KRIPTOGRAFI FILE MENGGUNAKAN METODE AES DUAL PASSWORD

***Imron Abdul Ilyas*[1]**
***Suryarini Widodo*[2]**

[1]*Jurusan Teknik Informatika, FTI, Universitas Gunadarma.*
[2]*Jurusan Sistem Informasi, FIKTI, Universitas Gunadarma.*
[2]***srini@staff.gunadarma.ac.id***

Abstrak

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 untuk menjaga kerahasiaan dan keaslian data serta melindungi data dari 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. Pada aplikasi ini, hampir semua jenis file dapat dienkripsi menggunakan algoritma AES ini seperti file gambar, teks, sura, video, aplikasi dan lain-lan. Algoritma AES adalah blok chipertext simetrik yang dapat mengenkripsi (encipher) dan mendekripsi (decipher) informasi menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekrip data pada blok 128 bits. Selain itu, AES mempunyai keunggulan dalam keamanan, kecepatan, dan karakteristik algoritma beserta implementasinya. .Hasil penelitian menunjukkan bahwa algoritma AES dengan panjang kunci 256 bit dapat menyandikan file sehingga dapat mengamankan file tersebut. Ukuran file hasil enkripsi tidak berubah dari file asli.

***Kata Kunci** : Advanced Encryption Standard, Dekripsi, Enkripsi, Kriptografi, Rijndael.*

PENDAHULUAN

Berkat perkembangan tekno-logi yang begitu pesat memung-kinkan manusia dapat ber-komunikasi dan saling bertukar informasi/data tanpa dihalangi oleh jarak. Seiring dengan itu tuntutan akan sekuritas (keamanan) terhadap kerahasiaan

informasi yang saling dipertukarkan tersebut semakin meningkat. Begitu banyak peng-guna seperti dalam institusi atau perusahaan atau bahkan individu-individu tidak ingin informasi yang disampaikannya diketahui oleh orang lain atau kompetitor-nya. Oleh karena itu dikembang-kanlah cabang ilmu yang mem-pelajari tentang cara-cara pengamanan data atau dikenal dengan istilah Kriptografi.

Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi/data yang hendak dikirim

diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awal-nya dengan menggunakan algorit-ma tertentu yang disebut *ciphertext*. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk ter-samar tersebut menjadi informasi awal yang kita kenal sebagai *plaintext.*

Advanced Encryption Standard (AES) merupakan algo-ritma *cryptographic* yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan mendekripsi (*decipher*) informasi. Algoritma AES ini menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekrip data pada blok 128 bits. Selain itu, AES mempunyai ke-unggulan dalam keamanan, ke-cepatan, dan karakteristik algorit-ma beserta implementasinya.

Metode AES Dual Password digunakan untuk lebih menjamin *file* tidak bisa dilihat oleh pihak yang tidak berhak atas *file* tersebut karena menggunakan dua buah kunci enkripsi dan dekripsi.

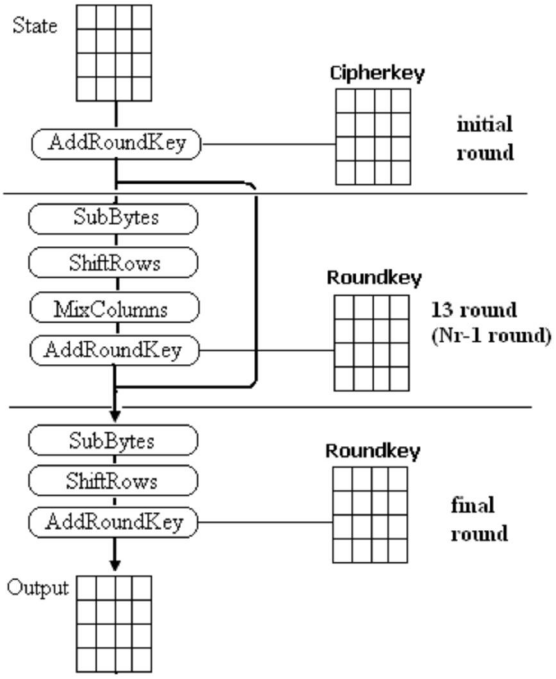
Tujuan yang ingin dilakukan dalam penulisan ini adalah membuat program aplikasi krip-tografi semua jenis *file* dengan metode AES Dual untuk menjaga kerahasiaan *file* agar tidak dapat diketahui oleh pihak yang tidak berhak *(anauthorized persons),* serta memudahkan pengguna untuk menyimpan *file* secara rahasia dan pribadi. Selain itu juga bisa menjadi acuan untuk pembuatan aplikasi keamanan yang serupa.

METODE PENELITIAN

Proses Enkripsi *Advanced Enryption Standard 256*

Proses enkripsi algoritma AES terdiri dari 4 jenis trans-formasi *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*, *Mix-Columns,* 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 meng-alami transformasi *Mi-xColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada gambar 1 di bawah ini.

