JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 13 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i1.5620

# IMPLEMENTASI STEGANOGRAFI CITRA DIGITAL LSB MENGGUNAKAN ENKRIPSI AES-256 DAN *EMBEDDING PSEUDORANDOM*

**Muhamad Akbar Firdaus[1*], Alam Rahmatulloh[1]**

[1]Informatika, Fakultas Teknik, Universitas Siliwangi

Received: 5 Desember 2024
Accepted: 14 Januari 2025
Published: 20 Januari 2025

**Keywords:**
AES; Citra;
Embedding Pseudorandom;
LSB.

**Corespondent Email:**
alam@unsil.ac.id

**Abstrak.** Perkembangan teknologi digital yang pesat membuat keamanan informasi menjadi sangat penting. Steganografi citra digital merupakan salah satu teknik pengamanan data yang memungkinkan penyembunyian informasi dalam media gambar. Penelitian ini mengembangkan program steganografi yang menggabungkan metode Least Significant Bit (LSB) dengan enkripsi Advanced Encryption Standard (AES) dan embedding pseudorandom. Pendekatan ini mengenkripsi pesan menggunakan AES-256 sebelum menyisipkannya ke dalam bit-bit LSB dari citra cover dengan pola pseudorandom, menciptakan sistem keamanan berlapis. Hasil pengujian menunjukkan keberhasilan program dalam menyisipkan dan mengekstraksi pesan dari gambar. Penggunaan format PNG memungkinkan penyimpanan data yang lebih transparan tanpa kompresi lossy, sehingga kualitas gambar hasil penyisipan tetap terjaga dengan baik. Analisis menunjukkan bahwa panjang pesan yang disisipkan berpengaruh signifikan terhadap waktu proses embedding dan ekstraksi. Penggunaan AES-256 memberikan lapisan keamanan tambahan pada data yang disembunyikan.

**Abstract.** *The rapid development of digital technology makes information security very important. Digital image steganography is one of the data security techniques that allows information to be hidden in image media. This study develops a steganography program that combines the Least Significant Bit (LSB) method with Advanced Encryption Standard (AES) encryption and pseudorandom embedding. This approach encrypts messages using AES-256 before inserting them into the LSB bits of the cover image with a pseudorandom pattern, creating a layered security system. The test results show the success of the program in inserting and extracting messages from images. The use of the PNG format allows for more transparent data storage without lossy compression, so that the quality of the embedded image is maintained well. The analysis shows that the length of the inserted message has a significant effect on the embedding and extraction process time. The use of AES-256 provides an additional layer of security for the hidden data.*

## 1. PENDAHULUAN

Seiring dengan kemudahan pertukaran data digital, kebutuhan akan keamanan informasi menjadi semakin krusial. Berbagai ancaman keamanan seperti pencurian data, penyadapan, dan manipulasi informasi menjadi risiko yang nyata dalam komunikasi digital. Sebagai salah satu solusi pengamanan data, steganografi menawarkan pendekatan unik dengan menyembunyikan informasi rahasia di dalam media digital, seperti gambar, audio, atau video. Kebutuhan akan steganografi semakin meningkat guna mengoptimalkan keamanan dalam transmisi informasi. Steganografi

