APLIKASI PENGAMANAN PESAN TEKS PADA CITRA DIGITAL MENGGUNAKAN *ADVANCE ENCRYPTION STANDARD* 128 DAN *LEAST SIGNIFICANT BIT*-1

Jamelia Putri[1], Ernawati[2], Arie Vatresia[3]

[1,2,3]Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu
Jl. W.R. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(Telp: 0736-341022, Fax: 0736-341022)

[1]jameliaputri95@gmail.com
[2]ernawati@unib.ac.id
[3]arie.vatresia@unib.ac.id

Abstrak: Kriptografi adalah teknik menyandikan informasi menjadi bentuk yang tidak terbaca [1], sedangkan steganografi adalah teknik menyembunyikan informasi kedalam sebuah media digital [2]. AES 128 merupakan salah satu algoritme kriptografi yang handal mengatasi teknik pemecahan sandi [3], sedangkan algoritme steganografi LSB memiliki keunggulan dalam menstabilkan ukuran *stego image* dan menjaga kualitas *stego image* dengan baik [4]. Pengiriman pesan teks secara elektronik masih rentan terhadap serangan yang dapat memecah kerahasiaan pesan [5]. Penerapan AES 128 dan LSB-1 bertujuan untuk meningkatkan keamanan pesan. Pada penelitian ini, algoritme AES 128 dan inovasi algoritme LSB-1 telah dikombinasikan pada aplikasi pengamanan pesan teks pada citra digital. *Stego image* yang dihasilkan sistem berkualitas baik dan memiliki ukuran yang sama dengan *cover image* sehingga tidak menimbulkan kecurigaan dan pesan dapat diamankan. Algoritme AES 128 dapat menghasilkan *chipertext* dan mendekripsikan kembali menjadi *plaintext.* Sementara LSB-1 mampu meng-*embed chipertext* kedalam citra digital *(jpg, png, bmp, tif)* dan mengekstrak pesan yang tersembunyi dalam citra tersebut.

Kata kunci: Pengamanan pesan, kriptografi, steganografi, *AES 128, LSB-1.*

Abstract : **Cryptography is a technique of encoding information into an illegible form [1], whereas steganography is a technique of hiding information into a digital media [2]. AES 128 is a reliable cryptographic algorithm that overcomes password-solving techniques [3], while LSB steganography algorithm has the advantage of stabilizing the size of the stego image and maintaining the quality of the stego image properly [4]. However, sending text messages electronically is still vulnerable to attacks that can break the confidentiality of messages [5]. The application of AES 128 and LSB-1 aims to improve message security. In this study, the AES 128 algorithm and the LSB-1 algorithm innovation have been combined in the application of securing text messages on digital images. The Stego image produced by the system is of good quality and has the same size as the cover image so that it does not arouse suspicion and the message can be secured. The AES 128 algorithm can generate ciphertext and decrypt it into a plaintext. While LSB-1 is able to embed chipertext into digital images (jpg, png, bmp, tif) and extract messages hidden in that image.**

Keywords: **message security, cryptography, steganography, AES 128, LSB-1.**

I. PENDAHULUAN

Dukungan era digital, proses bertukar pesan atau informasi semakin mudah dan cepat. Dalam kondisi tertentu, pesan dapat bersifat rahasia atau disebut dengan Informasi Yang Dikecualikan dalam UU Keterbukaan Informasi Publik(KIP) Pasal 17, seperti informasi tentang kepentingan pribadi, informasi kepentingan komersil, informasi kebijakan pemerintah, dll. Namun, pertukaran pesan atau informasi secara elektronik rentan terhadap serangan yang dapat memecah kerahasiaan pesan. Untuk itu perlu dilakukan pengamanan agar pesan atau informasi tidak diketahui oleh pihak yang tidak berwenang maupun dimodifikasi.

Kriptografi adalah teknik menyandikan informasi menjadi bentuk yang tidak terbaca [1]. Metode kriptografi yang digunakan pada penelitian ini adalah metode AES 128 Bit, dengan penggunaan kunci yang panjang yakni 128 bit membuat algoritme AES 128 ini cukup handal mengatasi berbagai teknik kriptanalisis. AES memiliki kesederhanaan desain dan fleksibel disemua *platform software* dan *hardware* [3].

Pengamanan pesan dapat digandakan dengan menggunakan teknik steganografi. Steganografi merupakan seni untuk menyembunyikan pesan didalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan didalam media tersebut [2]. Algoritme yang digunakan untuk menyisipan pesan rahasia ke dalam media digital adalah modifikasi dari metode LSB yakni *Least Significant Bit-1* (*LSB-1*) yang menyisipkan bit rahasia di bit ke-7 dari 8 bit penyusun. Pada penelitian ini menggunakan citra digital sebagai *cover-image* dan pesan teks sebagai *embedded-message*. LSB-1 memiliki kelebihan pada ukuran wadah penampung yang tidak berubah sehingga tidak menimbulkan kecurigaan dan jika citra dilihat dengan kasat mata tidak terdapat perubahan warna yang berarti karena hanya 1 bit posisi terakhir yang berubah [4].

Kombinasi steganografi dan kriptografi ini akan diterapkan pada bahasa pemrograman MATLAB berbasis desktop dengan tampilan yang *userfriendly* dan dapat digunakan untuk mengamankan pesan dalam kehidupan sehari-hari.

