JNATIA Volume 1, Nomor 1, November 2022
Jurnal Nasional Teknologi Informasi dan Aplikasinya

Pengembangan Aplikasi Berbasis Mobile Untuk Pengamanan Teks Menggunakan Metode Advanced Encryption Standard dan Least Significant Bit

Bhisma Satwika Ari Priandana[a1], I Made Widiartha[a2],

[a]Informatics Departmen, Udayana University
Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia
[1]bhisma7x@gmail.com
[2] madewidiartha@unud.ac.id

Abstract

The rapid development of technology is very influential in all aspects of life. Data is one of the most sensitive and important things in today's technological developments. In this regard, data security is very important for system developers and system users. Cryptography and steganography are options for securing data where cryptography supports security aspects such as data integrity, data confidentiality which makes messages have no real meaning and also eliminates suspicion of messages sent. The results showed that the combination of the AES cryptographic method and the LSB steganography method was very well used to secure messages and the results also showed that there was no difference between the original image and the encoded image.

Keywords: *Encryption, Cryptography, Steganography, AES algorithm, LSB method*

Abstrak

Perkembangan teknologi yang sangat pesat sangat berpengaruh dalam segala aspek kehidupan. Data menjadi salah satu hal yang sangat sensitif dan juga penting dalam perkembangan teknologi saat ini. Berkaitan dengan itu, keamanan data menjadi hal yang sangat penting bagi pengembang sistem dan juga pengguna sistem. Kriptografi dan steganografi menjadi pilihan untuk mengamankan data dimana kriptografi mendukung aspek keamanan seperti data integritas, kerahasiaan data yang menjadikan pesan tidak memiliki makna yang sebenarnya dan juga menghilangkan kecurigaan pesan yang dikirim. Hasil penelitian menunjukan penggabungan metode kriptografi AES dan juga metode steganografi LSB sangat baik digunakan untuk mengamankan pesan dan hasil juga menunjukan tidak ada perbedaan antara gambar asli dan gambar yang telah disisipkan.

Kata Kunci: Enkripsi, Kriptografi, Steganografi, algoritma AES, metode LSB

1. Pendahuluan

Seiring dengan perkembangan zaman, pertukaran informasi secara konvensional sekarang hampir sudah tertinggal dan digantikan dengan pertukaran informasi dunia maya dimana semuanya sudah dapat diakses melalui internet dengan bebas. Hampir semua orang sekarang sudah memiliki *smartphone* masing-masing dimana penggunaan smartphone sangat mempermudah perolehan informasi. Berkaitan dengan itu, keamanan pada sistem *smartphone* atau *mobile* juga sangat diperlukan.

Kriptografi adalah salah satu pilihan untuk mengamankan data dimana Kriptografi dapat mengamankan data atau pesan dengan cara mengenkripsi pesan, artinya pesan yang akan dikirimkan akan sangat jauh berbeda dengan pesan aslinya [1]. Kriptografi sudah sangat berkembang dari pertama kali ditemukan, dimana kriptografi modern saat ini sudah sangat sulit untuk diretas misalnya seperti kriptografi *advanced encryption standard* (AES) yang paling sering digunakan karena memiliki kombinasi keamanan efisiensi, fleksibilitas, dan kinerja yang baik pada perangkat keras dan perangkat lunak [2]. Akan tetapi, kriptografi juga memiliki kelemahan

tersendiri dimana pesan hasil enkripsi tersebut masih bisa dilihat meskipun dalam bentuk yang tidak beraturan, artinya ketika peretas tidak bisa memecahkan pesan, peretas akan mencoba untuk menghancurkan pesan tersebut seperti menambahkan beberapa kata, menghapus kata, atau mengacak pesan tersebut agar pihak yang dituju tidak menerima pesan dengan utuh [3].

