IMPLEMENTASI ALGORITMA KRIPTOGRAFI AES UNTUK ENKRIPSI DAN DEKRIPSI EMAIL

Ahmad Rosyadi

E-mail: mattady@ymail.com

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

Abstrak

Perkembangan dunia informatika yang sangat pesat saat ini membawa pertumbuhan dunia ke dalam masa teknologi informasi. Karena itulah nilai informasi saat ini sangat penting. Salah satu contohnya adalah menggunakan email. Algoritma kriptografi AES digunakan untuk proses penyandian email. Aplikasi ini menggunakan bahasa pemrograman Java dan Netbeans 7.0 sebagai perangkat lunak. Server mail yang digunakan adalah Google mail dan menggunakan port 465. Kunci yang digunakan menggunakan kunci 128-bit, sehingga hanya ada 10 putaran kunci. Langkah – langkah penelitian yang dilakukan adalah pertama, mengunduh email dari Google server kemudian mengenkripsi pesan tersebut. Kedua, pesan yang telah dienkripsi selanjutnya akan didekripsi untuk membuktikan pesan tersebut masih sama dengan pesan asli sebelum dienkripsi dengan menggunakan kunci yang sama. Hasil penelitian ini adalah suatu aplikasi enkripsi dan dekripsi email dengan menggunakan algoritma kriptografi AES (Rinjdael). Dengan perangkat lunak ini, keamanan dalam mengirim dan menerima email dapat terjamin. Walaupun pesan email bisa diambil orang lain tetapi mereka tetap tidak akan bisa membacanya karena teks tertampil dalam bentuk karakter heksadesimal dan jika dijadikan string maka akan tampil sebagai simbol-simbol yang tidak jelas.

Kata kunci: Kriptografi, Rijndael, Enkripsi dan Dekripsi Email.

Abstract

During development of the informatics technology at this time bring the world into the future growth of information technology. That's why the current information is very important. One example is the use of email. AES cryptographic algorithm used for email encryption process. This application uses the Java programming language and Netbeans 7.0 as software. Mail server used Google mail and using port 465. The key to use is 128-bit key, so there are only 10 round keys. The first steps of research is download email from the Google server then encrypts its message. Second, the encrypted message will then be decrypted to prove the message is still the same as the original message before it is encrypted using the same key. The results of this research is an email encryption and decryption application using a cryptographic algorithm AES (Rinjdael). With this application, security in sending and receiving email is secure. Although email messages can be retrieved by others but they still will not be able to be read because the text is displayed in hexadecimal character.

Keywords: Cryptography, Rijndael, Email Encryption and Decryption.

1. Pendahuluan

Sejalan dengan perkembangan teknologi, semakin mengubah cara masyarakat dalam berkomunikasi. Dulu komunikasi jarak jauh masih menggunakan cara yang konvensional, yaitu dengan cara saling mengirim surat, tetapi sekarang komunikasi jarak jauh dapat dilakukan dengan mudah dan cepat yaitu dengan adanya teknologi seperti *email*, SMS (*Short Messaging Service*), dan Internet yang merupakan salah satu teknologi telekomunikasi yang paling banyak digunakan.

Namun tidak semua perkembangan teknologi komunikasi memberikan dampak yang positif dan menguntungkan. Salah satu dampak negatif dalam perkembangan teknologi adalah adanya penyadapan data, yang merupakan salah satu masalah yang paling ditakuti oleh para pengguna jaringan komunikasi. Karena itulah dibutuhkan suatu metode yang dapat menjaga kerahasiaan informasi ini. Metode yang dimaksud adalah kriptografi. Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital[2].

Secara umum kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita [7].

Pada Penelitian ini dirancang sebuah sistem aplikasi *mail server* dengan kliennya dan melakukan metode enkripsi - dekripsi menggunakan algoritma AES (*Advance Encryption Standart*) pada isi pesan bertipe *plaintext*.

Tujuan dari pembuatan penelitian ini adalah untuk merancang dan membuat aplikasi *mail client* yang dirasakan aman dari para *hacker* yang sering melakukan pengendusan data *email* yang sedang lalu lalang melalui jaringan Internet dan menerapkan algoritma kriptografi AES untuk enkripsi dan dekripsi pada email.