merupakan metode penyembunyian pesan atau data sensitif di dalam media lain sedemikian rupa sehingga terlihat seperti media biasa. Hanya pihak yang mengetahui kunci khusus yang dapat mengungkap pesan tersembunyi. Teknik ini biasanya melibatkan penggunaan dua jenis media berbeda untuk menciptakan lapisan keamanan tambahan. Satu media berfungsi sebagai penyimpanan informasi, dan yang lain berfungsi sebagai media pembawa informasi[1]. Dalam penelitian ini, peneliti menggunakan algoritma AES, *Advanced Encryption Standard* (AES) adalah standar enkripsi simetris yang digunakan untuk melindungi data. AES adalah algoritma enkripsi yang paling populer dan sering diterapkan secara global dalam berbagai sistem keamanan digital [2]. Dengan objek yang akan diteliti untuk penelitian ini yaitu citra digital PNG. Steganografi citra digital dapat digunakan untuk mengirimkan pesan atau informasi rahasia, sehingga perlu menguji ketahanan terhadap serangan. Faktor ketahanan atau kekokohan adalah salah satu ukuran kekuatan metode steganografi [3]. Metode yang digunakan dalam penelitian ini adalah Metode *Least Significant Bit* (LSB) merupakan salah satu teknik steganografi yang populer, namun memiliki kelemahan dalam hal keamanan dan ketahanan terhadap deteksi. Penelitian ini bertujuan untuk mengatasi kelemahan tersebut dengan mengimplementasikan metode steganografi yang menggabungkan enkripsi AES-256 dengan embedding pseudorandom, sehingga meningkatkan keamanan dan ketahanan terhadap berbagai serangan steganalisis. Pesan yang akan dimasukkan terdiri dari rangkaian karakter atau string yang membentuk satu kesatuan. Pesan ini dapat terdiri dari huruf, angka, dan simbol yang didasarkan pada kode ASCII [4]. ASCII, singkatan dari *American Standard Code for Information Interchange*, merupakan standar internasional yang sudah lama digunakan dalam teknologi komputer untuk menggambarkan karakter teks dalam format digital. Meskipun terdapat sistem pengkodean lain seperti *hexadecimal* dan *Unicode*, ASCII tetap menjadi pilihan yang universal dan kompatibel secara luas. Secara fundamental, ASCII menggunakan struktur 7-bit untuk mengkodekan berbagai karakter, namun dalam praktiknya, sistem ini diimplementasikan dalam format 8-bit dengan menambahkan satu bit signifikan di posisi tertinggi. Bit tambahan tersebut biasanya dimanfaatkan untuk menentukan prioritas dalam komunikasi digital. Karakter kontrol dalam ASCII dibagi menjadi lima kategori berdasarkan fungsinya, yakni komunikasi logis, pengendalian perangkat, pemisah informasi, ekstensi kode, dan komunikasi fisik. Kode ASCII ini sangat umum ditemukan pada papan ketik komputer atau berbagai perangkat digital lainnya [5].

[6]. Angga Aditya Permana dan Habib Amna (2022) melakukan penelitian Implementasi Steganografi File Citra Digital Menggunakan Metode *Least Significant Bit*, dari hasil penelitian tersebut didapat kesimpulan GUI steganografi dapat dikembangkan menggunakan metode *Least Significant Bit* (LSB), sebuah teknik untuk menyembunyikan informasi rahasia dalam file gambar digital. Proses ini melibatkan penggantian bit ke-8, 16, dan 24 pada representasi biner dari file gambar BMP 24-bit dengan representasi biner dari pesan yang ingin disembunyikan. Pendekatan ini menawarkan beberapa keuntungan, termasuk kemampuan untuk menyembunyikan informasi tanpa mengubah ukuran file gambar asli [7].

Dalam proses penelitian ini, penulis menggunakan beberapa referensi penelitian sebelumnya sebagai bahan acuan dan pertimbangan. Salah satu referensi yang dicantumkan adalah sebuah penelitian dengan topik "Penyembunyian Pesan Rahasia Pada Citra Digital dengan Teknik Steganografi Menggunakan Metode *Least Significant Bit* (LSB)". dari hasil penelitian yang didapat metode LSB dapat digunakan dengan sangat baik untuk menyembunyikan pesan rahasia kedalam sebuah gambar, dengan memiliki kekurangan karena metode tersebut mempunyai sifat *fragile* (mudah rusak) ketika mengalami rotasi, pembesaran, *cropping*, dan gangguan [1]. A.Muh.Ramadhani.S dan Tasrif Hasanuddin (2021) melakukan penelitian modifikasi LSB pada gambar sebagai data *hiding steganography* dari hasil penelitian tersebut didapat kesimpulan bahwa *image* yang digunakan dengan metode LSB diperoleh stego *image* yang baik dengan nilai MSE 0,0005 dB dan nilai PNSR 77,3737dB, nilai tersebut menghasilkan stego image yang sangat baik karena tidak ada *noise* dan tidak banyak