Berdasarkan pemaparan latar belakang di atas maka penelitian ini mengangkat judul "Aplikasi Pengamanan Pesan Teks Pada Citra Digital Menggunakan *Advance Encryption Standard (*AES*)* 128 dan *Least Significant Bit-1* (LSB-1)".

II. TINJAUAN PUSTAKA

A. Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi [1]. Pesan atau informasi yang dapat dibaca disebut sebagai *plaintext* atau *cleartext*. Teknik untuk membuat pesan menjadi tidak dapat dibaca disebut sebagai enkripsi. Pesan yang tidak dapat dibaca disebut *chipertext*. Proses untuk mengembalikan *chipertext* menjadi pesan asli yang terbaca disebut dekripsi.

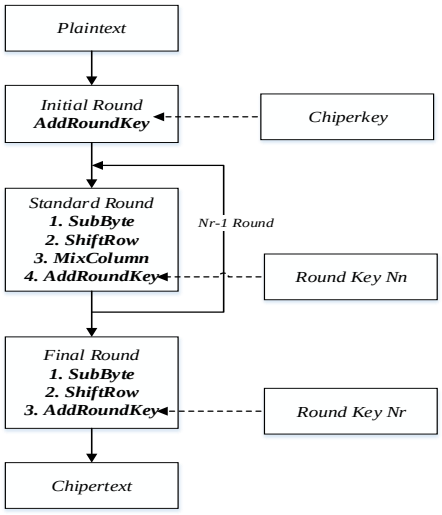
B. Advance Encryption Standard (AES) 128

Algoritme kriptografi AES adalah blok *ciphertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Algoritme AES beroperasi menggunakan blok *chipertext* 16 *byte* dan kunci kriptografi 128, 192, atau 256 bit. Perbedaan panjang kunci akan memiliki panjang *round* AES yang berbeda pula [3]

C. Enkripsi AES

Enkripsi AES menggunakan operasi substitusi, permutasi dan sejumlah putaran (*cipher* berulang), setiap putaran mengunakan kunci internal yang berbeda (kunci setiap putaran disebut *round key*) dan beroperasi dalam orientasi *byte.*

AES 128 menggunakan panjang kunci N_k=4 *word* yang setiap *word* nya berisi 32 bit sehingga total kuncinya 128 bit. Dengan panjang kunci 128-bit, maka terdapat sebanyak 2^{128}=3,4x10^{38} kemungkinan kunci dan jika digunakan komputer tercepat yang dapat mencoba 1 juta kunci setiap milidetik, maka akan dibutuhkan waktu 5,4 x 10^{18} tahun untuk mencoba seluruh kemungkinan kunci. Blok diagram enkripsi AES secara umum sebagai berikut:



Gambar 1. Blok Diagram Enkripsi AES 128

1. *AddRoundKey*: melakukan xor antara state awal(*plaintext*) dengan *cipher key*. Tahap ini disebut juga *initial round.*
2. Putaran sebanyak Nr –1 kali(pada AES 128=9 kali). Proses nya antara lain:

a. SubBytes

Mentransformasi byte setiap elemen pada *state* dan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box).

b. ShiftRows

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran bit, yaitu bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Transformasi ini diterapkan pada baris 2, baris 3, dan baris 4. Baris 2 akan mengalami pergeseran bit sebanyak satu kali, sedangkan baris 3 dan baris 4 masing-masing mengalami pergeseran bit sebanyak dua kali dan tiga kali.

c. *MixColumns*

Proses mengacak data di masing-masing kolom *array state*. *MixColumns* mengoperasikan setiap elemen yang berada dalam satu kolom pada *state*. Elemen pada kolom dikalikan dengan suatu polinomial tetap $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$.

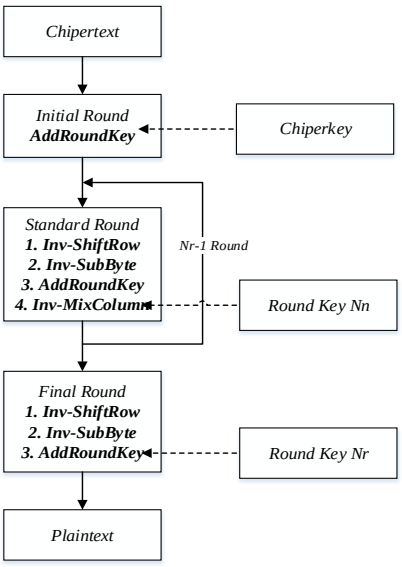
d. AddRoundKey

Proses *AddRoundKey*, sebuah *round key* ditambahkan pada *state* dengan operasi bitwise XOR. Setiap *round key* terdiri dari *Nb word*, tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state*

3. *Final round*. Tahap putaran terakhir enkripsi akan dilakukan 3 proses yaitu *SubBytes*, *ShiftRows* dan *AddRoundKey*.