2. Metode

2.1. Enkripsi

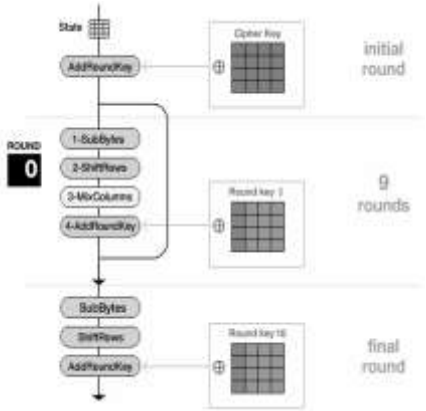
Enkripsi adalah proses mengubah suatu pesan asli (*plaintext*) menjadi suatu pesan dalam bahasa sandi (*ciphertext*).

C = E (M) (1)

dimana

M = Pesan asli (*Plaintext*)
E = Proses enkripsi dengan *Key Private*
C = *Chipertext* (*Plaintext* yang terenkripsi AES)

Dibawah ini merupakan gambar diagram proses enkripsi seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Diagram Proses Enkripsi

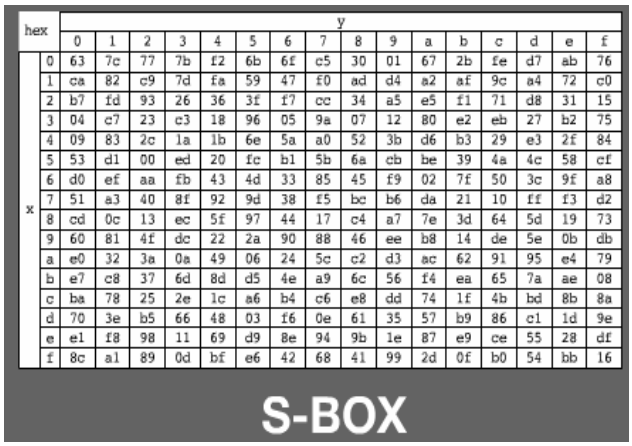
Garis besar Algoritma AES *Rijndael* yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan *round key*).

1. *AddRoundKey*: melakukan *XOR* antara *state* awal (*plainteks*) dengan *cipherkey*. Tahap ini disebut juga *initial round***.**
2. *Round :* Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a) *SubBytes*: substitusi *byte* dengan menggunakan table substitusi (*S-box*).
   b) *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
   c) *MixColumns*: mengacak data di masing-masing kolom *array state.*
   d) *AddRoundKey*: melakukan XOR antara *state* sekarang *round key*.
3. *Final round***:** proses untuk putaran terakhir:
   a) *SubBytes*
   b) *ShiftRows*
   c) *AddRoundKey*

Langkah kerja dari enkripsi adalah sebagai berikut.

2.1.1. Transformasi *SubBytes()*

Transformasi *SubBytes()* memetakan setiap byte dari *array state* dengan menggunakan tabel substiusi *S-box*. Tabel *S-box* yang digunakan adalah seperti yang ditunjukkan pada gambar 2.



hex	y															
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

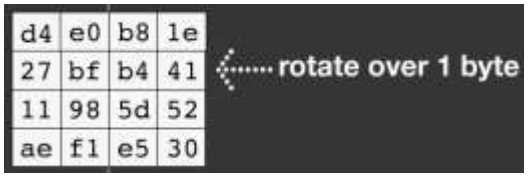
S-BOX

Gambar 2. Tabel S-BOX

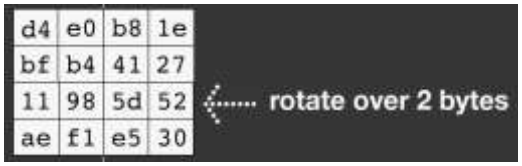
2.1.2. Transformasi *ShiftRows()*

Melakukan pergeseran secara *wrapping* (siklik) pada 3 baris terakhir dari array state. Jumlah pergeseran bergantung pada nilai baris (r). Baris *r* = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 *byte*, dan baris *r* = 3 digeser sejauh 3 *byte*. Baris *r* = 0 tidak digeser.
Contoh ditunjukkan pada Gambar 3 berikut.