mengalami perubahan [8]. Pada penelitian Irvan Maulana Yusup (2020) melakukan penggabungan algoritma *caesar cipher* dan LSB untuk file dokumen, dari hasil penelitian tersebut didapat kesimpulan dari total 5 file dokumen semuanya bisa di enkripsi dengan baik, dan file yang bisa di *embedding* oleh perangkat lunak ini hanya 2 file, maka saat di *extraction* 2 file ini bisa dikembalikan [9]. Muhammad Azlansyah dan Budi Setiyono (2019) melakukan penelitian Penyisipan Pesan Pada Citra Digital Menggunakan *Metode Least Significant Bit*, dari hasil penelitian tersebut didapat kesimpulan dari ke 3 citra yang di ujikan untuk penyisipan pesan kedalam gambar dengan penyisipan teks 250 kata menghasilkan rata-rata nilai PSNR, hasil pengukuran menunjukkan tingkat kualitas citra setelah penyisipan pesan sebagai berikut: citra langit memiliki nilai 68.22975 dB, citra wajah mencapai 72.228575 dB, dan citra pemandangan menunjukkan 74.322525 dB. Lebih lanjut, proses ekstraksi mampu mengembalikan pesan ke kondisi aslinya tanpa mengalami perubahan signifikan.

**a. Steganografi (*Steganografhy*)**

Steganografi merupakan teknik dan metode untuk menyampaikan pesan secara tersembunyi, sedemikian rupa sehingga keberadaan pesan tersebut tidak diketahui atau tidak dapat dilihat secara langsung [10]. Sebuah ilustrasi sederhana dari konsep ini adalah ketika seseorang memilih sebuah gambar sebagai media pembawa informasi rahasia. Dalam proses ini, warna dari tiap *pixel* ke-100 pada gambar tersebut dimodifikasi secara halus untuk merepresentasikan huruf-huruf dalam alfabet. Perubahan yang dilakukan begitu minimal dan nyaris tak terdeteksi, sehingga gambar tetap terlihat normal tanpa adanya indikasi bahwa di dalamnya tersimpan informasi rahasia. Tanpa menggunakan alat khusus atau aplikasi pendeteksi steganografi, akan sangat sulit bagi pihak ketiga untuk menyadari adanya pesan tersembunyi dalam gambar digital tersebut, bahkan jika mereka mengamatinya dengan seksama [11]. Dalam konteks steganografi citra digital, *metode Least Significant Bit* (LSB) merupakan salah satu teknik penyembunyian data yang paling fundamental dan banyak diaplikasikan. Metode ini didasarkan pada kesederhanaannya dalam implementasi serta tingkat keamanan yang cukup memadai untuk menyembunyikan pesan rahasia. Prinsip kerja LSB melibatkan modifikasi bit paling tidak signifikan dalam representasi digital sebuah *pixel*, yang umumnya adalah bit ke-8 dalam urutan biner. Proses ini dapat dilakukan pada sebagian atau keseluruhan byte dalam citra, di mana bit-bit tersebut digantikan dengan bit-bit dari pesan yang ingin disembunyikan [12].