D. Dekripsi AES

Dekripsi akan mengubah *chipertext* menjadi *plaintext*, sehingga pesan yang diterima dari pengirim dapat terbaca. Berikut diagram dekripsi AES:



Gambar 2. Blok Diagram Dekripsi AES 128

1. *AddRoundKey* : melakukan xor antara *state* awal (*cipherteks*) dengan *cipher key*. Tahap ini disebut juga *initial round*.
2. Putaran sebanyak Nr- 1(10-1=9) kali. Proses yang dilakukan pada setiap putaran :
   a. *InvShiftRow*
      Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan.
   b. *InvSubByte*
      Pada *InvSubBytes*, tiap elemen pada state dipetakan dengan menggunakan *Inverse S-Box*.
   c. *AddRoundKey* : melakukan operasi xor antara *state* sekarang dengan *round key*.
   d. *InvMixColumn* : mengacak data pada masing-masing kolom *array state* dengan cara mengalikan *array* polinomial dan *array state.*
3. *Final round*. Tahap putaran terakhir proses dekripsi akan dilakukan 3 proses yakni *InvShiftRow*, *Inv SubByte* dan *AddRoundKey.*

E. Ekspansi Kunci

Sebelum melakukan *AddRoundKey*, terdapat proses pembangkitan kunci yang dilakukan berulang sebanyak Nr. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Hasil *key schedule* terdiri dari *array* 4 *byte word* linear yang dinotasikan dengan $[w_i]$.

F. Steganografi

Steganografi merupakan seni untuk menyembunyikan pesan didalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan didalam media tersebut [2]. Steganografi membutuhkan dua properti, yaitu wadah

penampung dan data rahasia yang akan disembunyikan.

G. Least Significant Bit-1(LSB-1)

LSB-1 adalah modifikasi dari metode LSB, LSB menyembunyikan data dengan cara mengganti bit-bit data yang paling kurang berarti(bit terakhir) di dalam segmen citra dengan bit-bit data rahasia [2]. Sedangkan LSB-1, bit akan disisipkan pada bit ke 7 dari 8 bit penyusun piksel citra.

Contoh: Ada 3 piksel dari *image* 24 bit :
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
Akan disembunyikan karakter A (10000001)
Hasil : (00100**1**1 11101**0**1 11001**0**0)
(00100**0**1 11001**0**0 11101**0**1)
(11001**0**0 00100**1**1 11101001)

Dengan modifikasi letak penyisipan bit, *stegofile* lebih tahan terhadap penghancuran pesan, kualitas citra tetap baik dan ukuran *file* tidak berubah [4].

III. METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini adalah penelitian terapan, yaitu suatu kegiatan yang sistematis dan logis dalam rangka menemukan sesuatu yang baru dari penelitian yang telah dilakukan selama ini [6].

B. Teknik Pengumpulan Data

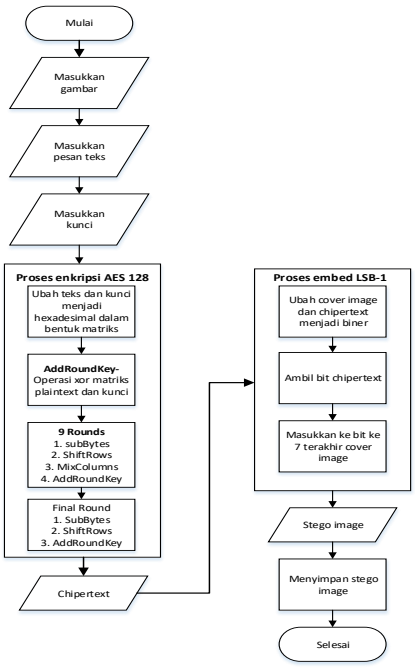
Sumber data citra digital diperoleh dari internet dan pesan teks diperoleh dari buatan penulis sendiri. Dalam penelitian ini menggunakan citra sebanyak 40 citra. Literature-literatur yang digunakan sebagai berikut: jurnal, skripsi, dan buku referensi

IV. ANALISA DAN PERANCANGAN

A. Analisis Alur Kerja Sistem

1. Alur sistem penyembunyian pesan

Pada Gambar 3, alur kerja sistem dimulai dari memasukkan gambar, pesan teks dan kunci 16 karakter.



Gambar 3. Alur Kerja Sistem Proses Penyembunyian Pesan

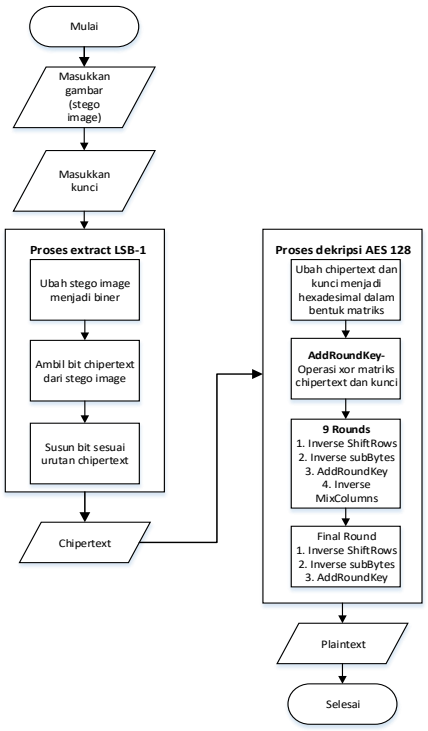
a. Proses enkripsi
   1) Konversi pesan teks dan kunci menjadi bentuk hexadecimal.
   2) *AddRoundKey*.
   3) *Round*, putaran sebanyak 9 kali. Proses yang dilakukan pada setiap putaran adalah *SubBytes*,. *ShiftRows, MixColumns*, *AddRoundKey.*
   4) Final Round, putaran terakhir(ke-10). Melakukan proses *SubBytes, ShiftRows* dan *AddRoundKey.* Hasil dari *AddRoundKey* ke 10 ini merupakan *chipertext.*

b. Proses embed

   Langkah proses embed adalah mengubah *cover image* dan *chipertext* menjadi biner. Lalu ambil bit-bit dari seluruh *chipertext* dan masukkan ke dalam bit-bit *cover image.* Proses memasukkan nya adalah dengan mengganti bit ke-7 dari tiap *byte cover image* dengan bit-bit dari *chipertext.* Output proses ini adalah *stego image* yang akan disimpan di media penyimpanan.

