Jurnal ICT : Information Communication & Technology
Vol. 25, N0.1, Juli 2025, pp. 95-101
p-ISSN: 2302-0261, e-ISSN: 2303-3363

Optimasi Ukuran File dalam Steganografi Gambar Menggunakan Metode Least Significant Bit dan Huffman Coding

Yuliana[1*], Rohmi Dyah Astuti [2], Ade Laelani[3], Linda Rassiyanti[4], Yusni Puspha Lestari[5], Ronal[6]
[1,2,3,4,5]Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera, Indonesia
[6]Program Studi Rekayasa Instrumentasi Automasi, Fakultas Teknik Industri, Institut Teknologi Sumatera, Indonesia
Email: [1]yuliana@sd.itera.ac.id, [2]rohmi.astuti@sd.itera.ac.id, [3]linda.rassiyanti@sd.itera.ac.id, [4]ade.lailani@sd.itera.ac.id, [5]yusni.lestari@sd.itera.ac.id, [6]ronal@ia.itera.ac.id

INFORMASI ARTIKEL

Histori artikel:

Naskah masuk, 1 Juli 2025
Direvisi, 28 Juli 2025
Diiterima, 31 Juli 2025

Kata Kunci:

Steganografi
Least Significant Bit
Optimasi file
Kompresi Huffman

ABSTRAK

Abstract- *Steganography is a technique used to conceal secret messages within digital media to maintain information confidentiality. This study applies a message embedding method using the Least Significant Bit (LSB) algorithm, in which secret messages are directly inserted into the least significant bits of grayscale image pixels. After the embedding process, the stego images are stored in three different formats—PNG, WebP, and ZIP—to evaluate the impact of compression on message integrity and image quality. The evaluation was carried out based on four parameters: file size, image quality degradation (PSNR), structural similarity (SSIM), and message extraction success. The results show that the PNG format can optimally preserve both image quality and message integrity (PSNR 75.58 dB, SSIM 1.0000), while lossy compression in WebP causes message corruption. The ZIP format successfully maintains the integrity of the stego file. This research focuses on evaluating the LSB method across various compression formats to optimize file size without compromising the integrity of hidden data.*

Abstrak- Steganografi merupakan teknik untuk menyembunyikan pesan rahasia dalam media digital guna menjaga kerahasiaan informasi. Penelitian ini menerapkan metode penyisipan pesan menggunakan algoritma Least Significant Bit (LSB), di mana pesan rahasia disisipkan langsung ke dalam bit-bit paling tidak signifikan dari piksel citra grayscale. Setelah proses penyisipan, citra stego disimpan dalam tiga format berbeda—PNG, WebP, dan ZIP—untuk mengevaluasi dampak kompresi terhadap integritas pesan dan kualitas citra. Evaluasi dilakukan berdasarkan empat parameter: ukuran file, degradasi kualitas citra (PSNR), kesamaan struktur visual (SSIM), dan keberhasilan ekstraksi pesan. Hasil menunjukkan bahwa format PNG mampu mempertahankan kualitas citra dan integritas pesan secara optimal (PSNR 75,58 dB, SSIM 1,0000), sedangkan kompresi lossy pada WebP menyebabkan kerusakan pesan. Format ZIP terbukti dapat menjaga file stego secara utuh. Penelitian ini fokus pada evaluasi metode LSB pada berbagai format kompresi untuk mengoptimalkan ukuran file tanpa mengorbankan keutuhan data tersembunyi.

***Penulis Korespondensi**:*

Yuliana
Sains Data, Fakultas Sains, Intitut Teknologi Sumatera,
Jalan Terusan Ryacudu, Way Huwi, Kecamatan Jati Agung, Kabupaten Lampung Selatan, Provinsi Lampung 35365, Indonesia.
Email: yuliana@sd.itera.ac.id

1. Pendahuluan

Perkembangan Teknologi komunikasi telah menyederhanakan dan mempercepat dalam pengiriman data. Namun kemudahan ini meningkatkan resiko penyadapan, penyalinan, modifikasi hingga perusakan data oleh pihak pihak yang tidak berwenang. Sehingga dibutuhkan teknologi dalam menjaga kerahasiaan data, baik saat data disimpan maupun saat data akan dikirimkan. Salah satu Teknik dalam penyembunyian data dalam mendukung kemanan jaringan dan melindungi informasi dari akses yang tidak sah adalah steganografi [1]. Steganografi adalah teknik dalam menyembunyikan pesan atau data rahasia dalam suatu wadah, seperti gambar, audio, teks, dan video sehingga keberadaan data tersebut tidak diketahui. Berbeda dengan kriptografi yang menyandikan sebuah pesan, Steganografi menyembunyikan keberadaan pesan itu sendiri. Dengan menyembunyikan pesan kedalam media digital tanpa merusak kualita aslinya secara signifikan menjadi metode yang efektif dalam menjaga privasi dan keamanan data, terutama ditengah komunikasi digital saat ini.

