ENKRIPSI DAN DEKRIPSI DENGAN ALGORITMA AES 256 UNTUK SEMUA JENIS FILE

Voni Yuniati [1] , Gani Indriyanta[2], Antonius Rachmat C[3]

Abstrak:

Kemajuan teknologi komputer dan telekomunikasi telah menjadi kebutuhan dan sangat membantu dalam menyelesaikan banyak pekerjaan dengan cepat, akurat, dan efisien. Namun seiring dengan kemajuan teknologi, terdapat dampak negatif berupa penyadapan data sehingga aspek keamanan dalam pertukaran informai dianggap penting. Dalam dunia informasi terdapat data-data penting dan bersifat rahasia yang tidak boleh diketahui oleh umum.

Kriptografi merupakan salah satu solusi atau metode pengamanan data yang tepat untuk menjaga kerahasiaan dan keaslian data, serta dapat meningkatkan aspek keamanan suatu data atau informasi. Metode ini bertujuan agar informasi yang bersifat rahasia dan dikirim melalui suatu jaringan, seperti LAN atau Internet, tidak dapat diketahui atau dimanfaatkan oleh orang atau pihak yang tidak berkepentingan. Kriptografi mendukung kebutuhan dua aspek keamanan informasi, yaitu perlindungan terhadap kerahasiaan data informasi dan perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan.

Hasil penelitian menunjukkan bahwa algoritma AES dengan panjang kunci 256 *bit* dapat menyandikan isi suatu file sehingga dapat mengamankan file tersebut. Ukuran file enkripsi akan bertambah 11 *bytes* dari file asli karena adanya proses penambahan *header* yang berisi informasi ekstensi file. Dalam pengembangan sistem berikutnya diharapkan sistem dapat mempunyai fasilitas untuk menyembunyikan *folder* yang digunakan untuk menyimpan file enkripsi maupun file dekripsi.

Kata Kunci : *Advanced Encryption Standard*, dekripsi, enkripsi, kriptografi, Rijndael

1. Pendahuluan

Perkembangan teknologi komputer dan telekomunikasi dewasa ini telah mengalami kemajuan yang sangat pesat dan sudah menjadi suatu kebutuhan, karena banyak pekerjaan dapat diselesaikan dengan cepat, akurat, dan efisien. Sejalan dengan perkembangan teknologi tersebut, 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. Internet telah membuat komunikasi semakin terbuka dan pertukaran informasi juga semakin cepat melewati batas-batas negara dan budaya. 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. Dengan adanya penyadapan data maka aspek keamanan dalam pertukaran informasi dianggap penting, karena suatu komunikasi data jarak jauh belum tentu memiliki jalur transmisi yang aman dari penyadapan sehingga keamanan informasi menjadi bagian penting dalam dunia informasi itu sendiri. Di dalam dunia informasi terdapat data-data yang tidak terlalu penting jadi jika publik mengetahui data tersebut pemilik data tidak terlalu dirugikan. Tetapi apabila pemilik data adalah pihak militer atau pihak pemerintah, keamanan dalam pertukaran informasi menjadi sangatlah penting karena data yang mereka kirim adalah data-data rahasia yang tidak boleh diketahui oleh publik.

[1] *Voni Yuniati, Alumni Fakultas Teknik Informatika – Universitas Kristen Duta Wacana Yogyakarta. Email : 22043475@ukdw.ac.id*

[2] *Ir. Gani Indriyanta, M.T., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana. Email : ganind@ukdw.ac.id*

[3] *Antonius Rachmat C, S.Kom., M.Cs., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana. Email : anton@ukdw.ac.id*

Kriptografi merupakan salah satu solusi atau metode pengamanan data yang tepat untuk menjaga kerahasiaan dan keaslian data, serta dapat meningkatkan aspek keamanan suatu data atau informasi. Metode ini bertujuan agar informasi yang bersifat rahasia dan dikirim melalui suatu jaringan, seperti LAN atau internet, tidak dapat diketahui atau dimanfaatkan oleh orang atau pihak yang tidak berkepentingan. Kriptografi mendukung kebutuhan dua aspek keamanan informasi, yaitu perlindungan terhadap kerahasiaan data informasi dan perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan.

