e-ISSN 2540-7902 dan p-ISSN 2541-366X

Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)

1st * Faturungi Muharram
Universitas Muslim Indonesia
Fakultas Ilmu Komputer
Makassar, Indonesia
faturungimuharram@gmail.com

2nd Huzain Azis
Universitas Muslim Indonesia
Fakultas Ilmu Komputer
Makassar, Indonesia
huzain.azis@umi.ac.id

3rd Abdul Rachman Manga'
Universitas Muslim Indonesia
Fakultas Ilmu Komputer
Makassar, Indonesia
abdulrachman.manga@umi.ac.id

***Abstrak*—Meninjau dalam penggunaan teknologi, manusia tak pernah lepas dari kebutuhan akan sebuah informasi. Beberapa informasi dapat berupa file gambar, dokumen, dan video. Salah satu dari informasi tersebut banyak mengandung informasi penting yaitu informasi dalam bentuk file dokumen. Beberapa informasi memiliki privasi yang tidak boleh tersebar oleh *public*, oleh karena itu diperlukan cara dalam mengamankan informasi agar informasi tidak tersebar luas kepada pihak yang tak berwenang, dalam hal ini keamanan adalah salah satu hal yang penting. Salah satu cara yang diperlukan adalah menggunakan metode kriptografi. Dalam proses kriptografi terdapat konsep dasar yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal** [1]**. Pada proses enkripsi terdapat beberapa cara yang dapat digunakan dan memiliki tingkat kekuatan serta kecepatan dan kelemahan dalam proses enkripsi tersendiri. Terdapat jenis model kriptografi, namun pada penelitian ini akan menyajikan analisis terhadap model kriptografi *Advanced Encryption Standard* (AES).**

***Kata Kunci*—kriptografi, enkripsi, dekripsi, AES**

I. PENDAHULUAN

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi akan tetapi masalah keamanan sering kali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting [2]. Perkembangan teknologi pada bidang komputer berkaitan dengan semakin banyaknya penggunaan sistem komputer. Perkembangan ini pun dibarengi dengan bertambahnya konektifitas internet melalui jaringan sebagai media bertukar data dan informasi. Sejak lahirnya konsep open sistem, semua data dapat mengalir bebas melewati jaringan komputer. Namun, hal ini menjadi resiko tersendiri bagi pengguna karena data tersebut dapat diakses oleh pihak yang tidak berkepentingan. Berbagai cara dilakukan untuk mendapatkan data dan informasi, mulai dari tingkatan yang mudah sampai pada cara-cara yang rumit [3]. Salah satu cara untuk mengamankan data dari tindakan kejahatan adalah menggunakan konsep kriptografi. Kriptografi adalah bidang ilmu yang mempelajari bagaimana cara mengamankan suatu pesan atau informasi. Upaya untuk menjaga pesan atau informasi rahasia telah ada sejak zaman dahulu kala. Julius Caesar, Kaisar Romawi, telah menggunakan metode enkripsi sederhana dengan cara menggeser setiap karakter dengan nilai tertentu [4]. Hal tersebut membawa pula dampak pada sebuah usaha dalam melakukan tindak kejahatan yakni pencurian informasi secara ilegal. Gangguan ini dapat menjadi hal yang sangat fatal karena bias saja informasi pribadi dapat dengan mudah dibaca oleh pihak lain.

Oleh sebab itu dibutuhkan sebuah teknik dalam mengamankan data menggunakan kriptografi. Kriptografi menjadi teknik alternatif untuk memungkinkan dua orang saling bertukar pesan dengan mengubah pesan menjadi pesan sandi yang memungkinkan tidak dapat dibaca oleh orang yang tidak berhak. Kriptografi mendukung kebutuhan dari dua aspek keamanan informasi, yaitu *secrecy* dan *authenticity*. *Secrecy* merupakan perlindungan terhadap kerahasiaan berkas informasi, sedangkan *Authenticity* merupakan perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan [5]. Proses pengiriman pesan akan melalui proses enkripsi untuk mengubah teks asli (*plaintext*) menjadi teks sandi (*ciphertext*). 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 [6].

Dengan enkripsi, suatu informasi akan menjadi lebih sulit untuk diketahui oleh orang yang tidak berhak. Keamanan tersebut diperlukan untuk menghindari adanya penyadapan atau pembajakan terhadap gambar yang mengandung informasi penting bagi penggunanya. Keamanan diperlukan untuk menjaga integritas gambar tersebut agar tetap aman. Terdapat lebih dari satu model kriptografi salah satunya adalah AES.