2. Alur sistem ekstraksi pesan

Proses ekstraksi pesan terdiri dari 2 proses yakni proses ekstraksi dan proses dekripsi.



Gambar 4. Alur Kerja Sistem Proses Ekstraksi Pesan

Pada Gambar 4, alur kerja sistem dimulai dari memasukkan gambar(*stego image)* dan kunci 16 karakter.

1. Proses ekstraksi

Ubah file *stego image* menjadi bentuk biner, lalu ambil bit ke-7 dari tiap *byte stego image* dan susun kembali bit-bit menjadi urutan *chipertext* semula.

2. Proses dekripsi.

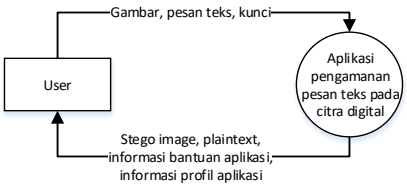
a. Chipertext dan kunci akan diubah menjadi bentuk hexadecimal.
b. *AddRoundKey*.
c. *Round*, sebanyak 9 kali (*InverseShiftRows, InverseSubBytes, AddRoundKey, InverseMixColumns*)
d. Final Round, putaran terakhir(ke-10). Melakukan proses *InverseShiftRows, InverseSubBytes* dan *AddRoundKey*. Hasil dari *AddRoundKey* ke-10 ini merupakan *plaintext.*

B. Perancangan Sistem

Perancangan aplikasi menggunakan *Data Flow Diagram* (DFD) sebagai salah satu *tool* atau model untuk merancang pengembangan perangkat lunak.

1. Diagram Konteks

Diagram konteks menggambarkan proses inti dan hubungan sistem dengan lingkungan disekitar sistem serta memberikan deskripsi sistem secara umum.



Gambar 5. Diagram Konteks

2. Diagram Level 1

Diagram level 1 akan memperinci proses yang terjadi di diagram konteks, didalam diagram level 1 terdapat 4 proses yaitu Penyembunyian Pesan, Ekstraksi Pesan, Bantuan dan Profil. Diagram Level 2

3. Diagram level 2 proses 1 Penyembunyian Pesan

Diagram level 2 proses 1 adalah penjelasan proses penyembunyian pesan. Diagram level 2 proses 1 menunjukkan terdapat 9 proses yakni input gambar, input pesan teks, input kunci, enkripsi AES 128, hitung ukuran plaintext, hitung ukuran chipertext, embed LSB-1, hitung ukuran stego image, hitung MSE dan PSNR, simpan stego image dan analisis lanjutan.

4. Diagram level 2 proses 2 Ekstraksi Pesan

Dalam diagram level 2 proses 2 terdapat 4 proses yang menjelaskan proses ektraksi pesan, prosesnya yakni input *stego image* inputkan kunci 16 karakter, ekstrak LSB-1, dekripsi AES 128.

V. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Implementasi antarmuka adalah tahap pembangunan sistem berdasarkan hasil analisis dan perancangan sistem.

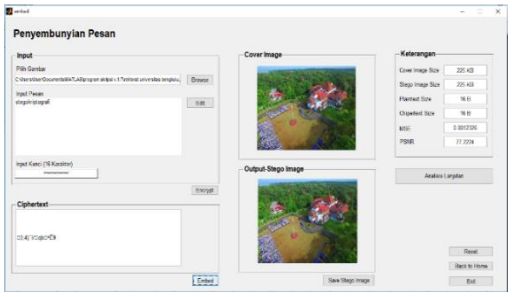
1. Halaman Utama Aplikasi

Gambar 6. Halaman utama

a. Halaman utama aplikasi adalah halaman pertama yang akan tampil saat sistem dijalankan. Berdasarkan Gambar 6 terdapat empat menu utama yaitu

b. Penyembunyian Pesan berfungsi untuk mengenkripsi pesan teks dan menyembunyikannya kedalam citra digital.
c. Ekstraksi Pesan berfungsi untuk mengekstrak pesan yang tersembunyi didalam citra digital dan mengembalikan pesan ke semula(dekripsi).
d. Bantuan berfungsi untuk menampilkan informasi petunjuk penggunaan aplikasi.
e. Profil berfungsi untuk menampilkan informasi identitas pembuat aplikasi.