Untuk mengetahui apakah suatu algoritma kriptografi dapat mengamankan data dengan baik dapat dilihat dari segi lamanya waktu proses pembobolan untuk memecahkan data yang telah disandikan. Seiring dengan perkembangan teknologi komputer yang semakin canggih, maka dunia teknologi informasi membutuhkan algoritma kriptografi yang lebih kuat dan aman. Saat ini, AES (*Advanced Encryption Standard*) merupakan algoritma *cipher* yang cukup aman untuk melindungi data atau informasi yang bersifat rahasia. Pada tahun 2001, AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (*National Institute of Standard and Technology*) sebagai pengganti algoritma DES (*Data Encryption Standard*) yang sudah berakhir masa penggunaannya. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit.

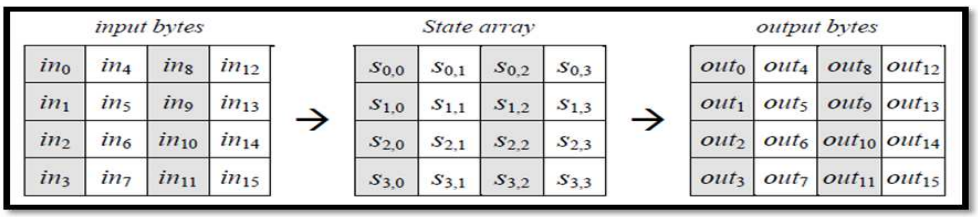
2. Deskripsi Singkat Mengenai *Advanced Encryption Standard*

Input dan *output* dari algoritma AES terdiri dari urutan data sebesar 128 *bit*. Urutan data yang sudah terbentuk dalam satu kelompok 128 *bit* tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext*. *Cipher key* dari AES terdiri dari *key* dengan panjang 128 *bit*, 192 *bit*, atau 256 *bit*. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini. Berikut ini adalah Tabel 1 yang memperlihatkan jumlah *round* / putaran (Nr) yang harus diimplementasikan pada masing-masing panjang kunci.

Tabel 1 Perbandingan Jumlah Round dan Key
Dikutip dari: Kriptografi, 2006, halaman 158

	Jumlah Key (Nk)	Ukuran Block (Nb)	Jumlah Putaran (Nr)
AES – 128	4	4	10
AES – 192	6	4	12
AES – 256	8	4	14

Pada dasarnya, operasi AES dilakukan terhadap *array of byte* dua dimensi yang disebut dengan *state*. *State* mempunyai ukuran *NROWS* X *NCOLS*. Pada awal enkripsi, data masukan yang berupa in0, in2, in3, in4, in5, in6, in7, in8, in9, in10, in11, in12, in13, in14, in15 disalin ke dalam *array state*. *State* inilah yang nantinya dilakukan operasi enkripsi / dekripsi. Kemudian keluarannya akan ditampung ke dalam *array out*. Gambar 1 mengilustrasikan proses penyalinan dari *input bytes*, *state array*, dan *output bytes* :



Gambar 1 Proses *Input Bytes, State Array, dan Output Bytes*
Dikutip dari: Kriptografi, 2006, halaman 161

Pada saat permulaan, *input bit* pertama kali akan disusun menjadi suatu *array byte* dimana panjang dari *array byte* yang digunakan pada AES adalah sepanjang 8 *bit* data. *Array byte* inilah yang nantinya akan dimasukkan atau di*copy* ke dalam *state* dengan urutan dimana r (*row* / baris) dan c (*column*/kolom) :

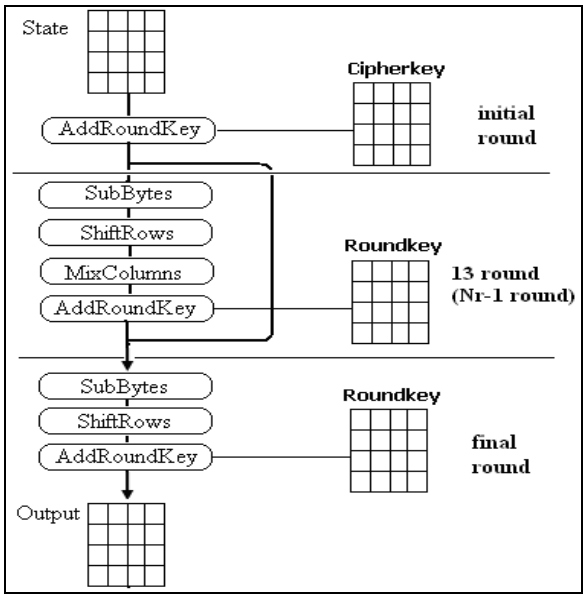
$$s[r,c] = in[r+4c] \text{ untuk } 0 \le r < 4 \text{ dan } 0 \le c < Nb$$