Salah satu cara untuk menangani kekurangan tersebut adalah dengan cara menggabungkan kriptografi dengan steganografi dimana pesan yang sudah di enkripsi dapat disembunyikan di dalam data lain misalnya seperti gambar, audio, atau video. Dalam beberapa tahun terakhir sudah ada beberapa teknik steganografi yang dikembangkan, salah satunya adalah teknik steganografi *Least Significant Bits* (LSB) yang menyembunyikan data dengan menggunakan bit yang paling tidak signifikan dari gambar [4]. Karena mayoritas masyarakat sekarang sudah menggunakan perangkat *mobile* seperti *smartphone* baik itu *android* maupun IOS maka salah satu cara untuk mengembangkan sistem berbasis mobile adalah menggunakan *framework flutter*. Flutter dapat mengembangkan aplikasi *multiplatform* dimana artinya dalam satu *base code* sudah dapat mengembangkan aplikasi baik *android*, IOS, *web app*, maupun *desktop app* [5].

2. Landasan Teori

2.1 Kriptografi

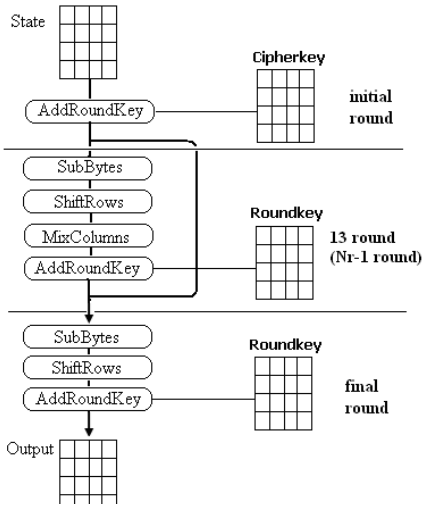
Kriptografi berasal dari bahasa yunani yaitu *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Secara umum kriptografi diartikan metode untuk menyandikan pesan sehingga pesan tidak memiliki makna yang berarti. Kriptografi memenuhi beberapa aspek keamanan diantara lain adalah *confidentiality* atau kerahasiaan data, *data integrity* atau keutuhan data, *authentication* atau autentikasi data dan juga *non repudiation* atau tidak dapat disangkal. Algoritma kriptografi dapat dibagi menjadi dua yaitu kriptografi simetris dan juga kriptografi asimetris. Algoritma simetris yaitu menyandikan pesan hanya dengan menggunakan satu kunci sedangkan kriptografi asimetris memerlukan dua kunci yaitu kunci publik untuk enkripsi dan kunci *private* untuk dekripsi.

2.2 Algoritma AES

Algoritma AES merupakan algoritma *chiper* yang aman untuk melindungi data atau informasi yang bersifat rahasia. AES dipublikasikan oleh NIST (National Institute of Standard and Technology) pada tahun 2001 yang digunakan untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dibobol. Input dan output dari algoritma AES terdiri dari urutan data sebesar 128-bit. 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 [6]. Panjang kunci ini akan memengaruhi panjang perputaran pada algoritma AES dimana jumlah perputaran juga akan memengaruhi tingkat keamanan dan ke acakan dari *cipertext*.

2.2.1 Proses Enkripsi AES

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, masukan yang telah disalik 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* [6]. Ilustrasi proses enkripsi algoritma AES dapat dilihat pada gambar 1.



Gambar 1. Proses Enkripsi AES

a. *AddRoundKey*
Tahapan *AddRoundKey* adalah tahapan paling awal dalam proses enkripsi dimana pada tahap permulaan, transformasi *AddRoundKey* dilakukan dengan kunci utama sedangkan untuk tahapan selanjutnya transformasi *AddRoundKey* dilakukan dengan *roundkey* atau kunci putaran. Tahapan *AddRoundKey* adalah operasi XOR antara *array state* dengan *round key*. Hasil dari proses *AddRoundKey* tidak akan mengubah matriks array akan tetap sama karena operasi XOR akan dilakukan pada masing-masing byte dalam array sehingga hasil akhir array pasti sama.

