p-ISSN : 2598-0416 e-ISSN : 2615-689X

JURNAL ARMADA INFORMATIKA

STMIK Methodist Binjai

jurnal.stmikmethodistbinjai.ac.id/jai

Keamanan Komputer

Steganography Menggunakan Advanced Encryption Standard dan Metode Least Significant Bit pada File Bitmap 24-bit

Marwa Halim[1]

Wulan Sri Lestari[2]

[1] *STMIK Methodist Binjai, Teknik Informatika, STMIK Methodist Binjai, Binjai, Indonesia*

[2] *Fakultas Informatika, Teknologi Informasi, Universitas Mikroskil, Medan, Indonesia*

INFORMASI ARTIKEL

Diterima Redaksi: 01 Desember 2023
Revisi Akhir: 05 Desember 2023
Diterbitkan *Online*: 07 Desember 2023

KATA KUNCI

Steganografi, LSB, Keamanan, Bitmap.

KORESPONDENSI

Phone: +6285100868120

E-mail: marwahalim@methodistbinjai.sch.id

A B S T R A K

Penelitian ini membahas penerapan teknik steganografi menggunakan *Advanced Encryption Standard* (AES) dan metode *Least Significant Bit* (LSB) pada file bitmap berwarna 24-bit. Tujuan utama penelitian ini adalah meningkatkan keamanan dan kapasitas penyimpanan pesan rahasia dalam gambar tanpa mengurangi kualitas visual. Proses steganografi dilakukan dengan menyisipkan bit-bit pesan rahasia ke dalam 2 bit terakhir (LSB) dari komponen warna pada setiap piksel gambar. Sebagai langkah tambahan, pesan rahasia dienkripsi menggunakan AES dengan melakukan rotasi 1 bit pada setiap karakter yang ada sebelum disisipkan, meningkatkan tingkat keamanan dan ketahanan terhadap deteksi. Eksperimen dilakukan dengan menggunakan berbagai citra bitmap 24-bit, dan hasil evaluasi mencakup analisis terhadap kapasitas penyimpanan, kekuatan enkripsi, dan dampak terhadap kualitas visual. Hasil penelitian menunjukkan bahwa kombinasi AES dan LSB mampu memberikan tingkat keamanan yang tinggi sambil mempertahankan kapasitas penyimpanan yang memadai dan meminimalkan perubahan visual yang terlihat. Temuan ini memiliki potensi aplikasi luas dalam keamanan komunikasi dan perlindungan informasi sensitif dalam domain steganografi.

PENDAHULUAN

Steganography adalah suatu teknik pengamanan informasi yang bertujuan menyembunyikan pesan rahasia dalam suatu media yang tampaknya biasa. Salah satu metode *steganography* yang umum digunakan adalah *Least Significant Bit* (LSB), di mana pesan rahasia disisipkan dalam bit paling tidak signifikan dari data gambar atau file lainnya. Meskipun metode ini dapat memberikan keamanan yang relatif baik, namun masih rentan terhadap serangan kriptografi dan analisis *steganalysis*.

Penting untuk mengembangkan metode *steganography* yang lebih canggih guna meningkatkan tingkat keamanan. *Advanced Encryption Standard* (AES) adalah algoritma kriptografi yang telah terbukti aman dan banyak digunakan dalam pengamanan data. Integrasi AES dengan metode LSB pada file bitmap 24-bit dapat memberikan tingkat keamanan yang lebih tinggi dan melindungi informasi yang disembunyikan dari serangan kriptografi.

File bitmap 24-bit dipilih karena dapat menyimpan informasi warna dengan resolusi yang tinggi, yang memungkinkan penyisipan pesan rahasia dalam berbagai tingkatan. Penggunaan metode LSB pada file bitmap 24-bit juga memberikan keuntungan dalam hal keaslian visual, di mana perubahan pada gambar yang dihasilkan tidak terlalu terlihat oleh mata manusia.

https://doi.org/

Dengan menggabungkan AES dan metode LSB pada file bitmap 24-bit, diharapkan dapat menciptakan suatu sistem *steganography* yang aman dan tangguh terhadap serangan. Penelitian ini bertujuan untuk menguji keefektifan dan keamanan dari metode tersebut, serta memberikan kontribusi terhadap pengembangan teknik *steganography* yang lebih baik.