Proses *AddRoundKey* pada enkripsi dan dekripsi AES adalah 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 *SubBytes* merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box) seperti terlihat pada tabel 1 berikut.



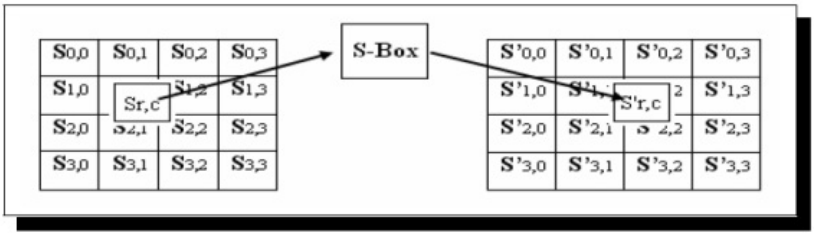
Gambar 1. Proses Enkripsi AES 256
Sumber : Rinaldi Munir (2006)

Tabel 1. Tabel SubBytes S-Box

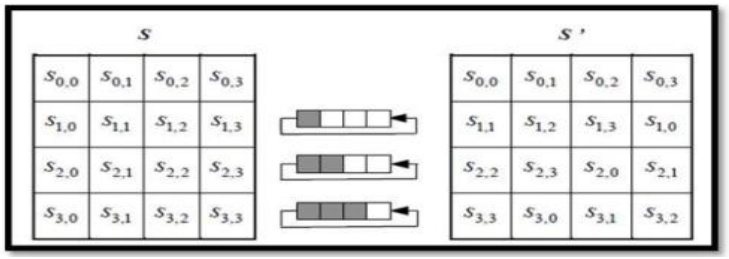
		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	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	24	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 2 mengilustrasikan pengaruh peme-taan *byte* pada setiap *byte* dalam *state*. Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran *bit* dimana *bit* paling kiri akan dipindahkan menjadi *bit* paling kanan. Proses pergeseran *Shiftrow* ditunjukkan dalam gambar 3 berikut. *MixColumns* mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi *mixcolumns* dapat dilihat pada perkalian matriks pada gambar 4.



Gambar 2 Pengaruh Pemetaan pada Setiap *Byte* dalam *State*
Sumber : Rinaldi Munir (2006)



Gambar 3 Transformasi *ShiftRows*
Sumber : Rinaldi Munir (2006)

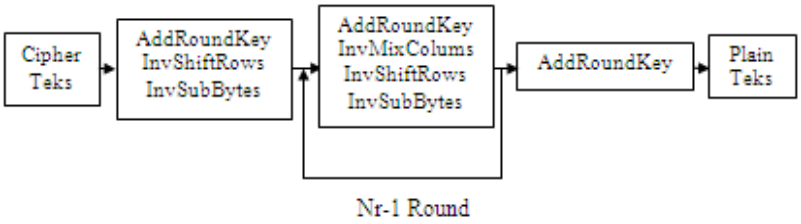
$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Gambar 4. Perkalian matriks

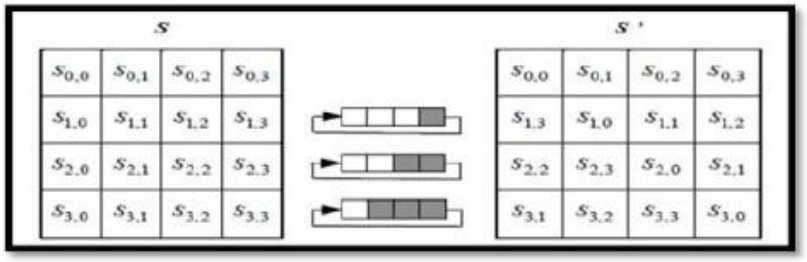
Proses Dekripsi *Advanced Enryption Standard 256*

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Trans-formasi byte yang digunakan pada *invers cipher* adalah *InvShift-Rows, InvSubBytes, InvMix-Columns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada gambar 5 berikut ini. *InvShiftRows* adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dila-kukan pergeseran *bit*

ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi *InvShift-Rows* terdapat pada gambar 6. *InvSubBytes* juga merupakan transformasi *bytes* yang ber-kebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada *state* dipetakan dengan menggunakan tabel *Inverse S-Box*. Tabel *Inverse S-Box* ditunjukkan pada tabel 2 berikut ini. Kemudian setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan seperti pada gambar 7.