sedangkan dari state akan dicopy ke output dengan urutan :

$$out[r+4c] = s[r,c] \text{ untuk } 0 \le r < 4 \text{ dan } 0 \le c < Nb$$

3. Proses Enkripsi *Advanced Enryption Standard*

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes, ShiftRows, Mixcolumns,* dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah di*copy*kan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows*, *MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar 2 di bawah ini :



Gambar 2 Ilustrasi Proses Enkripsi AES
Dikutip dari: Kriptografi, 2006, halaman 159

3.1 *AddRoundKey*

Pada proses enkripsi dan dekripsi AES proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada *state* dengan operasi XOR. Setiap *round key* terdiri dari Nb *word* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :

$$\left[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}\right] = \left[s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}\right] \oplus \left[w_{round*Nb+c}\right] \text{untuk } 0 \le c \le Nb$$

[w_i] adalah *word* dari *key* yang bersesuaian dimana i = round*Nb+c. Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada *round* = 0 untuk *round* selanjutnya *round* = *round* + 1, pada proses dekripsi pertama kali pada *round* = 14 untuk *round* selanjutnya *round* = *round* - 1.

3.2 *SubBytes*

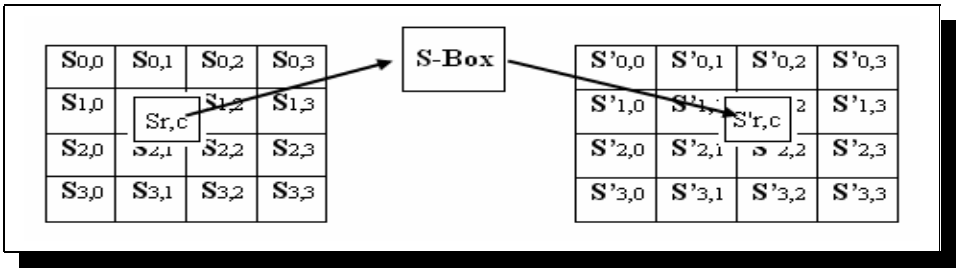
SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam Tabel 2.

Tabel 2 Tabel S-Box *SubBytes*

Dikutip dari: *Federal Information Processing Standart*-197 [FIPS-197], 2001, halaman 16

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

Untuk setiap *byte* pada *array state*, misalkan *S*[*r*, *c*] = *xy*, yang dalam hal ini *xy* adalah *digit* heksadesimal dari nilai *S*[*r*, *c*], maka nilai substitusinya, dinyatakan dengan *S*'[*r*, *c*], adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris *x* dengan kolom *y*. Gambar 3 mengilustrasikan pengaruh pemetaan *byte* pada setiap *byte* dalam *state*.

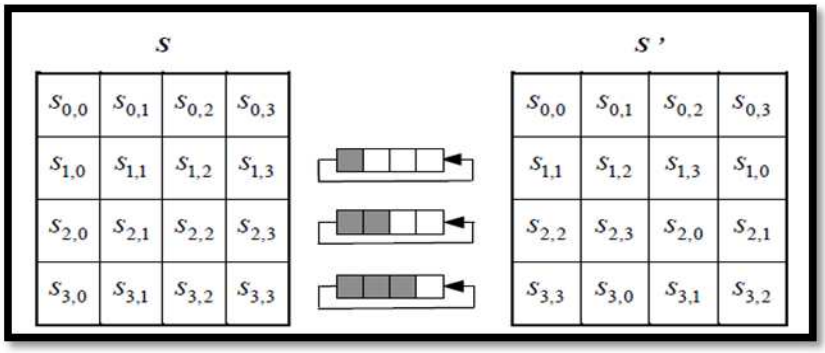


Gambar 3 Pengaruh Pemetaan pada Setiap *Byte* dalam *State*

Dikutip dari: Kriptografi, 2006, halaman 163

3.3 *Shiftrows*

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran *bit* dimana *bit* paling kiri akan dipindahkan menjadi *bit* paling kanan (rotasi *bit*). Proses pergeseran *Shiftrow* ditunjukkan dalam Gambar 4 berikut:



Gambar 4 Transformasi *ShiftRows*
Dikutip dari: Kriptografi, 2006, halaman 165

3.4 *MixColumns*

MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi *mixcolumns* dapat dilihat pada perkalian matriks berikut ini:

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \quad \text{… [1]}$$

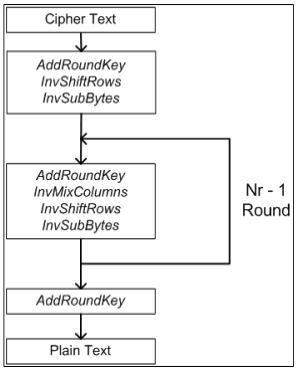
Hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada di bawah ini :

$$\begin{aligned} s'_{0,c} &= (\{02\} \bullet s_{o,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\ s'_{3,c} &= (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \end{aligned} \quad \text{… [2]}$$

4. Proses Dekripsi *Advanced Encryption Standard*

4.1 Proses Dekripsi AES

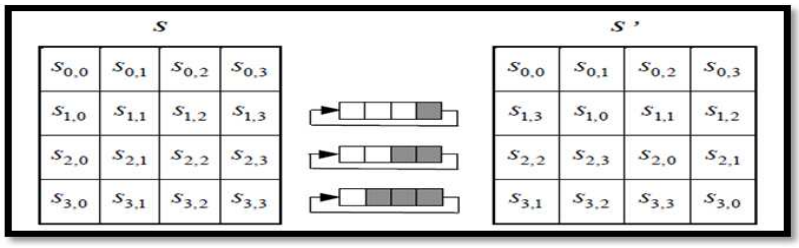
Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada *invers cipher* adalah *InvShiftRows, InvSubBytes, InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada skema berikut ini :



Gambar 5 Ilustrasi Proses Dekripsi AES
Dikutip dari: Kriptografi (Keamanan Data dan Komunikasi), 2006, halaman 169

4.2 *InvShiftRows*

InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi *InvShiftRows* terdapat pada Gambar 6:



Gambar 6 Transformasi *InvShiftRows*
Dikutip dari: Kriptografi (Keamanan Data dan Komunikasi), 2006, halaman 165

4.3 *InvSubBytes*

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse S-Box*. Tabel *Inverse S-Box* akan ditunjukkan dalam Tabel 3 berikut:

Tabel 3 Tabel *Inverse S-Box*
Dikutip dari: *Federal Information Processing Standart*-197 [FIPS-197], 2001, halaman 22

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

4.4 *InvMixColumns*

Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{... [3]}$$

Hasil dari perkalian dalam matrik adalah :

$$s'_{0,c} = (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

…[4]

5. Proses Ekspansi Kunci

Algoritma AES mengambil kunci *cipher* dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Algoritma ini membutuhkan set awal key yang terdiri dari Nb *word*, dan setiap *round* Nr membutuhkan data kunci sebanyak Nb *word*. Hasil *key schedule* terdiri dari *array* 4 *byte word* linear yang dinotasikan dengan [w_i]. *SubWord* adalah fungsi yang mengambil 4 *byte word* input dan mengaplikasikan S-Box ke tiap-tiap data 4 *byte* untuk menghasilkan *word output*. Fungsi *RotWord* mengambil *word* [a_0, a_1, a_2, a_3] sebagai *input*, melakukan permutasi siklik, dan mengembalikan *word* [a_1, a_2, a_3, a_0]. Rcon[i] terdiri dari nilai-nilai yang diberikan oleh [x^{i-1}, {00}, {00}, {00}], dengan x^{i-1} sebagai pangkat dari x (x dinotasikan sebagai {02}). Pseudocode dari proses ekspansi kunci dapat dilihat dalam gambar berikut ini:

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word  temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end
```

Gambar 7 *Pseudocode* Ekspansi Kunci

Dikutip dari:Ariyus Dony (2006), Kriptograf (Keamanan Data dan Komunikasi), halaman 94

Dari Gambar 7 dapat dilihat bahwa *word* ke *Nk* pertama pada ekspansi kunci berisi kunci *cipher*. Setiap *word* berikutnya, w[i], sama dengan XOR dari *word* sebelumnya, w[i-1] dan *word Nk* yang ada pada posisi sebelumnya, w[i-Nk]. Untuk *word* pada posisi yang merupakan kelipatan *Nk*, sebuah transformasi diaplikasikan pada w[i-1] sebelum XOR, lalu dilanjutkan oleh XOR dengan konstanta *round*, Rcon[i]. Transformasi ini terdiri dari pergeseran siklik dari *byte* data dalam suatu *word RotWord*, lalu diikuti aplikasi dari *lookup* tabel untuk semua 4 *byte* data dari *word SubWord*.

6. Penerapan Algoritma AES 256

Proses enkripsi adalah proses yang mengubah *plain text* menjadi *cipher text* yang bertujuan untuk mengamankan data atau isi pesan yang bersifat rahasia agar tidak disadap oleh kriptanalis. Sedangkan proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu mengubah *cipher text* menjadi *plain text*. Dalam melakukan proses dekripsi, isi file yang berupa *cipher text* harus diubah kembali menjadi pesan atau file asli (*plain text*). Untuk itu perlu

dilakukan analisis isi file dari segi ketepatan atau kecocokan file apakah file yang akan didekripsi bisa kembali ke file asli dengan sempurna atau tidak. Berikut ini akan diberikan 1 contoh mengenai analisis isi file pada file image.

File image :

File asli (pic.jpeg) :

Gambar 8 File *Image* Sumber "pic.jpeg"

File setelah dienkripsi (pic.aes) :

File setelah didekripsi (pic.jpeg) :

Gambar 9 File Hasil Dekripsi dari "pic.aes"

Dari contoh file pic.jpeg menunjukkan bahwa setelah dilakukan proses dekripsi, image kembali seperti semula (file asli). Oleh karena itu, dapat disimpulkan bahwa isi file enkripsi cocok atau sesuai dengan file asli, sehingga pada waktu dilakukan proses dekripsi, image bisa kembali semula seperti pada file asli.

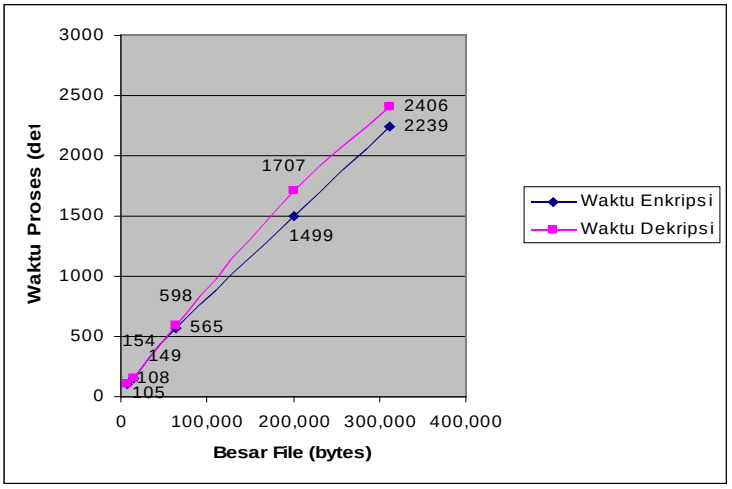
6.1 Analisis Waktu Enkripsi dan Dekripsi

Waktu adalah salah satu faktor dalam melakukan proses enkripsi dan dekripsi. Dengan adanya waktu, kecepatan proses enkripsi dan dekripsi dapat diketahui. Berikut ini akan diberi tabel perbandingan waktu enkripsi dan dekripsi pada semua tipe file :

Berikut ini terdapat tabel dan grafik yang menyajikan hubungan antara besar file dengan waktu yang dibutuhkan untuk proses enkripsi dan proses dekripsi:

Tabel 4
Tabel Hubungan antara Besar File dengan Waktu Proses Enkripsi dan Proses Dekripsi

No	Nama File	Kata Kunci	Ukuran File Asli (bytes)	Waktu Enkripsi	Waktu Dekripsi
1	Keygen.exe	9999999999999999 9999999999999999	8,192	00:01:45	00:01:48
2	UKDWNM.exe	coba	14,926	00:02:29	00:02:34
3	Blocks.exe	aku1aku1aku1aku1 aku1aku1aku1aku1	63,504	00:09:25	00:11:38
4	Cat3.exe	coba	200,704	00:24:59	00:28:27
5	SinChan.exe	coba	310,799	00:37:19	00:40:06



Gambar 10
Grafik Hubungan antara Besar File dengan Waktu Proses Enkripsi dan Proses Dekripsi