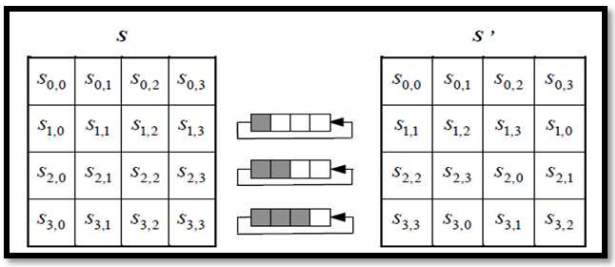
b. *SubBytes*
SubBytes merupakan transformasi dimana tiap byte pada array akan di substitusi kan dengan tabel S-Box. Tabel S-Box dapat dilihat pada gambar 2. Untuk setiap byte pada array state, misalkan S[r, c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r, c], maka nilai substitusinya, dinyatakan dengan S'[r, c], adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris x dengan kolom y. Gambar 3 mengilustrasikan pengaruh pemetaan byte pada setiap byte dalam state [6].

HEX		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

Gambar 2. S-Box

c. *ShiftRows*

Pada tahap *ShiftRows* hanya perlu menggeser baris dari matriks array ke sebelah kiri sebanyak baris ke-n-1 jadi bari 1 tidak digeser, baris 2 digeser satu kali dan seterusnya. Proses pergeseran *ShiftRows* dapat dilihat pada gambar 3.



Gambar 3. Transformasi *ShiftRows*

d. *MixColumns*

Transformasi *MixColumns* dilakukan setelah transformasi *ShiftRows*, merupakan sumber utama dari difusi pada algoritma AES. Difusi merupakan prinsip yang menyebarkan pengaruh satu bit plaintext atau kunci ke sebanyak mungkin ciphertext. Transformasi MixColumns() mengalikan setiap kolom dari array state dengan *polinom* a(x) mod (x4 + 1). Setiap kolom diperlakukan sebagai *polinom* 4 suku pada GF (28)[6].
Polinom a(x) yang ditetapkan pada persamaan 1.

$$a(x) = \{03\}x3 + \{01\}\,x2 + \{01\}\,x + \{02\} \quad (1)$$

Transformasi ini dinyatakan sebagai perkalian matriks seperti pada persamaan 2.

$$s'(x) = a(x) \otimes s(x) \quad (2)$$

Perkalian dalam matriks pada *MixColumns* dapat dilihat pada gambar 4. Hasil perkalian pada gambar 4 dapat dilihat pada persamaan ke 3.

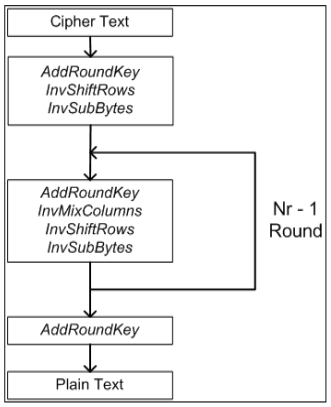
$$\begin{aligned} s'0,c &= (\{02\}\bullet s0,c) \oplus (\{03\}\bullet s1,c) \oplus s2,c \oplus s3,c \\ s'1,c &= s0,c \oplus (\{02\} \bullet s1,c) \oplus (\{03\} \bullet s2,c) \oplus s3,c \\ s'2,c &= s0,c \oplus s1,c \oplus (\{02\}\bullet s2,c) \oplus (\{03\}\bullet s3,c) \\ s'3,c &= (\{03\} \bullet s0,c) \oplus s1,c \oplus s2,c \oplus (*02\} \bullet s3,c) \end{aligned} \quad (3)$$

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

Gambar 4. Perkalian MixColumns

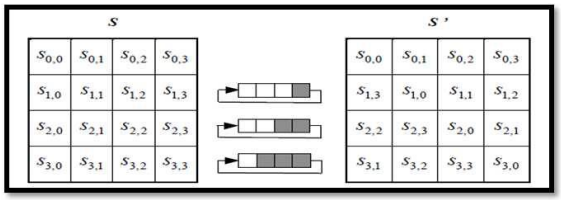
2.2.2 Proses Dekripsi AES

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 gambar 5.