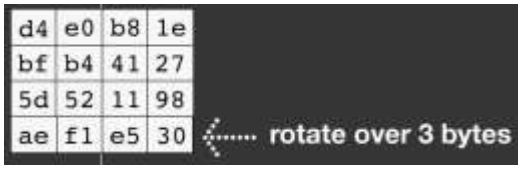
Geser baris ke-1:



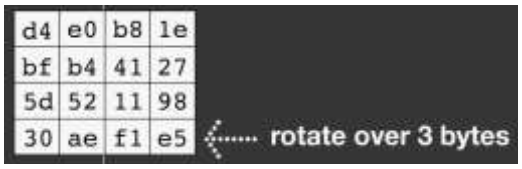
Hasil pergeseran baris ke-1 dan geser baris ke-2:



Hasil pergeseran baris ke-2 dan geser baris ke-3:



Hasil pergeseran baris ke-3:



Gambar 3. Contoh Transformasi ShiftRows ()

2.1.3. Transformasi *MixColumns()*

Transformasi MixColumns() mengalikan setiap kolom dari *array state* dengan polinom a(x) mod ($x^4 + 1$). Setiap kolom diperlakukan sebagai polinom 4-suku pada GF(2^8). a(x) yang ditetapkan adalah a(x) = {03}x^3 + {01}x^2 + {01}x + {02}.
Transformasi ini dinyatakan sebagai perkalian matriks

$$s'(x) = a(x) \otimes s(x)$$

$$s'(x) = a(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

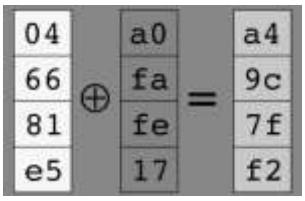
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{3,c})$$

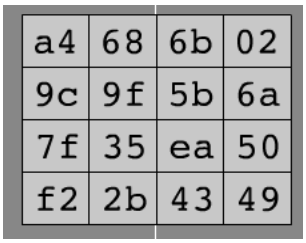
2.1.4. Transformasi *AddRoundKey()*

Transformasi ini melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya disimpan di *array state*. Gambar 4 menunjukkan contoh transformasi AddRoundKey ().

XOR-kan kolom pertama *state* dengan kolom pertama *round key*



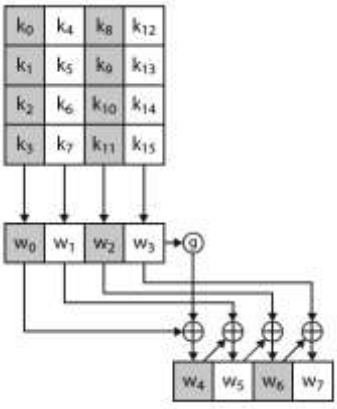
Hasil *AddRoundKey()* terhadap seluruh kolom:



Gambar 4. Contoh Transformasi AddRoundKey *()*

2.1.5. Ekspansi Kunci (*Key Expansion*)

Ekspansi kunci pada AES 128-bit (16-byte) menggunakan 4-*words* (16 byte) sebagai input dan menghasilkan perluasan kunci menjadi 44 *words* (176 bytes). Gambar 5 menunjukkan proses dari *key expansion ().*



Gambar 5. Proses *KeyExpansion*()

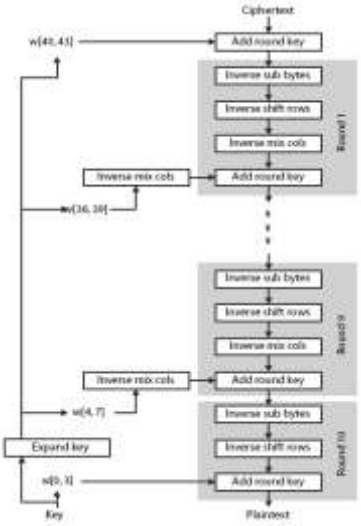
2.2. Dekripsi

Dekripsi adalah proses mengubah pesan dalam suatu bahasa sandi menjadi pesan asli kembali.

M = D (C) (2)
dimana

C = *Chipertext* (Hasil *Plaintext* terenkripsi)
D = Proses dekripsi menggunakan *key private*
M = Pesan asli setelah di dekripsi

Gambar 6 menunjukkan proses dari sebuah dekripsi pesan.