Pada Gambar 10 menunjukkan kecenderungan kenaikan waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi. Semakin besar ukuran suatu file maka semakin lama pula waktu yang dibutuhkan untuk melakukan proses enkripsi dan proses dekripsi.

7. Kesimpulan

Berdasarkan rumusan masalah, implementasi, dan analisa sistem, maka didapat kesimpulan sebagai berikut :

- Dalam penelitian ini, file dekripsi dapat kembali seperti ekstensi file sumber karena saat sistem melakukan proses enkripsi ditambahkan *header* untuk menyimpan informasi ekstensi

file sumber. Oleh karena itu, ukuran file enkripsi akan bertambah 11 *byte* dari ukuran file asli. Sedangkan ukuran file dekripsi akan kembali ke ukuran file asli.

- Dari hasil penelitian telah dibuktikan bahwa isi file yang telah dienkripsi merupakan isi file dari file sumber, sehingga apabila akan dilakukan proses dekripsi, maka akan kembali seperti file sumber semula.
- Waktu yang diperlukan untuk proses enkripsi pada penelitian ini tidak sama dengan waktu proses dekripsi yang dikarenakan adanya pemakaian *resource* komputer.
- Dalam penelitian ini, saat terjadi proses *save* pada file .aes dengan maupun tanpa mengganti informasi ekstensi file sumber maka pada saat dilakukan proses dekripsi, terdapat isi file yang tidak kembali seperti file sumber. Hal ini disebabkan file .aes tidak lagi berbasis file hasil proses enkripsi melainkan akan menjadi file .aes berupa file teks sehingga akan terjadi perubahan *header* file dalam file .aes.

8. Daftar Pustaka

Adhi, J. S., (2005), Kriptografi dengan Algoritma Rijndael untuk Penyandian Data, Skripsi, Yogyakarta: Universitas Kristen Duta Wacana.

Dony, A. (2005), Kriptografi Keamanan Data dan Komunikasi, Yogyakarta: Penerbit Andi Offset.

Federal Information Processing Standards Publications. (2001). National Institute of Standards and Technology.

Galice, S. & Minier, M. (2007). Improving Integral Attacks Against Rijndael-256 Up to 9 Rounds. France: Laboratoire CITI, INSA de Lyon.

Kurniawan, Budi. (2007). Penyandian Teks dengan AES-128. Yogyakarta: Universitas Kristen Duta Wacana.

Kristina, Yochebed K (2009). Enkripsi Email Client dengan Algoritma Rijndael 128. Yogyakarta: Universitas Kristen Duta Wacana.

Munir, Rinaldi. (2006). Kriptografi. Bandung: Penerbit Informatika.

Sijoni. (2008). Implementasi Advanced Encryption Standard (AES) dan Cipher Block Chaining (CBC) untuk Enkripsi dan Dekripsi File Text. Yogyakarta: Universitas Kristen Duta Wacana.

Yolanda, Elfira S. (2008). Implementasi Disk Encryption Menggunakan Algoritma Rijndael. Bandung: Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

9. Riwayat Penulis

[1] **Voni Yuniati**: Alumni Fakultas Teknik Informatika – Universitas Kristen Duta Wacana Yogyakarta, e-mail: 22043475@ukdw.ac.id

[2] **Ir. Gani Indriyanta, M.T.** : Dosen Fakultas Teknik Informatika – Universitas Kristen Duta Wacana Yogyakarta, e-mail: ganind@ukdw.ac.id

[3] **Antonius Rachmat C, S.Kom, M.Cs**: Dosen Fakultas Teknik Informatika – Universitas Kristen Duta Wacana Yogyakarta, e-mail: anton@ukdw.ac.id