2. Halaman Menu Penyembunyian Pesan



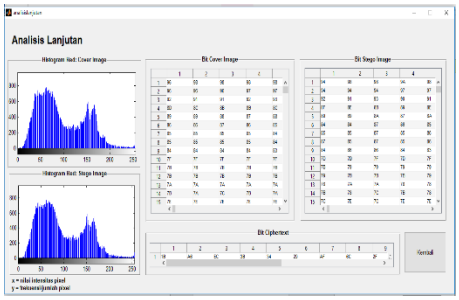
Gambar 96. Menu Penyembunyian Pesan

Pada Gambar 9, user dapat memulai dengan memilih gambar sebagai cover image dengan klik tombol *Browse* dan gambar akan tampil di panel *Cover image*. Kemudian input pesan yang diinginkan serta input kunci sebanyak 16 karakter. Pada pengujian ini pesan teks yang digunakan adalah "stegokriptografi" Tombol *Encrypt* berfungsi untuk menampilkan chipertext hasil enkripsi AES 128 yakni *"ESC¦ì;4)¯l/€qb"^È9"*.

Tombol *Embed* untuk menyembunyikan *chipertext* ke dalam *cover image* dan menampilkan output berupa stego image di panel *Output-Stego Image.*

Panel *Keterangan* menampilkan informasi ukuran *cover image*=225 KB, *stego image*=225 KB, *plaintext*=16 B, *chipertext*=16 B serta untuk mengukur kualitas *stego image* dilihat pada nilai MSE=0,0012326 dan PSNR=77,2224. Kualitas *stego image* yang baik memiliki PSNR >=40dB [7]. Stego image dapat disimpan dengan 4 pilihan ekstensi yaitu *.jpg, .png, .bmp, .tif.*

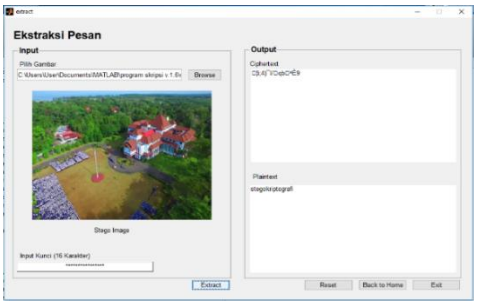
Tombol *Analisis Lanjutan* akan menampilkan halaman baru yang berisi histogram *cover image* dan *stego image* serta tabel yang berisi nilai bit dari *chipertext*, *cover image* dan *stego image.* Bit di tampilkan dalam bentuk bilangan hexadesimal yang dapat dilihat pada Gambar 10 berikut:



Gambar 10. Halaman Analisis Lanjutan

Penyisipan bit chipertext pada cover image tampak pada perubahan bit cover image dan bit stego image. Bit chipertext disisipkan ke bit ke-7 dari 8 bit penyusun cover image.

3. Halaman Menu Ekstraksi Pesan



Gambar 117. Menu Ekstraksi Pesan

Pada Gambar 11, terlebih dahulu user harus menginput stego image dengan klik tombol *Browse* dan gambar akan tampil dibawahnya. Kemudian menginput kunci sebanyak 16 karakter. Tombol *Extract* berfungsi untuk menampilkan chipertext *"ESC¦ì;4)¯l/€qb"^È9"* yang tersembunyi didalam *stego image* dan menampilkan *plaintext* atau pesan asli yaitu *"stegokriptografi"*.

B. Pengujian Black Box Sistem

Tabel 1. Pengujian Black Box

Kelas Uji	Aktifitas Pengujian	Hasil Yang Diharapkan	Hasil
Menu utama	Memilih menu penyembunyian pesan	Menampilkan menu penyembunyian pesan	Sukses
	Memilih menu esktraksi pesan	Menampilkan menu esktraksi pesan	Sukses
	Memilih menu profil	Menampilkan menu profil	Sukses
	Memilih menu bantuan	Menampilkan menu bantuan	Sukses
Menu penyembunyian pesan	Klik tombol Browse	Memilih gambar sebagai cover image	Sukses
	Input pesan dan kunci	Sistem menerima inputan pesan teks dan kunci	Sukses
	Klik tombol Encrypt	• Sistem mengenkripsi pesan teks menggunakan algoritme AES 128 • menampilkan di kolom Chipertext	Sukses

	Klik tombol Embed	• Sistem menyisipkan chipertext ke dalam cover image dengan algoritme LSB-1 • menampilkan stego image	Sukses
	Klik tombol Save Stego Image	Sistem dapat menyimpan stego image di folder yang kita inginkan	Sukses
	Klik tombol Analisis Lanjutan	Sistem menampilkan halaman Analisis Lanjutan	Sukses
	Klik tombol Reset	Sistem me-reset menu penyembunyian pesan	Sukses
	Klik tombol Back to Home	Sistem menuju ke halaman utama	Sukses
	Klik tombol Exit	Sistem akan menutup dan keluar dari aplikasi	Sukses
Halaman Analisis Lanjutan	Klik Kembali	Sistem akan kembali menuju menu penyembunyian pesan	Sukses
Menu ekstraksi pesan	Klik tombol Browse	Memilih gambar (stego image)	Sukses
	Input kunci	Sistem menerima inputan kunci	Sukses
	Klik tombol Extract	• Sistem akan meng-extract chipertext yang tersembunyi di stego image dengan LSB-1 • Menampilkan chipertext • mendekripsi chipertext menjadi plaintext • menampilkan plaintext	Sukses
	Klik tombol Reset	Sistem me-reset menu penyembunyian pesan	Sukses
	Klik tombol Back to Home	Sistem menuju ke halaman utama	Sukses
	Klik tombol Exit	Sistem akan menutup dan keluar dari aplikasi	Sukses

C. Pengujian Manual

1. Enkripsi AES 128

Plaintext : stegokriptografi
Kunci : zxcvbnmlkjhgfdsa

Konversikan plaintext dan kunci ke bilangan hexadecimal dan transformasikan ke dalam matriks 4x4.