Penelitian ini bertujuan untuk mengimplementasikan teknik *steganography* menggunakan metode LSB dan AES pada file gambar 24-bit bitmap. Penelitian ini menggunakan beberapa parameter untuk mengoptimalkan hasil *steganography*, yaitu:

- Tingkat penyisipan (*embedding rate*), yaitu persentase bit data rahasia yang disisipkan ke dalam gambar.
- Kualitas gambar, yaitu tingkat kejelasan gambar setelah disisipkan data rahasia.
- Keamanan, yaitu tingkat kesulitan untuk mendeteksi adanya data rahasia di dalam gambar.

TINJAUAN PUSTAKA

Steganografi dan Keamanan Informasi

Steganografi adalah suatu teknik atau seni penyembunyian informasi secara rahasia di dalam suatu media atau medium komunikasi tanpa menarik perhatian dari pihak yang tidak berkepentingan. Tujuan utama dari steganografi adalah menyembunyikan keberadaan pesan atau data tersembunyi sehingga orang yang tidak memiliki kunci atau pengetahuan khusus tidak dapat mendeteksinya.

Dalam konteks komunikasi digital, steganografi sering kali diterapkan untuk menyisipkan data tersembunyi dalam file gambar, audio, video, atau dokumen [1] lainnya tanpa mengubah secara signifikan penampilan visual atau auditorial dari media tersebut. Metode steganografi memanfaatkan kelemahan-kelemahan kecil pada format file atau media untuk menyembunyikan informasi tambahan.

Dibandingkan dengan kriptografi, yang fokus pada pengamanan data melalui enkripsi, steganografi menekankan pada aspek ketidak-diketahuan terhadap adanya pesan tersembunyi. Kombinasi kriptografi dan steganografi dapat meningkatkan tingkat keamanan komunikasi dengan tidak hanya melindungi isi pesan secara matematis (kriptografi), tetapi juga menyembunyikan keberadaan pesan (steganografi).

Advanced Encryption Standard

AES (*Advanced Encryption Standard*) adalah sebuah standar enkripsi yang digunakan secara luas untuk mengamankan data di dalam komunikasi dan penyimpanan. AES adalah salah satu algoritma kriptografi simetris, yang berarti kunci yang sama digunakan untuk melakukan enkripsi dan dekripsi. [2] Algoritma ini dipilih sebagai standar pengganti DES (*Data Encryption Standard*) oleh *National Institute of Standards and Technology* (NIST) pada tahun 2001.
Berikut adalah beberapa poin penting terkait dengan AES:

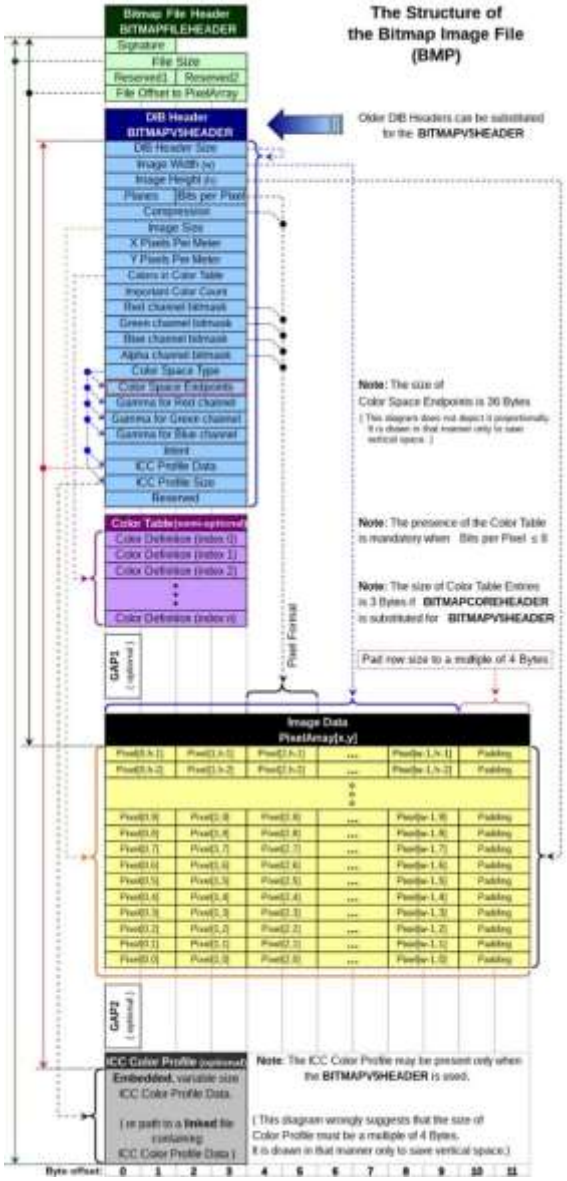
1. Kunci Simetris:
   AES menggunakan kunci simetris, yang berarti kunci yang sama digunakan baik untuk enkripsi maupun dekripsi. Kunci ini dapat memiliki panjang 128, 192, atau 256 bit, menawarkan tingkat keamanan yang sesuai dengan panjang kunci yang digunakan.
2. *Block Cipher*:
   AES adalah *block cipher*, yang berarti data dipecah menjadi blok-blok tetap sebelum dienkripsi atau didekripsi. Ukuran blok standar untuk AES adalah 128 bit.
3. *Ronde* dan *Subkey*:
   AES menggunakan serangkaian ronde (*rounds*) untuk mengolah data. Jumlah ronde tergantung pada panjang kunci: 10 ronde untuk kunci 128-bit, 12 ronde untuk kunci 192-bit, dan 14 ronde untuk kunci 256-bit. Setiap ronde melibatkan tahap-tahap khusus termasuk *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*.
   SubBytes: Melibatkan substitusi *byte* berdasarkan tabel substitusi *s-box*.
   ShiftRows: Melibatkan pergeseran baris dalam blok data.
   MixColumns: Melibatkan operasi matriks terhadap kolom-kolom blok data.
   AddRoundKey: Melibatkan operasi XOR antara blok data dan subkunci ronde.
4. Keamanan Tinggi:
   AES dianggap sebagai algoritma kriptografi yang sangat aman dan tahan terhadap berbagai serangan kriptoanalisis.