*Corresponding Author

Algoritma AES merupakan algoritma *chiper* yang aman untuk melindungi data atau informasi yang bersifat rahasia.

II. METODOLOGI

A. Deskripsi Algoritma AES

Advanced Encryption Standard (AES) adalah algoritma kriptografi yang menjadi standar algoritma enkripsi kunci simetris pada saat ini. *Advanced Encryption Standard* (AES) dipublikasikan oleh NIST (*National Institute* of *Standard and Technology*) pada tahun 2001. AES merupakan blok kode simetris untuk menggantikan DES (Data *Encryption* Standard) [7].

Saat ini, AES merupakan algoritma kriptografi 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 sebagai pengganti algoritma DES yang sudah berakhir masa penggunaannya [8].

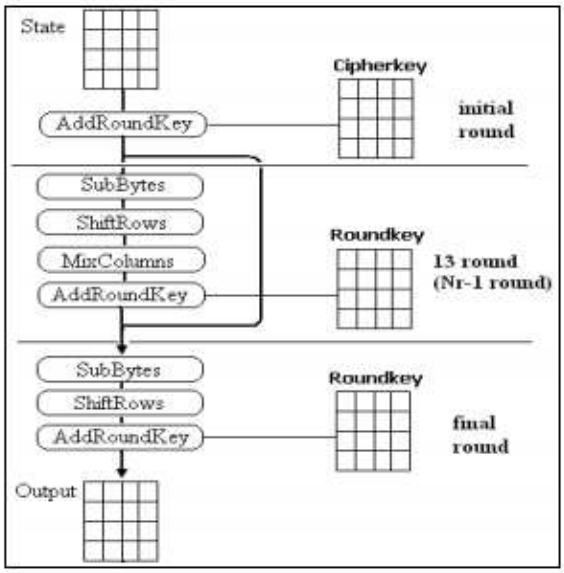
Algoritma AES merupakan algoritma simetris yaitu mengunakan kunci yang sama untuk proses enkripsi dan dekripsi. Algoritma AES memiliki tiga pilihan kunci yaitu tipe: AES-128, AES-192 dan AES-256 [9]. Dalam algoritma kriptografi AES 128, 1blok plainteks berukuran 128 bit terlebih dahulu dikonversi menjadi matriks heksadesimal berukuran 4x4 yang disebut state [10]. AES dipublikasikan oleh NIST pada tahun 2001 yang digunakan untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dibobol. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [3]. Urutan data dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *chipertext*. Panjang kunci dari AES terdiri dari panjang kunci 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini yang nantinya mempengaruhi jumlah putaran pada algoritma AES ini. Jumlah putaran yang digunakan algoritma ini ada tiga macam seperti pada Tabel I.

TABEL I. PERBANDINGAN JUMLAH ROUND DAN KEY DIKUTIP DARI KRIPTOGRAFI, 2016, HALAMAN 158

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

B. Proses Enkripsi

Proses enkripsi pada algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey* [1]. Seluruh fungsi operasi (penjumlahan dann perkalian) yang tercakup dalam AES merupakan operasi-operasi yang didefinisikan dalam ruang lingkup *finite field* GF(2^8) dengan polinominal irreducible pembangkit $m(x) = x^8 + x^4 + x^3 + x + 1$. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar 1.



Gambar. 1.Ilustrasi Proses Enkripsi AES

1) AddRoundKey

Pada proses AES AddRoundKey, sebuah round key ditambahkan pada state dengan operasi XOR. Setiap key terdiri dari Nb word dimana tiap word tersebut akan dijumlahkan dengan word atau kolom yang bersesuaian dari state sehingga menjadi persamaan (1).

$$[S'_{0,c}, s'1,c, s'2,c, s'3,c] = [s0,c, s1,c, s3,c] \Phi [w_{round*Nb+c}] \quad (1)$$

untuk $0 \leq c \leq Nb$

$[W_1]$ adalah world dari key yang bersesuaian dimana i = round * Nb+c. Transformasi AddRoundKey pada proses enkripsi pertama kali pada round = 0 untuk round selanjutnya = round + 1, pada proses dekripsi pertama kali pada round = 14 untuk round selanjutnya round = round -1.

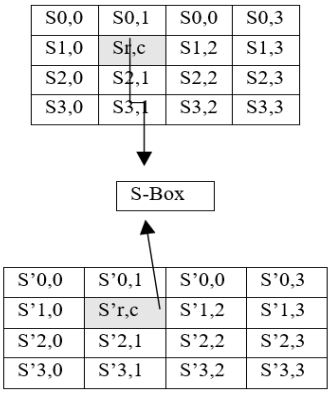
2) SubBytes

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 gambar 2.