Plaintext(hex)/*state* Kunci(hex)/*chiperkey*

$$\begin{bmatrix} 73 & 6F & 70 & 72 \\ 74 & 6B & 74 & 61 \\ 65 & 72 & 6F & 66 \\ 67 & 69 & 67 & 69 \end{bmatrix} \quad \begin{bmatrix} 7A & 62 & 6B & 66 \\ 78 & 6E & 6A & 64 \\ 63 & 6D & 68 & 73 \\ 76 & 6C & 67 & 61 \end{bmatrix}$$

a. Roundkey

Tabel 2. Roundkey

Round key	Chiperkey
0	7A 78 63 76 62 6E 6D 6C 6B 6A 68 67 66 64 73 61
1	38 F7 8C 45 5A 99 E1 29 31 F3 89 4E 57 97 FA 2F
2	B2 DA 99 1E E8 43 78 37 D9 B0 F1 79 8E 27 0B 56
3	7A F1 28 07 92 B2 50 30 4B 02 A1 49 C5 25 AA 1F
4	4D 5D E8 A1 DF EF B8 91 94 ED 19 D8 51 C8 B3 C7
5	B5 30 2E 70 6A DF 96 E1 FE 32 8F 39 AF FA 3C FE
6	B8 DB 95 09 D2 04 03 E8 2C 36 8C D1 83 CC B0 2F
7	B3 3C 80 E5 61 38 83 0D 4D 0E 0F DC CE C2 BF F3
8	16 34 8D 6E 77 0C 0E 63 3A 02 01 BF F4 C0 BE 4C
9	B7 9A A4 D1 C0 96 AA B2 FA 94 AB 0D 0E 54 15 41
10	A1 C3 27 7A 61 55 8D C8 9B C1 26 C5 95 95 33 84

b. AddRoundKey

= plaintext ⊕ Roundkey 0

$$= \begin{bmatrix} 73 & 6F & 70 & 72 \\ 74 & 6B & 74 & 61 \\ 65 & 72 & 6F & 66 \\ 67 & 69 & 67 & 69 \end{bmatrix} \oplus \begin{bmatrix} 7A & 62 & 6B & 66 \\ 78 & 6E & 6A & 64 \\ 63 & 6D & 68 & 73 \\ 76 & 6C & 67 & 61 \end{bmatrix}$$

$$= \begin{bmatrix} 09 & 0D & 1B & 14 \\ 0C & 05 & 1E & 05 \\ 06 & 1F & 07 & 15 \\ 11 & 05 & 05 & 08 \end{bmatrix}$$

c. 9 Round (R)

Tabel 3. Chipertext 9 Round

R-1	SubByte $\begin{bmatrix} 01 & D7 & AF & FA \\ FE & 6B & 72 & 6B \\ 6F & C0 & C5 & 59 \\ 82 & 6B & 63 & 30 \end{bmatrix}$	ShiftRow $\begin{bmatrix} 01 & D7 & AF & FA \\ 6B & 72 & 6B & FE \\ C5 & 59 & 6F & C0 \\ 30 & 82 & 6B & 63 \end{bmatrix}$
	MixColumns $\begin{bmatrix} 4A & F8 & FC & 55 \\ B3 & 5A & A3 & 25 \\ AB & 8A & A7 & 3A \\ CD & 56 & 38 & ED \end{bmatrix}$	AddRoundKey(Rk1) $\begin{bmatrix} 72 & A2 & CD & 02 \\ 44 & C3 & 50 & B2 \\ 27 & 6B & 2E & C0 \\ 88 & 7F & 76 & C2 \end{bmatrix}$
R-2	SubBytes $\begin{bmatrix} 40 & 3A & BD & 77 \\ 1B & 2E & 53 & 37 \\ CC & 7F & 31 & BA \\ C4 & D2 & 38 & 25 \end{bmatrix}$	ShiftRows $\begin{bmatrix} 40 & 3A & BD & 77 \\ 2E & 53 & 37 & 1B \\ 31 & BA & CC & 7F \\ 25 & C4 & D2 & 38 \end{bmatrix}$
	MixColumns $\begin{bmatrix} E6 & FF & 26 & 84 \\ 6A & 8D & 4E & F8 \\ 63 & 51 & 64 & DA \\ 95 & 34 & 98 & 8D \end{bmatrix}$	AddRoundKey(Rk-2) $\begin{bmatrix} 54 & 17 & FF & 0A \\ B0 & CE & FE & DF \\ FA & 29 & 95 & D1 \\ 8B & 03 & E1 & DB \end{bmatrix}$
R-3	SubBytes $\begin{bmatrix} 20 & F0 & 16 & 67 \\ E7 & 8B & BB & 9E \\ 2D & A5 & 2A & 3E \\ 3D & 7B & F8 & B9 \end{bmatrix}$	ShiftRows $\begin{bmatrix} 20 & F0 & 16 & 67 \\ 8B & BB & 9E & E7 \\ 2A & 3E & 2D & A5 \\ B9 & 3D & 7B & F8 \end{bmatrix}$