Gambar 5. Tahap Dekripsi AES

a. *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. Proses *InvShiftRows* dapat dilihat pada gambar 6.



Gambar 6. Transformasi *InvShiftRows*

b. *InvSubBytes*
Pada tahap ini tidak jauh berbeda dengan tahap *SubBytes*, hanya saja pada tahap ini menggunakan *Inverse S-Box*. Gambar *Inverse S-Box* dapat dilihat pada gambar xx.

HEX		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	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

Gambar 7. Inverse *S-Box*

c. *InvMixColumns*
Inverse yang dilakukan dilakukan pada mixcolumns yaitu perkalian pada *InvMixColumns* berkebalikan dengan perkalian *mixColumns.* Perkalian pada *InvMixColumns* dapat dilihat pada gambar 8. Sedangkan persamaan untuk perkalian dapat dilihat pada persamaan 4.

$$s'0,c = (\{0E\}\bullet s0,c) \oplus (\{0B\}\bullet s1,c) \oplus (\{0D\}\bullet s2,c) \oplus (\{09\}\bullet s3,c)$$

s'1,c = ({09}•s0,c)⊕ ({0E} • s1,c) ⊕ ({0B} • s2,c) ⊕ ({0D}•s3,c)
s'2,c = ({0D}•s0,c)⊕ ({09}•s1,c)⊕ ({0E}• s2,c)⊕ ({0B}• s3,c)
s'3,c = ({0B}•s0,c) ⊕ ({0D}•s1,c)⊕ ({09}•s2,c) ⊕(*0E} • s3,c) (4)

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

Gambar 8. Perkalian *InvMixColumns*

2.3 Steganografi

Steganografi (steganography) berasal dari bahasa Yunani yaitu “steganos” yang berarti “tersembunyi” atau “terselubung”, dan “graphein” yang artinya “menulis”. Steganografi dapat diartikan “tulisan tersembunyi” (covered writing). Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Steganografi membutuhkan dua properti, yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video, atau teks. Pesan yang disembunyikan dapat berupa sebuah artikel, gambar, kode program, atau pesan lain. Proses penyisipan pesan ke dalam media covertext dinamakan encoding, sedangkan ekstraksi pesan dari stegotext dinamakan decoding. Kedua proses ini mungkin memerlukan kunci rahasia (yang dinamakan stegokey) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstraksi [6].