**b. Algoritma AES**

Pada 1997, dimulai kompetisi global untuk menentukan standar kriptografi baru pengganti DES. Dari 21 peserta awal, tersisa 5 finalis pada 1999: Serpent, MARS, Twofish, Rijndael, dan RC6, masing-masing dikembangkan oleh tim ahli kriptografi terkemuka. Pada tahun 2000, algoritma Rijndael, yang diciptakan oleh Dr. Vincent Rijmen dan Dr. Joan Daemen, terpilih sebagai pemenang. Rijndael dinilai unggul dalam aspek keamanan dan efisiensi implementasi, kemudian diresmikan sebagai *Advanced Encryption Standard* (AES). Nama "Rijndael" sendiri merupakan gabungan dari nama kedua penciptanya [13]. *Advanced Encryption Standard* (AES) telah muncul sebagai salah satu algoritma enkripsi paling andal dan terpercaya dalam bidang keamanan informasi digital. Algoritma ini telah diadopsi secara luas sebagai standar industri untuk pengamanan data. AES merupakan *evolusi signifikan* dari pendahulunya, Data *Encryption Standard* (DES), yang dikembangkan sebagai respons terhadap kemajuan teknologi hardware yang mampu membobol sistem keamanan DES. Inovasi AES menawarkan tingkat keamanan yang jauh lebih tinggi, membuatnya menjadi pilihan utama untuk melindungi informasi sensitif di era digital saat ini [14].

**c. Embedding Pseudorandom**

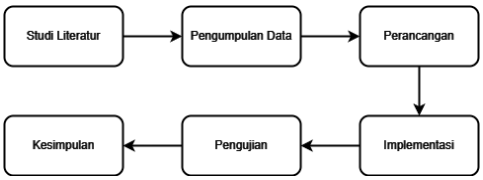
Proses *embedding* dalam steganografi ini memanfaatkan generator angka acak pseudo (RNG) untuk menentukan posisi *pixel* yang akan dimodifikasi, sehingga meningkatkan keamanan penyisipan pesan rahasia. Algoritma *Xorshift* digunakan sebagai RNG dengan parameter tertentu untuk menghasilkan posisi *pixel* yang acak dan tidak terduga. Penggunaan RNG dalam proses penyisipan pesan rahasia bertujuan untuk membuat posisi *pixel* yang dipilih menjadi tidak terduga, sehingga

meningkatkan keamanan dari pesan yang disisipkan dan mengurangi kemungkinan deteksi oleh analisis *steganalysis* [15].

## 2. METODE PENELITIAN

Dalam penelitian ini, penulis menerapkan pendekatan eksperimental untuk mengembangkan dan memvalidasi sebuah algoritma steganografi yang memanfaatkan citra digital. Algoritma tersebut menggunakan enkripsi AES-256 dan menerapkan teknik penyembunyian pesan secara *Pseudorandom Embedding*. Metodologi penelitian dirancang dalam beberapa tahapan yang sistematis.

### 3.1 Tahapan Penelitian



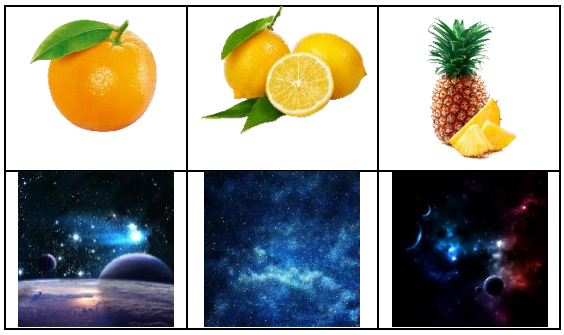

Gambar 1. Tahapan Penelitian

Gambar 1 menggambarkan kerangka konseptual penelitian yang dirancang untuk menguji penelitian terkait efektivitas teknik steganografi pada citra digital. Fokus utama penelitian adalah menilai bagaimana metode *Least Significant Bit* (LSB) yang dipadukan dengan enkripsi AES-256 dan teknik *Pseudorandom Embedding* dapat meningkatkan keamanan data rahasia. Tujuan utamanya adalah mengembangkan metode perlindungan pesan dengan cara menyembunyikan teks rahasia di dalam citra digital, sekaligus melindungi pesan dari potensi pencurian atau akses tidak sah.