	MixColumns $\begin{bmatrix} 55 & 2E & C3 & A1 \\ EA & E2 & 3D & BE \\ 2F & 70 & 5F & C2 \\ A8 & F4 & 7F & 00 \end{bmatrix}$	AddRoundKey (Rk-3) $\begin{bmatrix} 2F & BC & 88 & 64 \\ 1B & 50 & 3F & 9B \\ 07 & 20 & FE & 68 \\ AF & C4 & 36 & 1F \end{bmatrix}$
R-4	SubBytes $\begin{bmatrix} 15 & 65 & C4 & 43 \\ AF & 53 & 75 & 14 \\ C5 & B7 & BB & 45 \\ 79 & 1C & 05 & C0 \end{bmatrix}$	ShiftRows $\begin{bmatrix} 15 & 65 & C4 & 43 \\ 53 & 75 & 14 & AF \\ BB & 45 & C5 & B7 \\ C0 & 79 & 1C & 05 \end{bmatrix}$
	MixColumns $\begin{bmatrix} A4 & 69 & 76 & DE \\ A5 & 39 & A4 & C1 \\ 70 & 11 & 65 & 96 \\ 4C & 6D & BE & D7 \end{bmatrix}$	AddRoundKey(Rk-4) $\begin{bmatrix} E9 & B6 & E2 & 8F \\ F8 & D6 & 49 & 09 \\ 98 & A9 & 7C & 25 \\ ED & FC & 66 & 10 \end{bmatrix}$
R-5	SubBytes $\begin{bmatrix} 1E & 4E & 98 & 73 \\ 41 & F6 & 3B & 01 \\ 46 & D3 & 10 & 3F \\ 55 & B0 & 33 & CA \end{bmatrix}$	ShiftRows $\begin{bmatrix} 1E & 4E & 98 & 73 \\ F6 & 3B & 01 & 41 \\ 10 & 3F & 46 & D3 \\ CA & 55 & B0 & 33 \end{bmatrix}$
	MixColumns $\begin{bmatrix} E7 & BB & DE & C5 \\ 13 & 2C & E0 & AC \\ 8D & F4 & DE & DA \\ 4B & 7C & 8F & 61 \end{bmatrix}$	AddRoundKey(Rk5) $\begin{bmatrix} 52 & D1 & 20 & 6A \\ 23 & F3 & D2 & 56 \\ A3 & 62 & 51 & E6 \\ 3B & 9D & B6 & 9F \end{bmatrix}$
R-6	SubBytes $\begin{bmatrix} 00 & 3E & B7 & 02 \\ 26 & 0D & B5 & B1 \\ 0A & AA & D1 & 8E \\ E2 & 5E & 4E & DB \end{bmatrix}$	ShiftRows $\begin{bmatrix} 00 & 3E & B7 & 02 \\ 0D & B5 & B1 & 26 \\ D1 & 8E & 0A & AA \\ DB & E2 & 5E & 4E \end{bmatrix}$
	MixColumns $\begin{bmatrix} 1D & D4 & E9 & 8A \\ A9 & 24 & 8E & E5 \\ C2 & B1 & F0 & B9 \\ 71 & A6 & C5 & 16 \end{bmatrix}$	AddRoundKey(Rk-6) $\begin{bmatrix} A5 & 06 & C5 & 09 \\ 72 & 20 & B8 & 29 \\ 57 & B2 & 7C & 09 \\ 78 & 4E & 14 & 39 \end{bmatrix}$
R-7	SubBytes $\begin{bmatrix} 06 & 6F & A6 & 01 \\ 40 & B7 & 6C & A5 \\ 5B & 37 & 10 & 01 \\ BC & 2F & FA & 12 \end{bmatrix}$	ShiftRows $\begin{bmatrix} 06 & 6F & A6 & 01 \\ B7 & 6C & A5 & 40 \\ 10 & 01 & 5B & 37 \\ 12 & BC & 2F & FA \end{bmatrix}$
	MixColumns $\begin{bmatrix} CC & D7 & D7 & 0F \\ 51 & 08 & 35 & 22 \\ A7 & DE & C4 & 3A \\ 89 & BF & 51 & 9B \end{bmatrix}$	AddRoundKey(Rk-7) $\begin{bmatrix} 7F & B6 & 9A & C1 \\ 6D & 30 & 3B & E0 \\ 27 & 5D & CB & 85 \\ 6C & B2 & 8D & 68 \end{bmatrix}$
R-8	SubBytes $\begin{bmatrix} D2 & 4E & B8 & 78 \\ 3C & 04 & E2 & E1 \\ CC & 4C & 1F & 97 \\ 50 & 37 & 5D & 45 \end{bmatrix}$	ShiftRows $\begin{bmatrix} D2 & 4E & B8 & 78 \\ 04 & E2 & E1 & 3C \\ 1F & 97 & CC & 4C \\ 45 & 50 & 37 & 5D \end{bmatrix}$
	MixColumns $\begin{bmatrix} E9 & 66 & A8 & A5 \\ BE & 63 & 19 & 89 \\ 27 & 69 & 83 & 3B \\ FC & 07 & 90 & 42 \end{bmatrix}$	AddRoundKey(Rk-8) $\begin{bmatrix} FF & 11 & 92 & 51 \\ 8A & 6F & 1B & 49 \\ AA & 67 & 82 & 85 \\ 92 & 64 & 2F & 0E \end{bmatrix}$
R-9	SubBytes	ShiftRows

