruhi lamanya waktu yang dibutuhkan. Semakin besar *file* yang akan diproses, maka waktu yang dibutuhkan juga semakin lama.

7. REFERENSI

[1] Coppersmith, D. 1999. *The MARS Encryption Algorithm*. Retrieved May 22, 2014, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.5887&rep=rep1&type=pdf

[2] Cryptographic Competitions. (2014). *AES: the Advanced Encryption Standard*. Retrieved May 22, 2014, from http://competitions.cr.yp.to/aes.html

[3] Djebbar, F., Ayad, B., Meraim, K. A., & Hamam, H. (2014). *Comparative Study of Digital Audio Steganography Techniques.* 4. Retrieved May 22, 2014, from http://biblio.telecom-paristech.fr/cgi-bin/download.cgi?id=12699

[4] Ilmy, M. 2014. *Perbandingan Algoritma Mars Dan Rijndael Dalam Beberapa Mode Operasi Cipher Blok*. Retrieved May 18, 2014, from http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2006-2007/Makalah1/Makalah1-034.pdf.

[5] Rojali. 2013. *Kriptografi dan Steganografi*. Retrieved May 29, 2014, from http://socs.binus.ac.id/2013/07/30/kriptografi-dan-steganografi/

[6] Sadikin, R. 2012. *Kriptografi untuk Keamanan Jaringan*. Yogyakarta: Penerbit ANDI.

[7] Stallings, W. 2011. *Cryptography and Network Security 5th Edition*. USA: Prentice Hall