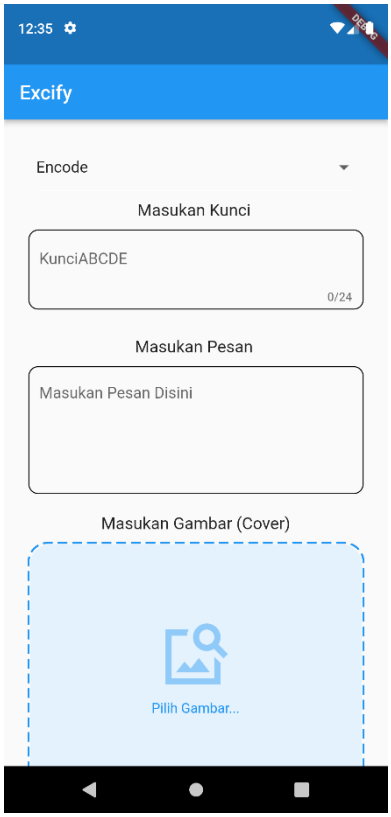
2.3.1 *Least Significant Bit*

Metode LSB merupakan metode steganografi yang paling sederhana dan mudah diimplementasikan. Metode ini menggunakan citra digital sebagai penampung pesan. Pada susunan bit di dalam *sebuah* byte (1 byte = 8 bit), ada bit yang paling depan (*most significant bit* atau MSB) dan bit yang paling akhir (*least significant bit* atau LSB). Sebagai contoh byte 11010010, angka bit 1 (pertama, digarisbawahi) adalah bit MSB, dan angka bit 0 (terakhir, digaris-bawahi) adalah bit LSB. Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan byte tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti. Mata manusia tidak dapat membedakan perubahan kecil tersebut.

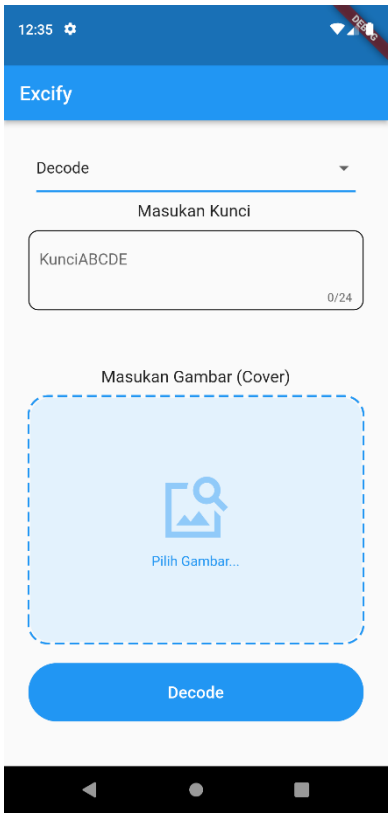
3. Implementasi dan Hasil

3.1 Implementasi

Implementasi yang digunakan untuk membuat aplikasi pengamanan teks menggunakan algoritma kriptografi AES dan steganografi LSB ini adalah menggunakan pengembangan aplikasi *multiplatform* yaitu *flutter* dengan menggunakan bahasa pemrograman *dart* dan menggunakan *tools visual studio code*. Gambar 9 merupakan gambar tampilan program untuk proses enkripsi dan penyisipan. Untuk melakukan proses enkripsi dan penyisipan, pertama masukan kunci rahasia yang hanya diketahui oleh pengirim dan penerima, lalu masukan pesan yang akan disisipkan ke dalam gambar, setelah itu masukan gambar yang akan menjadi penampung pesan berformat .bmp. Selanjutnya tekan tombol encode untuk mendapatkan hasil program berupa gambar yang telah disisipkan pesan rahasia. Gambar 10 merupakan gambar tampilan program untuk proses dekripsi dan ekstraksi. Untuk melakukan proses dekripsi dan ekstraksi masukan kunci dan juga pesan yang sudah disisipkan.



Gambar 9. Tampilan *Encoding* Aplikasi



Gambar 10. Tampilan *Decoding* Aplikasi

3.2 Hasil

3.2.1 Hasil Enkripsi

Setelah dilakukan percobaan enkripsi terhadap teks “saya sedang berada di london” dan dengan kunci “saya sedang makan” maka hasil dari enkripsi menggunakan AES-256 adalah e23c18494e142d098fd0607ce1752b9975db8f07b934c9339527957af7fa7756”. Dimana hasil dari enkripsi tersebut menjadi tidak berarti atau tidak memiliki makna lagi.