### 3.2 Pengumpulan Data

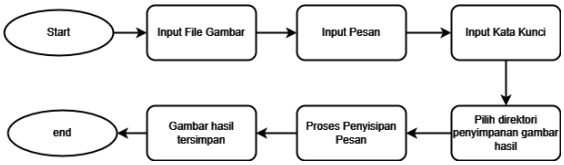
Pada penelitian ini penulis menggunakan data sekunder yang terdiri dari gambar digital berformat PNG dengan ukuran 500 × 500 piksel, yang diperoleh dari sumber-sumber daring sebagai citra penutup (*cover-image*). Selain itu, pesan teks yang digunakan memiliki panjang antara 10 hingga 15 karakter.

Tabel 1. Citra Digital berformat PNG



### 3.3 Flowchart Metode least significat bit (LSB)

a. *Embedding* (Penyisipan)

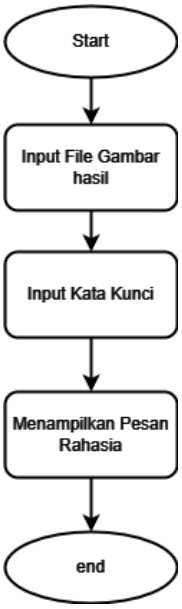



Gambar 2. Diagram alur proses penyisipan pesan

Dari gambar 2, Proses dimulai dengan menginisialisasi sistem, kemudian dilanjutkan dengan memasukkan file gambar yang akan digunakan sebagai media untuk menyisipkan pesan rahasia, dilanjutkan dengan memasukan pesan, kata kunci dan memilih di direktori mana gambar hasil akan disimpan, jika keempat proses sudah terpenuhi selanjutnya system melakukan proses *embedding* atau proses penyisipan pesan kedalam gambar, setelah itu system akan menyimpan gambar hasil ke direktori yang dipilih sebelumnya.

b. *Extraction* (Ekstraksi)

Dari gambar 3, Setelah sistem dimulai, proses berlanjut dengan memasukan gambar yang telah dimodifikasi dengan pesan rahasia. Selanjutnya, pengguna memasukkan kata kunci khusus yang digunakan dalam proses penyisipan pesan. Setelah langkah-langkah tersebut dilakukan, sistem akan menghasilkan tampilan pesan rahasia yang tersembunyi.




*Gambar 3. Diagram Alur Proses Extract*

### 3.4 Implementasi Kode

```
import cv2
import numpy as np
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from hashlib import sha256
import random
import sys
from skimage.metrics import structural_similarity as compare_ssim
import time
import tkinter as tk
from tkinter import filedialog, messagebox
import os
import hashlib
from tkinter import ttk
# Implementasi GUI dengan Tkinter
class SteganographyGUI:
    def __init__(self, master):
        pass
def main_gui():
    root = tk.Tk()
    gui = SteganographyGUI(root)
    root.mainloop()
if __name___= "__main__":
    main_gui()
```

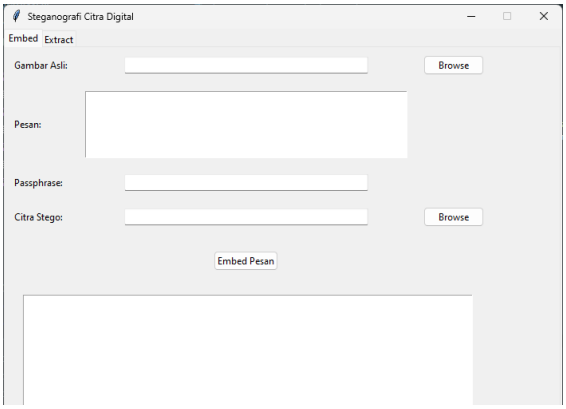
Implementasi dilakukan menggunakan Python dengan pustaka OpenCV untuk manipulasi citra, *PyCryptodome* untuk enkripsi AES-256, dan *Tkinter* untuk pengembangan GUI. Berikut adalah potongan kode utama yang digunakan dalam implementasi:

### 3.5 Pengembangan GUI

GUI dikembangkan menggunakan *Tkinter* dengan dua tab utama: *Embed* dan *Extract*. Antarmuka ini memungkinkan untuk memilih gambar asli, memasukkan pesan dan passphrase, menentukan lokasi penyimpanan gambar stego, serta melakukan embedding dan ekstraksi dengan mudah.