	$\begin{bmatrix} 16 & 82 & 4F & D1 \\ 7E & A8 & AF & 3B \\ AC & 85 & 13 & 97 \\ 4F & 43 & 15 & AB \end{bmatrix}$	$\begin{bmatrix} 16 & 82 & 4F & D1 \\ A8 & AF & 3B & 7E \\ 13 & 97 & AC & 85 \\ AB & 4F & 43 & 15 \end{bmatrix}$
	MixColumns $\begin{bmatrix} 77 & 2D & 3C & AB \\ C3 & 2A & 95 & AC \\ 7E & C9 & F2 & 81 \\ CC & 3B & CO & B9 \end{bmatrix}$	AddRoundKey(Rk-9) $\begin{bmatrix} C0 & ED & C6 & A5 \\ 759 & BC & 01 & F8 \\ DA & 63 & 59 & 94 \\ 1D & 89 & CD & F8 \end{bmatrix}$

d. Final Round

Tabel 4. Final Round

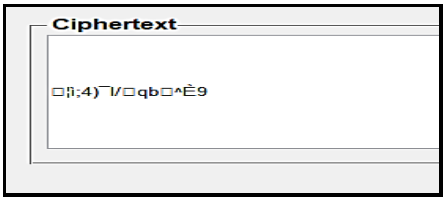
R-10	SubBytes $\begin{bmatrix} BA & 55 & B4 & \\ CB & 65 & 7C & \\ 57 & FB & CB & \\ A4 & A7 & BD & \end{bmatrix}$	ShiftRows $\begin{bmatrix} BA & 55 & B4 & 06 \\ 65 & 7C & 41 & CB \\ CB & 22 & 57 & FB \\ 41 & A4 & A7 & BD \end{bmatrix}$
	AddRoundkey(Rk-10) $\begin{bmatrix} 1B & 34 & 2F & \\ A6 & 29 & 80 & \\ EC & AF & 71 & \\ 3B & 6C & 62 & \end{bmatrix}$	

Maka chipertext nya adalah **1B A6 EC 3B 34 29 AF 6C 2F 80 71 62 93 5E C8 39**

Tabel 5. Konversi Chipertext Hex ke Char/Symbol

Chipertext **(hex)**	1B	A6	EC	3B	34	29	AF	6C
Chiper text **(char/ symbol)**	ESC	¦	ì	;	4	)	¯	l
Chipertext **(hex)**	2F	80	71	62	93	5E	C8	39
Chiper text **(char/ symbol)**	/	€	q	b	“	^	È	9

Chipertext hasil perhitungan manual dengan keluaran sistem menunjukkan hasil yang sama, seperti pada Gambar 12.



Gambar 12. Screenshoot Chipertext Pada Aplikasi

KESIMPULAN

A. Kesimpulan

Berdasarkan hasil dan pembahasan yang telah diuraikan sebelumnya, maka kesimpulannya sebagai berikut:

1. Penelitian ini berhasil menghasilkan aplikasi pengamanan pesan teks pada citra digital menggunakan AES 128 dan LSB-1.

2. Metode AES 128 berhasil mengenkripsi pesan teks menjadi bentuk yang tidak terbaca dan mendekripsikan kembali.
3. Penggunaan metode LSB-1 tidak merubah ukuran *cover image* dan *stego image* berkualitas baik.

REFERENSI

[1] S. Kromodimoeljo, Teori dan Aplikasi kriptografi, Jakarta: SPK IT Consulting, 2009.

[2] T. Sutoyo, E. Mulyanto, V. Suhartono, O. D. Nurhayati and W. , Tero Pengolahan Citra Digital, Semarang: ANDI Yogyakarta, 2009.

[3] Y. Kurniawan, Kriptografi Keamanan Internet dan Jaringan Komunikasi, Bandung: Informatika Bandung, 2004.

[4] Y. Andrian, "Perbandingan Metode LSB, LSB+1 Dan MSB Pada Steganografi Citra Digital," *STMIK Potensi Utama*, 2013.

[5] Wisnu, "Modus Email Fraud Dominasi Kejahatan Cyber di Indonesia," 21 Desember 2015. [Online]. Available: https://www.aktual.com/modus-email-fraud-dominasi-kejahatan-cyber-di-indonesia/.

[6] A. M. Yusuf, Metode Penelitian Kuantitatif, Kualitatif & Penelitian Gabungan, Jakarta: KENCANA, 2017.

[7] A. Cheddad, J. Condell, K. Curran and P. M. Kevitt, "Digital Image Steganography: Survey an Analysis of Current Methods," *Elsevier*, pp. 727-752, 2010.

[8] L. Hasugian, "Jenis-Jenis Format Penyimpanan Pada Citra," Medan, 2013.

[9] A. N. Setiadi, "Pemanfaatan Kriptografi AES," *Makalah Struktur Diskrit ITB*, 2009.

[10] A. Farmani and H. B. Bahar, "Hardware Implementation of 128-Bit AES Image Encryption with Low Power Techniques on FPGA," *Majlesi Journal of Electrical Engineering*, Vols. Vol. 6, No. 4, p. 15, 2012.

[11] N. o. I. a. S. Technology(NIST), "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication(FIPS PUB) 197, Virginia America, 2001.