Gambar 6. Diagram Alur Proses Dekripsi

3. Hasil dan Analisa

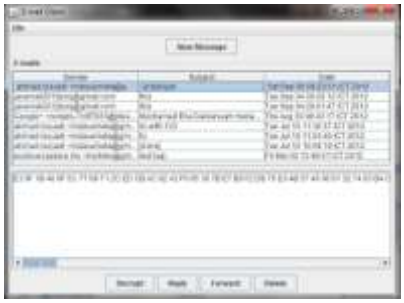
Langkah awal untuk menjalankan aplikasi enkripsi dan dekripsi email ini adalah membuka Netbeans 7.0 dan login menggunakan email user yang terhubung dengan internet. Tampilan awal dari program ini ditunjukkan pada Gambar 7.

Gambar 7. Tampilan menu utama

3.1 Enkripsi Pesan

Pilihan untuk mengenkripsi teks akan muncul ketika pesan diklik. Jika ingin mengenkripsi pesan cukup tekan '**Yes**' kemudian akan muncul pewaktu konfirmasi yang menunjukkan waktu untuk proses enkripsi pesan tersebut. Jika ingin pesan asli yang ditampilkan, maka cukup tekan '**No**'. Gambar 8 dan Gambar 9 menunjukkan proses enkripsi.

Gambar 8. Tampilan Menu pilihan untuk enkripsi



Gambar 9. Tampilan pesan yang telah dienkripsi

3.2 Dekripsi Pesan

Setelah mengenkripsi pesan dan kita ingin mengecek apakah isi pesan tersebut sama dengan yang asli, tombol **Decrypt** diklik, sehingga pesan asli akan tampil. Gambar 10 dan Gambar 11 menunjukkan proses dari dekripsi pesan.

Gambar 10. Tampilan konfirmasi untuk memasukkan kunci dekripsi

Gambar 11. Tampilan hasil dekripsi

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat diambil beberapa kesimpulan yaitu dengan perangkat lunak ini, keamanan dalam mengirim dan menerima email sekiranya dapat terjamin. Walaupun pesan *email* bisa diambil orang lain tetapi mereka tetap tidak akan bisa membacanya karena teks tertampil dalam bentuk karakter heksadesimal dan jika di jadikan string maka berupa simbol-simbol tidak jelas. Perangkat lunak ini hanya mengamankan isi text email bukan mengamankan jalur transfer email.

Saran untuk pengembangan aplikasi ini di masa depan yang akan datang, yaitu perlu dilakukan penelitian lebih lanjut untuk enkripsi dan dekripsi email menggunakan algoritma kriptografi yang lain dan jika ingin lebih aman sebaiknya menggunakan metode AES dengan kunci 256-byte, yaitu melakukan perputaran sebanyak 14 kali/ 14 Rounds sehingga ekspansi kuncinya semakin besar dan semakin sulit untuk di pecahkan.

6. Referensi

[1] Adhi, J. S., Kriptografi dengan Algoritma Rijndael untuk Penyandian Data, Skripsi S-1, Universitas Kristen Duta Wacana, Yogyakarta, 2005.
[2] Ariyus, D., Kriptografi Keamanan Data dan Komunikasi, Graha Ilmu, Yogyakarta, 2006.
[3] Galice, S. & Minier, M., Improving Integral Attacks Against Rijndael-256 Up to 9 Rounds, Laboratoire CITI, INSA de Lyon, France, 2007.
[4] Giyanto, T., Rancang Bangun Perangkat Lunak Pembelajaran Kriptografi Menggunakan Metode WAKE (Word Auto Key Encryption), Skripsi S-1, Institut Teknologi Sepuluh November, Surabaya, 2009.
[5] Williams, J. P., Advanced Encryption Standard Key Expansion Test Documentation, 2009.
[6] Yolanda, E. S., Implementasi Disk Encryption Menggunakan Algoritma Rijndael, Skripsi S-1, Institut Teknologi Bandung, Bandung, 2008.
[7] Menezes A, van Oorschot P, Vanstone S. Handbook of Applied Cryptography. CRC Press. 1996.