### 3.6 Implementasi

Tahap implementasi merupakan kelanjutan dari proses perancangan, di mana sistem yang telah dirancang dijalankan dalam kondisi nyata. Melalui tahap ini, akan dapat diketahui apakah sistem yang dikembangkan berhasil mencapai tujuan yang direncanakan sebelumnya [16].

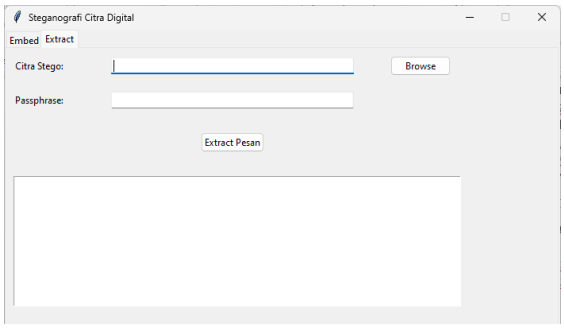



Gambar 4. GUI *Embedding*

Menu *embedding* pada gambar 5 akan muncul saat pengguna menekan tombol "*Embedding*". Di menu ini, pengguna dapat memilih gambar yang akan dijadikan sebagai wadah penyembunyian data (gambar asli), memilih file pesan yang ingin disembunyikan, dan memulai proses penyisipan. Selain itu, terdapat pula fitur untuk menyimpan hasil akhir dan kembali ke halaman utama. Sebagai tambahan, terdapat dua kolom teks dan dua panel yang berfungsi sebagai input kunci enkripsi.

Menu ekstraksi pada gambar 6 memungkinkan pengguna untuk mengambil kembali data rahasia yang telah tersembunyi dalam sebuah citra stego. Fitur utama pada menu ini meliputi pemilihan citra stego, input *passphrase*, dan tombol eksekusi ekstraksi.

Setelah proses ekstraksi selesai, data rahasia yang berhasil diambil akan ditampilkan pada antarmuka.




Gambar 5. GUI Extraction

Menu ekstraksi pada gambar 6 memungkinkan pengguna untuk mengambil kembali data rahasia yang telah tersembunyi dalam sebuah citra stego. Fitur utama pada menu ini meliputi pemilihan citra stego, input *passphrase*, dan tombol eksekusi ekstraksi. Setelah proses ekstraksi selesai, data rahasia yang berhasil diambil akan ditampilkan pada antarmuka.

## 3. HASIL DAN PEMBAHASAN

Elemen-elemen visual pada GUI telah dikembangkan sesuai dengan rincian yang ditentukan dan memenuhi kriteria yang ditetapkan dalam tahap perencanaan. Ini terbukti mampu menjalankan fungsinya dengan baik, yakni menyembunyikan informasi ke dalam sebuah gambar dan kemudian mengekstrak kembali informasi tersebut dari gambar yang sama.

### 4.1 Pengujian

#### 4.1.1 Black Box Testing

Pengujian pada tabel 2 dilakukan untuk memeriksa seluruh komponen GUI yang sudah dibuat,

Tabel 2. Pengujian GUI *Embedding*

| No | Input | Output | Keterangan |
|---|---|---|---|
| 1. | Menekan Tombol Browse | Memilih dan memasukan gambar yang akan disisipkan pesan | Berhasil |
| 2. | Memasukan pesan yang akan disisipkan | Pesan terlihat | Berhasil |
| 3. | Memasukan kunci enkripsi | Kunci berhasil dimasukan | Berhasil |
| 4. | Menekan tombol browse untuk memilih tempat menyimpan gambar yang sudah disisipkan pesan | Tempat penyimpanan berhasil dipilih | Berhasil |
| 5. | Menekan tombol Embedd pesan | Muncul keterangan | Berhasil |