Keamanannya didasarkan pada kombinasi dari struktur matematis yang kompleks dan kekuatan kunci yang cukup besar.

Penggunaan AES telah menjadi standar *de facto* dalam berbagai aplikasi, termasuk komunikasi melalui internet, penyimpanan data, dan pengamanan informasi di berbagai industri. Keandalan dan efektivitasnya menjadikannya salah satu algoritma enkripsi yang paling banyak digunakan di seluruh dunia.

Citra BMP 24 Bit

Pada file citra pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (LSB) pada data *pixel* yang menyusun *file* tersebut. Untuk *file* bitmap 24 bit, setiap *pixel* (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (*byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. [3]



Gambar 1. Format *File* Bitmap 24-bit [4]

Sebagai contoh *file* gambar BMP 24 bit dengan warna merah murni dalam format biner akan terlihat sebagai berikut :

00000000 00000000 11111111
00000000 00000000 11111111

Sedangkan untuk warna hijau murni dalam format biner akan terlihat sebagai berikut:

00000000 11111111 00000000
00000000 11111111 00000000

Sedangkan untuk warna biru murni dalam format biner akan terlihat sebagai berikut:

11111111 00000000 00000000
11111111 00000000 00000000

Metode LSB (Least Significant Bit)

Metode yang digunakan untuk menyembunyikan pesan pada media digital tersebut berbeda-beda. Contohnya, pada berkas *image* pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (LSB) pada data *pixel* yang menyusun *file* tersebut. Pada berkas bitmap 24 bit, setiap *pixel* (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (*byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian, pada setiap *pixel* berkas bitmap 24 bit kita dapat menyisipkan 3 bit data. [5] Pada citra 24 bit, setiap piksel terdiri dari 3 *byte* yang merepresentasikan warna *red* (merah), *green* (hijau), dan *blue* (biru).

METODOLOGI

Dalam penelitian ini akan dilakukan dengan membedah format *file* jenis BMP untuk menentukan *byte* dimana yang dapat digunakan untuk menyisipkan data. Pada struktur *file* BMP terbagi dua bagian besar yaitu *header file* dan *color space* yang dapat digunakan nantinya dalam proses penyisipan data. Untuk melihat struktur *file* BMP dapat dilihat pada Gambar 1. Ada dua hal utama yang akan dilakukan antara lain penyisipan pesan/data ke *file* gambar dan membaca kembali pesan yang telah disisipkan pada *file* gambar yang telah diolah.

Tahap 1 : Menyisipkan pesan/data

Pada tahap ini pesan/data yang ada akan dilakukan proses enkripsi, metode yang digunakan pada tahap ini adalah metode AES dengan cara menlakukan rotasi bit dari karakter yang ada. Selanjutnya proses penyisipan pesan/data akan menggunakan metode LSB dengan menyisipkan masing-masing 2 bit dari karakter pesan/data ke dalam *byte* warna R, G dan B, dari setiap piksel warna citra yang ada.

Setelah selesai dilakukan penyisipan pesan/data proses selanjutnya akan diciptakan *file* gambar yang baru yang mana *file* gambar tersebut telah mengandung pesan/data yang ada seperti yang terlihat pada Gambar 2.