Gambar 5 Proses Dekripsi AES 256
Sumber : Rinaldi Munir (2006)



Gambar 6 Transformasi *InvShiftRows*
Sumber : Rinaldi Munir (2006)

Tabel 2 *InvSubBytes*

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	C	d	e	f
X	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	ac	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	38	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

Sumber : Rinaldi Munir (2006)

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Gambar 7 Perkalian dalam matrik

HASIL DAN PEMBAHASAN

Hasil uji coba yang dilakukan terhadap sepuluh file dengan tipe yang berbeda dijelaskan sebagai berikut. File-file tersebut di-enkripsi dengan menggunakan dua *password* yaitu *password* satu dan *password* dua. Setelah file di enkripsi, maka file yang dienkripsi akan berubah ekstensi-nya menjadi .dualAES tetapi ukuran file tidak berubah seperti terlihat pada tabel 3. Untuk mendekripsi file-file tersebut diatas digunakan password dua *password* (kunci). yang sama dengan password ketika mengenkripsi file tersebut yaitu *password* satu dan *password* dua. Ekstensi *file* setelah didekripsi pun akan kembali seperti ekstensi file awal. Sedang untuk total jumlah byte enkripsi tidak sama dengan total jumlah byte dekripsi. Perbandingan ini berdasarkan total *byte* pada saat proses dienkripsi dan didekripsi menggunakan AES Dual Password. Untuk lebih jelasnya lihat tabel 4 berikut. Secara keseluruhan total byte dekripsi lebih banyak dibandingkan total byte enkripsi.

Tabel 3. Hasil Uji Coba Enkripsi File

Nama File	Ekstensi File Awal	Ukuran File Awal (KB)	Ekstensi File Hasil Enkripsi	Ukuran File Hasil Enkripsi (KB)
Abunawas	3DS Max Scene	136	dualAES	136
Black and yellow	Mp4 Video	16,84	dualAES	16,84
Eclipse-android	Shortcut	1	dualAES	1
Get Lucky	Mp3	5,82	dualAES	5,82
KRIPTOGRAFI	Word Document	93	dualAES	93
Lightroom_3_LS 11_win_3_6	Execute	242,95	dualAES	242,95
Logo	Adobe Photoshop	40,91	dualAES	40,91
Ralineshah	JPG	59	dualAES	59
RealSteel	WinRAR archive	53,24	dualAES	53,24
Rijndael	Text Document	3	dualAES	3

Tabel 4. Perbandingan Byte Enkripsi dan Dekripsi

Nama File	Ekstensi	Ukuran File (KB)	Byte Enkripsi	Byte Dekripsi
Abunawas	3DS Max Scene	136	139264	139280
Black and yellow	Mp4 Video	16,84	17242298	17242304
Eclipse-android	Shortcut	1	319488	319504
Get Lucky	Mp3	5,82	5954783	5954784
KRIPTOGRAFI	Word Document	93	95232	95248
Lightroom_3_LS 11_win_3_6	Execute	242,95	248776304	248776320
Logo	Adobe Photoshop	40,91	41895843	41895856
Ralineshah	JPG	59	59588	59600
RealSteel	WinRAR archive	53,24	54516381	54516384
Rijndael	Text Document	3	2270	2272

SIMPULAN DAN SARAN

Aplikasi kriptografi file dengan menggunakan AES Dual Password dapat menghasilkan sebuah file hasil enkripsi yang tidak dapat di buka oleh aplikasi apapun sehingga kerahasiaan file dapat terjaga, Aplikasi ini diterapkan untuk pada semua jenis file seperti file gambar, video, audio, teks, aplikasi, file hasil kompresi dan lain-lain. Dari hasil percobaan terlihat bahwa ukuran file hasil proses enkripsi dan dekripsi tidak berubah dan ukuran file akan mempengaruhi total pemrosesan *byte* pada proses enkripsi dan dekripsi. Metode AES memiliki ukuran *block* yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit. Hal yang membedakan dari masing-masing AES ini adalah banyaknya *round* yang dipakai. AES-128 mengg-unakan 10 *round*, AES-192 sebanyak 12 *round*, dan AES-256 sebanyak 14 *round.* Dengan menggunakan dua buah *password* (kunci) pada aplikasi ini akan lebih mengamankan file.

Sebagai saran untuk pengembangan aplikasi selanjut-nya adalah mengimplementasikan metode kritografi AES Dual Password ini untuk meng-enkripsi *folder.*

DAFTAR PUSTAKA

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

Anonim 2007 *128-Bit Versus 256-Bit AES Encryption* Seagate.

Benabdellah, Mohammed. Majid Himmi, Mohammed dan Zahid, Noureddine 2007 Encryption-Compression of Images Based on FMT and AES Algorithm *Applied Mathematical Sciences*, Vol. 1, 2007, No. 45, 2203 – 2219.

Eko Satria. 2009 *Study Algoritma RIJNDAEL Dalam Sistem Keamanan Data*. USU Repository : Medan.

Hollis, Billy 2014 *Tales From The Crypto* MSDN.

Munir, Rinaldi. 2006 *Kriptografi*. Informatika: Bandung.

Munir, Rinaldi 2004 *Advanced Encryption Standard (AES*). Institut Teknologi Bandung: Bandung.

National Institute of Standards and Technology 17 Mei 2014 *Announcing the ADVANCED ENCRYPTION STANDARD (AES)* http://csrc.nist.gov/publications/.

Thad Van den Bosch. 17 Mei 2014. *Encrypt/Decrypt Files in VB.NET (Using Rijndael)*. www.codeproject.com/Articles/12092/Encrypt-Decrypt-Files-in-VB-NET-Using-Rijndael.