Tabel 3. Pengujian GUI Extract

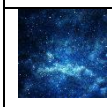
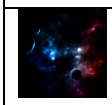
| No | Input | Output | Keterangan |
|---|---|---|---|
| 1. | Menekan tombol extract | Pindah ke menu extract | Berhasil |
| 2. | Menekan tombol *browse* | Memilih gambar yang sudah disisipkan pesan | Berhasil |
| 3. | Memasukan kunci enkripsi | Kunci berhasil dimasukan | Berhasil |
| 4. | Menekan tombol *extract* pesan | Muncul Pesan yang sudah di extract | Berhasil |

#### 4.1.2 Pengujian Penyisipan Pesan Dan Pengembalian Pesan

Hasil pengujian pada tabel 4 dan 5, menunjukkan keberhasilan dalam menyisipkan semua contoh pesan ke dalam gambar yang disediakan sebagai media penyimpanan. Setiap gambar yang telah disisipi pesan (*stego-image*) mampu menghasilkan kembali pesan asli dengan syarat kunci yang digunakan untuk mengekstrak pesan identik dengan kunci yang dipakai saat penyisipan, durasi proses

penyisipan dan ekstraksi pesan terbukti memiliki korelasi dengan jumlah karakter dalam pesan yang diproses.

Tabel 4. Hasil Pengujian Penyisipan Pesan

| Cover Image | Pesan | Kunci | Stego Image |
|---|---|---|---|
|  | Pesan Rahasia | Rahasia_ku | stego.png |
|  | Pesan Ini Rahasia | Rahasia1 | stego1.png |
|  | Rahasia Nanas | Rahasia | Stego2.png |
|  | Rahasia Planet1 | Rahasia2 | Stego3.png |
|  | Rahasia Planet2 | Rahasia3 | Stego4.png |
|  | Rahasia Planet3 | Rahasia4 | Stego5.png |

Tabel 5. Hasil Pengujian Status dan Durasi

| Status Embedd | Status Extract | Durasi Embedd | Durasi Extract |
|---|---|---|---|
| Berhasil | Berhasil | 0.5801 detik | 0.2382 detik |
| Berhasil | Berhasil | 26.0057 detik | 8.2187 detik |
| Berhasil | Berhasil | 0.6992 detik | 0.3039 detik |
| Berhasil | Berhasil | 1.5724 detik | 0.7026 detik |
| Berhasil | Berhasil | 0.6636 detik | 0.2970 detik |
| Berhasil | Berhasil | 1.7973 detik | 0.8256 detik |

## 4. KESIMPULAN

Setelah melakukan penelitian, dapat disimpulkan bahwa telah berhasil mengembangkan sebuah program penyisipan pesan ke dalam gambar. Program ini menggunakan metode *Least Significant Bit* (LSB) dengan menggabungkan algoritma enkripsi AES-256 dan teknik *Pseudorandom Embedding*. Selain itu, ditemukan bahwa panjang pesan yang disisipkan memberikan pengaruh signifikan terhadap durasi proses penyisipan dan ekstraksi pesan

Format PNG dipilih karena kemampuannya menyimpan data secara lebih transparan tanpa kompresi yang mengurangi kualitas, sehingga integritas gambar pasca penyisipan pesan tetap terpelihara. Analisis hasil uji coba mengindikasikan adanya hubungan yang kuat antara panjang pesan yang disembunyikan dengan waktu yang dibutuhkan untuk proses penyisipan dan ekstraksi. Implementasi enkripsi AES-256 meningkatkan tingkat keamanan informasi yang disembunyikan. Untuk riset mendatang, disarankan untuk menguji efektivitas metode ini pada format gambar alternatif seperti JPEG dan BMP, serta mengkaji ketahanannya terhadap berbagai teknik steganalisis.