Gambar 2. Proses Penyisipan Data ke *File* Gambar

Tahap 2 : Membaca kembali pesan dari file gambar

Pada tahap kedua ini akan dilakukan pembacaan kembali pesan yang telah disisipkan ke *file* gambar yang dilakukan pada tahap pertama, sebelum dapat membaca pesan yang ada terlebih dahulu aplikasi yang ada harus dapat menentukan apakah gambar yang ada sudah mengandung pesan yang dimaksud. Jika gambar yang ada mengandung pesan maka proses pembacaan pesan dapat diteruskan, bit-bit yang berhasil dibaca akan dilakukan proses dekripsi dengan mengembalikan bit yang ada ke posisi semula, selanjutnya masing-masing bit akan dikonversi kembali ke pesan yang dimaksud seperti yang terlihat pada Gambar 3.

Gambar 3. Proses Membaca Pesan dari Gambar

Tahap 3 : Proses Pengujian

Pada tahap ini akan dilakukan pengujian terhadap hasil yang diperoleh apakah semua proses di atas sudah berjalan sesuai dengan baik dan benar. Proses pengujian ini akan membandingkan gambar asli dengan gambar stego dengan menghitung MSE (*Mean Squared Error*) dari kedua gambar. Semakin kecil nilai MSE akan menunjukkan semakin mirip dari kedua gambar tersebut. [6]

Berikut ini adalah rumus yang akan digunakan untuk menhitung MSE dari dua buah gambar

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$$

Dimana *n* mewakili jumlah titik data, adalah nilai sebenarnya, dan adalah nilai prediksi. [7]

HASIL DAN PEMBAHASAN

Setelah mengetahui *byte-byte* yang terdapat pada suatu *file* gambar BMP yang terbagi 2 bagian besar yaitu *header file* dan *color space* dimana *header file* menggunakan *byte* ke-0 sampai ke-53, maka pada *header file* ini tidak akan disisipkan pesan. Pesan akan disisipkan mulai dari *byte* ke-54 dan seterusnya. Untuk informasi gambar yang sudah disisipi pesan rahasia, dalam penelitian ini akan disisipkan kode **MM** sebagai kode bahwa *file* BMP tersebut telah mengalami penyisipan pesan/data yang ada. Kode MM akan disisipkan pada *byte* ke-54 sampai *byte* ke-61 dimana setiap *byte* akan disisipkan 2 bit dari masing-masing karakter MM tersebut. Berikutnya panjang *text* yang akan disisipkan pada bit terakhir sebesar 2 *byte* atau sebesar 256 x 256 = 65536 *byte*, dimulai dari *byte* ke 62 sampai 69. Untuk lebih jelasnya struktur *byte* yang akan disisipi pesan/data dapat dilihat pada Tabel 1 berikut ini.

Tabel 1. Struktur *file* BMP yang akan digunakan

Header Image	Kode (MM)	Panjang Pesan	Pesan
Byte ke 0-53	Byte ke 54-61	Byte ke 62 - 69	Byte ke 70 – selanjutnya

Proses enkripsi

Pesan yang ada sebelum dilakukan proses penyisipan akan dilakukan proses enkripsi dengan metode AES dengan melakukan rotasi bit sebesar 1 bit dari kanan ke kiri dari setiap karakter yang akan disisipkan.

Pesan yang telah melalui proses enkripsi akan disisipkan mulai dari *byte* ke 70 sebanyak 2 bit terakhir pada *byte* warna merah, hijau dan biru sehingga diharapkan tidak akan terlalu besar mengalami perubahan pada citra gambar karena hanya melakukan perubahan maksimal 3 dan 255 pada masing-masing *byte*.

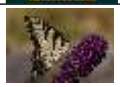
Tabel 2. Contoh rotasi 1 bit dari kanan ke kiri

Plain text	Biner	Rotasi 1	Cipher text
M	01001101	10100110	¦
e	01100101	10110010	²
t	01110100	00111010	:
h	01101000	00110100	4
o	01101111	10110111	·
d	01100100	00110010	2
i	01101001	10110100	´
s	01110011	10111001	¹
t	01110100	00111010	:

Hasil enkripsi selanjutnya akan diambil masing-masing 2 bit untuk disisipkan ke *byte image* mulai dari *byte* ke 70 dari *file image* asal. Proses selanjutnya hasil proses penyisipan ini akan ditulikan kembali menjadi *file image* baru dengan nama *file image* yang berbeda.