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 2 dan 3, mengilustrasikan pengaruh pemetaan byte pada setiap byte dalam state.

x \ y	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

Gambar. 2. S-Box SubBytes Dikutip: federal Information Processing Standart-197 [FIPS-197],2001, hal 16



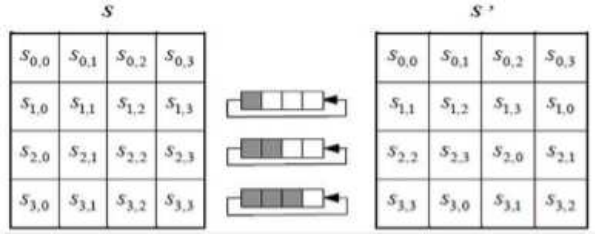
Gambar. 3. Pengaruh Pemetaan pada setiap Byte dalam state Dikutip Kriptografi, 2006, hal 163

3) Shifrows

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 berikut:

4) MixColumns

Mixcolumns mengoperasikan setiap elemen yang berada dalam satu kolom pada *state*. Secara lebih jelas, perkalian matriks *trasformasi mixcolumns* ditunjukkan pada Gambar 4



Gambar. 4. Perkalian matriks trasformasi mixcolumns

$$[S'_{0,c}, S'_1,c, S'_2,c, S'_3,c] = \begin{matrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{matrix} \begin{matrix} S0,c \\ S1,c \\ S2,c \\ S3,c \end{matrix} \quad (1)$$

Berdasarkan persamaan (1), hasil dari perkalian matriks diatas dapat dianggap seperti perkalian pada persamaan (2).

$$\begin{aligned} S'_{0,c} &= (\{02\}.S0,c) \Phi (\{03\}.S1,c) \Phi S2,c \Phi S3,c \\ S'_{1,c} &= S0,c \Phi (\{02\}.S1,c) \Phi (\{03\}.S2,c) \Phi S3,c \\ S'_{2,c} &= S0,c \Phi S1,c \Phi (\{02\}.S2,c) \Phi (\{03\}.S3,c) \\ S'_{3,c} &= (\{03\}.S1,c) \Phi S1,c \Phi S2,c \Phi (\{02\}.S3,c) \end{aligned} \quad (2)$$

C. Proses Dekripsi AES

Transformasi chipher dapat dibalikkan dan diimlementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Trasformasi byte yang digunakan pada invers cipher adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*.

1) 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

2) InvSubBytes

InvSubBytes juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan tabel Inverse S-Box.

3) InvMixColumns

Setiap kolom dalam state dikalikan dengan matrik perkalian dalam AES.

D. Proses Ekspansi Kunci

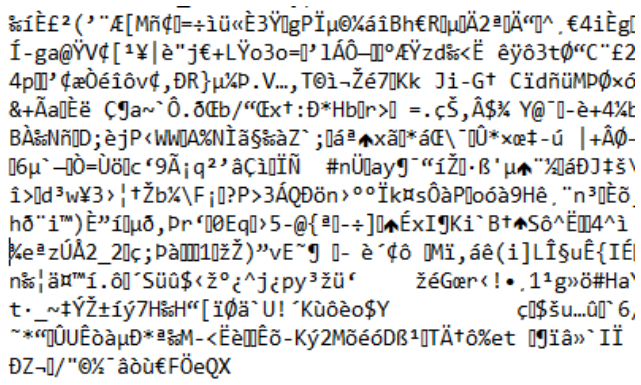
Algoritma AES mengambil kunci cipher dan melakukan rutin ekspansi kunci (key expansion) untuk membentuk key schedule. Ekspansi kunci menghasilkan total Nb(Nr+1) word. Algoritma ini membutuhkan set awal key yang terdiri dari Nb word, dan setiap round Nr membutuhkan data kunci sebanyak Nb word. Hasil key schedule terdiri dari array 4 byte word linear yang dinotasikan dengan [wi]. 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 [a0, a1, a2, a3] sebagai input, melakukan permutasi siklik, dan mengembalikan word [a1, a2, a3, a0]. Rcon[i] terdiri dari nilai-nilai yang diberikan oleh [xi-1, {00}, {00}, {00}], dengan xi-1 sebagai pangkat dari x (x dinotasikan sebagai {02}).