3.2.2 Hasil Penyisipan

Nama File	Size	Pesan yang dimasukan	Waktu Penyisipan
Gereja.bmp	786 kb	kemarin paman datang membeli peralatan memasak	1.62 detik
Lenna.bmp	768 kb	saya sudah tiba di london dua hari lagi saya akan pergi ke jakarta dan bali	1.98 detik
Fruits.bmp	394 kb	saya sedang berada di london	1.86 detik
Sawah.bmp	786 kb	orang itu sudah pergi dengan kerabatnya	1.97 detik
Pipit.bmp	786 kb	saya baru saja tertangkap polisi dan segera amankan barang barang kita	2.005

Setelah melakukan percobaan untuk proses enkripsi maka tahapan selanjutnya adalah tahap penyisipan dimana pesan yang sudah di enkripsi akan di sisipkan ke gambar penampung. Gambar 11 merupakan gambar penampung yang akan disisipkan pesan yang sudah ter enkripsi. Dimana gambar tersebut memiliki ukuran 512x512 pixel dengan ukuran file 800kb. Setelah

disisipkan dengan pesan “saya sedang berada di london” yang sudah ter enkripsi sehingga mendapatkan hasil yang bisa dilihat pada gambar 12. Dan untuk keseluruhan hasil penyisipan dapat dilihat pada tabel 1.

Gambar 11. Gambar Asli

Gambar 12. Gambar Hasil Penyisipan

Setelah dilakukan penyisipan pesan rahasia terhadap gambar penampung maka tampak tidak ada perbedaan antara gambar asli dan juga gambar hasil dari penyisipan jika dilihat dengan mata. Begitu pula dengan resolusi dan juga ukuran gambar.

3.2.3 Hasil Ekstraksi dan Dekripsi

Selain mencoba proses enkripsi dan juga penyisipan. Selanjutnya melakukan percobaan untuk ekstraksi dan juga dekripsi dimana proses ini dilakukan untuk mendapatkan pesan asli dari pengirim yang telah disisipkan ke gambar. Dimana dengan mengekstraksi gambar fruits.bmp maka hasil yang akan didapatkan adalah “saya sedang berada di london”.

4. Kesimpulan

Berdasarkan dari hasil dan pembahasan yang telah didapatkan maka dapat ditarik kesimpulan dimana algoritma *Advanced Encryption Standard* dan metode *Least Significant Bit* sangat baik untuk diimplementasikan untuk mengamankan teks yang kerahasiaan nya sangat dijaga. Dari hasil enkripsi, didapatkan teks yang sangat random atau tidak memiliki makna dan sulit untuk dimengerti. Pada proses penyisipan semakin panjang teks yang dimasukan atau disisipkan maka waktu untuk proses eksekusi penyisipan juga akan semakin lama hal itu disebabkan semakin banyak bit yang harus diganti pada pixel gambar. Hasil gambar dari proses penyisipan dan juga gambar asli tidak memiliki perbedaan jika dilihat dengan mata manusia. Hasil dekripsi dan juga hasil ekstraksi dapat dengan sukses mengembalikan pesan sesuai dengan pesan aslinya.

References

[1] P. Des, P. Kumar, and M. Sreenivasulu, Image Cryptography: A Survey towards its Growth. *Advance in Electronic and Electric Engineering*, 4.2, p.179-184. 2014.

[2] H. Alanazi, B. Zaidan, H. Jalab, M. Shabbir, and Y. A. Nabhani, New Comparative Study Between DES, 3DES and AES within Nine Factors. *JOURNAL OF COMPUTING*, 2.3**,** 2022.

[3] H. Abdulzahra, R. Ahmad, and N. NOOR, Combining Cryptography and Steganography for Data Hiding in Images. 2014.

[4] A. Reza, "PENERAPAN KRIPTOGRAFI MENGGUNAKAN ALGORITMA AES UNTUK DATA TEKS", 2017.

[5] F. Song, "Multi-Platform", *Flutter.dev*, 2022. [Online]. Available: https://flutter.dev/multi-platform. [Accessed: 02- Oct- 2022].

[6] V. Yuniati, G. Indriyanta and A. Rachmat C., "ENKRIPSI DAN DEKRIPSI DENGAN ALGORITMA AES 256 UNTUK SEMUA JENIS FILE", *Jurnal Informatika*, vol. 5, no. 1, 2011.

halaman ini sengaja dibiarkan kosong