Pada penelitian ini, pengujian dilakukan dengan menggunakan *standard dataset image* berupa *file image* dengan format BMP 24 bit. Ada beberapa gambar yang akan dilakukan pengujian dengan panjang pesan/data menggunakan abstrak pada tulisan ini. Hasil pengujian kemiripan gambar asli dan gambar hasil penyisipan dapat dilihat pada tabel berikut ini.

Tabel 3. Hasil Pengujian *File* Asli dengan *File* Hasil

No	File Asli			File Hasil			Hasil MSE
	Nama File	Ukuran	File Size	Nama File	Ukuran	File Size	
1		211 x 249	154 KB		211 x 249	154 KB	0,1188
2		225 x 222	146 KB		225 x 222	146 KB	0,0932
3		696 x 497	0,89 MB		696 x 497	0,89 MB	0,0115
4		400 x 267	312 KB		400 x 267	312 KB	0,0407

Dari tabel 3, dapat dilihat semakin kecil nilai MSE menunjukkan angka mendekati 0 berarti kedua gambar mendekati kemiripan. Pada hitungan besaran MSE akan sangat dipengaruhi besarnya *file* gambar akan berbanding lurus pada besarnya pesan yang akan disisipkan.

Penggunaan 2 bit pada *file* gambar BMP yang masing-masing *pixel* mengandung 3 *byte* warna merah, hijau dan biru yang mana setiap *byte* mengandung nilai 255 dan penggunaan 2 bit hanya akan mempengaruhi dengan nilai maksimal 3 sehingga pada prinsipnya penggunaan 2 bit akan sangat sedikit mempengaruhi mutu dari citra pada gambar BMP.

KESIMPULAN DAN SARAN

Penelitian ini menginvestigasi penerapan steganografi dengan memanfaatkan *Advanced Encryption Standard* (AES) dan metode *Least Significant Bit* (LSB) pada file bitmap berwarna 24-bit. Penggunaan AES dalam mengenkripsi pesan rahasia sebelum disisipkan secara signifikan meningkatkan keamanan steganografi. Selain itu penggunaan Metode LSB dalam menyisipkan pesan rahasia pada file bitmap 24-bit tidak akan merubah kapasitas penyimpanan antara *file* asli dengan *file* hasil. Sedangkan dari sisi visual perubahan pada gambar setelah proses penyisipan pesan sangat minim, sehingga sulit terdeteksi secara kasual.

DAFTAR PUSTAKA

[1] N. A. Ramadhani dan I. Susilawati, "Penerapan Steganografi untuk Penyisipan Pesan Teks pada Citra Digital dengan Menggunakan Metode Least Significant Bit," *Jurnal Multimedia & Artificial Intelligence*, vol. 4, no. 1 , pp. 21-27, Februari 2020.

[2] R. Siringoringo, "ANALISIS PSNR PADA STEGANOGRAFI LEAST SIGNIFICANT BIT DENGAN PESAN TERENKRIPSI ADVANCED ENCRIPTION SYSTEM," *Jurnal METHODIKA*, vol. 2, no. 1, pp. 124-130, 2016.

[3] DAVID, A. MURTADO dan U. KASMA, "STEGANOGRAFI PADA CITRA BMP 24-BIT MENGGUNAKAN METODE LEAST SIGNIFICANT BIT," *Jurnal Ilmiah SISFOTENIKA*, vol. 2, no. 1, pp. 71-80, 2012.

[4] anonim, "BMP file format," Wikipedia, 29 November 2023. [Online]. Available: https://en.wikipedia.org/wiki/BMP_file_format. [Diakses 15 Januari 2024].

[5] A. Ardiansyah dan M. Kurniasih, "Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganograpi Menggunakan Metode Least Significant Bit," *Jurnal Teknologi Informasi*, vol. XIII, no. 3, pp. 96-101, 2018.

[6] Widiarti, R. R. Pertiwi dan A. Sutrisno, "Perbandingan Mean Squared Error (MSE) Metode Prasad-Rao dan Jiang-Lahiri-Wan Pada Pendugaan Area Kecil," *Seminar Nasional TEKNOKA*, pp. 56-60, 2017.

[7] A. Kumar, "Mean Squared Error or R-Squared – Which one to use?," Analytics Yogi, 29 Dec 2023. [Online]. Available: https://vitalflux-com.translate.goog/mean-square-error-r-squared-which-one-to-use/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc. [Diakses 30 1 2024].

[8] S. T. Veena dan S. Arivazhagan, "Forensic steganalysis for identification of steganography software tools using multiple format image," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 10, no. 3, pp. 188-197, 2021.