Sebuah file dengan format .jpg akan dilakukan tahap uji melalai proses enkripsi dan dekripsi dengan ketetapan file

dapat kembali ke bentuk asli setelah dilakukan proses dekripsi pada file yang telah dienkripsi, Gambar 5, 6, dan 7.

a) File gambar asli :

Gambar. 5. File gambar uji algoritma AES



Gambar. 6. File gambar setelah dienkripsi

Gambar. 7. File gambar setelah didekripsi

Berdasarkan uji coba terjadi perubahan bentuk data dimana file gambar yang awalnya dapat dibaca dalam bentuk pixel, dapat berubah menjadi file yang tidak bisa dibaca setelah melalui proses enkripsi AES hal ini telah memenuhi tujuan penelitian ini dimana konsep algoritma AES dapat menjaga kerahasiaan data sehingga data tetap aman dan terjaga dari pihak yang tak berwenang.

III. HASIL DAN PEMBAHASAN

Dilakukan sebuah percobaan dekripsi dan enkripsi pada sebuah file gambar. Berdasarkan data olah, akan diperoleh hasil perbandingan enkripsi dan dekripsi berdasarkan ukuran file dan proses waktu enkripsi dan dekripsi.

TABEL II. PROSES UJI ALGORITMA PADA TIGA FILE DATA TESTING

No	Nama File	Kata Kunci	Ukuran File	Waktu Enkripsi	Waktu Dekripsi
1	Haha.png	123456	468,08 Kb	00:00:43	00:00:47
2	Tes2.jpg	111111	78,93 Kb	00:00:2	00:00:3
3	Tes3.jpg	111112	12,37 Kb	00:00:1	00:00:1

IV. KESIMPULAN

Dari hasil uji coba pada proses enkripsi dan dekripsi maka dapat disimpulkan bahwa file yang melalui uji coba dekripsi akan berubah bentuk menjadi file yang tak bias dibaca, file dapat kembali kebentuk asli jika melalui proses dekripsi dengan menggunakan kunci yang sama saat enkripsi. Dan waktu proses hasil enkripsi-dekripsi data dapat dipengaruhi oleh besar ukuran data yang akan di uji.

DAFTAR PUSTAKA

[1] F. N. Pabokory, I. F. Astuti, and A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," *J. Inform. Mulawarman*, vol. 10, no. 1, pp. 20–31, 2015.

[2] M. S. Dharmawan, Eka Adhitya , Erni Yudaningtyas, "Perlindungan Web pada Login Sistem Menggunakan Algoritma Rijndael," *Eeccis*, vol. 7, no. 1, pp. 77–84, 2013.

[3] A. F. Marisman and A. Hidayati, "Pembangungan Aplikasi Pembanding Kriptografi dengan Caesar Cipher dan Advance Ecryption Standard(AES) untuk File Teks," *J. Penelit. Komun. dan Opini Publik*, vol. 19, no. 3, pp. 213–222, 2015.

[4] R. Primartha, "Penerapan Enkripsi dan Dekripsi File menggunakan Algoritma Advanced Encryption Standard (AES)," *J. Res. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 13–18, 2013.

[5] A. Kurniawati and M. D. Darmawan, "Implementasi Algoritma Advanced Encryption Standard (Aes) Untuk Enkripsi Dan Dekripsi Pada Dokumen Teks."

[6] V. Yuniati, G. Indriyanta, and A. Rachmat C., "Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File," *J. Inform.*, vol. 5, no. 1, 2011.

[7] A. Arif and P. Mandarani, "Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma Advanced Encryption Standard (AES) 128 Bit Pada Sistem Keamanan Short Message Service (SMS) Berbasis Android," *Teknoif*, vol. 4, no. 1, pp. 1–10, 2016.

[8] S. H. Putra, E. Santoso, and L. Muflikhah, "Implementasi Algoritma Kriptografi Advanced Encryption Standard (AES) Pada Komprsi Data Teks," 2012.

[9] V. Lusiana, "Implementasi Kriptografi Pada File Dokumen Menggunakan Algoritma Aes-128," *J. Din. Inform.*, vol. 3, pp. 79–84, 2011.

[10] A. R. Tulloh, Y. Permanasari, and E. Harahap, "Kriptografi Advanced Encryption Standard (AES) Untuk Penyandian File Dokumen Cryptography Advanced Encryption Standard (AES) for File Document Encryption sebagai agensi departemen perdagangan AS menetapkan sebuah standard kriptografi Standard (AES)," *Pros. Mat.*, vol. 2, no. 2460–6464, pp. 118–125, 2016.