## UCAPAN TERIMA KASIH

Penulis menyampaikan ungkapan terima kasih yang mendalam kepada berbagai pihak yang memberikan kontribusi dan dukungan dalam proses penelitian ini. Secara khusus, apresiasi ditujukan kepada Bapak Alam Rahmatulloh yang berperan sebagai pembimbing, serta rekan-rekan yang telah memberikan bantuan dan semangat selama kegiatan penelitian dan penulisan berlangsung.

## DAFTAR PUSTAKA

[1] Lasarus Pelipus Malese, “Penyembunyian Pesan Rahasia Pada Citra Digital dengan Teknik Steganografi Menggunakan Metode Least Significant Bit (LSB ),” *Jurnal Ilmiah Wahana Pendidikan* , vol. Vol. 7, 2021.

[2] Kania Sutisnawinata, “Advanced Encryption Standard (AES): Penjelasan Lengkap,” ASDF.ID.

[3] I. Pujianto and D. Darwis, “Uji Ketahanan Citra Digital Terhadap Manipulasi Robustness Pada Steganography,” 2021. [Online]. Available: http://jim.teknokrat.ac.id/index.php/informatika

[4] S. R. Saragih and P. Utomo, “Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks,” vol. 4, no. 1, 2020, doi: 10.30865/komik.v4i1.2691.

[5] Herri Setiawan, Bedy Briliant Wijaya, and Dewi Sartika, “Metode Spread Spectrum untuk Penyisipan Pesan pada Citra Digital,”

*Bulletin of Computer Science Research*, vol. 4, no. 1, pp. 101–111, Oct. 2023, doi: 10.47065/bulletincsr.v4i1.310.

[6] Muhammad Azlansyah and Budi Setiyono, "Penyisipan Pesan Pada Citra Digital Menggunakan Metode Least Significant Bit," 2019.

[7] A. Aditya Permana *et al.*, "Implementasi Steganografi File Citra Digital Menggunakan Metode Least Significant Bit," 2022.

[8] T. Hasanuddin, "Modifikasi Least Significant Bits pada Gambar sebagai Data Hiding Steganography," *Indonesian Journal of Data and Science (IJODAS)*, vol. 2, no. 2, pp. 91–102, 2021.

[9] I. M. Yusup, C. Carudin, and I. Purnamasari, "Implementasi Algoritma Caesar Cipher Dan Steganografi Least Significant Bit Untuk File Dokumen," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 3, Dec. 2020, doi: 10.28932/jutisi.v6i3.2817.

[10] I. Pujianto and D. Darwis, "Uji Ketahanan Citra Digital Terhadap Manipulasi Robustness Pada Steganography," 2021. [Online]. Available: http://jim.teknokrat.ac.id/index.php/informatika

[11] Buha Johannes Simbolon, "Steganografi Penyisipan Pesan Pada File Citra Menggunakan Metode LSB (Least Significant Bit)," 2021.

[12] A. P. Ratnasari and F. A. Dwiyanto, "Metode steganografi citra digital," vol. 2, no. 2, p. 52, 2020.

[13] M. Azhari, J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 1, pp. 2809–476, 2022, doi: 10.47709/jpsk.v2i1.1390.

[14] L. B. Handoko and C. Umam, "Pengamanan File Lampiran Pada Email Berbasis TLS Menggunakan Algoritma AES dan LSB Security of Attachment Files in TLS-Based Emails Using AES and LSB Algorithms," 2022.

[15] S. Rani, A. Kurniawardhani, and Y. A. W. Rendani, "Steganography on Digital Color Image Using Modulo Function and Pseudo-Random Number Generator," *Int J Adv Sci Eng Inf Technol*, vol. 11, no. 6, pp. 2470–2475, 2021, doi: 10.18517/ijaseit.11.6.12687.

[16] G. Miftakhul Fahmi, K. N. Isnaini, and D. Suhartono, "Implementation Of Steganography On Digital Image Wiht Modified Vigenere Cipher Algorithm And Least Significant Bit (LSB) Method," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 2, pp. 333–344, Mar. 2023, doi: 10.52436/1.jutif.2023.4.2.340.