Salah satu tantangan dalam penerapan steganogarfi adalah ukuran file media yang digunakan sebagai wadah penyembuyian data. Semakin banyak data rahasia disembunyikan maka ukuran file cenderung meningkat, hal tersebut akan menjadi indikator mencurigakan terutama dalam system komunikasi. Perubahan ukuran file yang tidak wajar dapat memicu deteksi oleh system keamanan atau perangkat lunak analisis forensic digital.

Ukuran file yang besar dapat mempengaruhi efisiensi pengiriman, terutama jika bandwith terbatas atau transmisi dilakukan melalui jaringan nirkabel dan perangkat dengan keterbatasan sumber daya. Oleh karena itu Teknik kompresi dapat digunakan untuk mengurangi ukuran file tanpa merusak media atau menghilangkan data tersembunyi didaammnya. Integrasi antara teknik steganografi dan teknik kompresi tidak hanya menjaga kerasiahaan informasi tetapi juga memastikan keberhasilan pengiriman data dalam system komunikasi.

Teknik steganografi dalam menyembunyikan pesan adalah *Least Significant Bit* (LSB). *Least Significant Bit Least Significant Bit* LSB merupakan teknik penyisipan data rahasia kedalam bit tidak signifikan dari piksel untuk contoh media gambar atau video [2]. Karena perubahan bit paling tidak signifikan sehingga tidakberdampak terhadap kualitas media. *Least Significant Bit* dianggap sebagai metode yang sederhana namun efektif dalam menjaga tampilan asli media tetap utuh bagi pengamat biasa. Namun, meskipun teknik ini minim distorsi visual, kapasitas penyisipan tetap terbatas dan dapat menjadi permasalahan ketika volume data yang disembunyikan cukup besar atau ketika file mengalami peningkatan ukuran yang mencolok.

Penelitian sebelumnya yang dilakukan oleh Fadhlirrahman tahun 2023, tentang evaluasi performa metode *Least Significant Bit*. Pada penelitian tersebut metode *Least Significant Bit* dianggap sederhana, cepat, dengan kapasitas embedding yang cukup besar, dan masih efektif menyembunyikan pesan tanpa mudah terdeteksi. [3]. Penelitian lainnya dilakukan oleh Yasir 2023, menggunakan *Least Significant Bit* untuk meningkatkan ketahanan dan imperceptibility steganogarfi *Least Significant Bit* terhadap visual, dengan menyisipkan data hanya diarea berstektur tinggi. Dari penelitian yang dilakukan oleh Yasir di tahun 2024 melakukan kompresi menjadi suatu kebutuhan. [4]

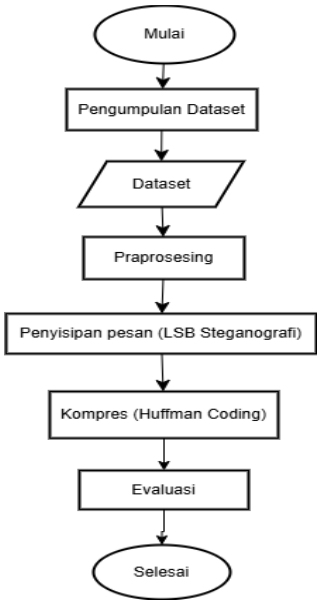
Kompresi merupakan proses pengurangan jumlah bit yang dibutuhkan untuk merepresentasikan informasi citra. Salah satu jenis kompresi adalah kompresi lossless. Kompresi lossless mempertahankan data asli secara utuh sehingga tidak ada informasi yang hilang. Salah satu metode kompresi lossless paling popular adalah kompresi Huffman coding [5]. Penelitian sebelumnya yang dilakukan oleh Shilpa tahun 2014. Tentan lossless image compression. Pada penelitian tersebut dikaji terkait mengurangi ukuran file tanpa kehiangan informasi, kulitas citra, rasio dan kecepaan dari proses kompresi.[6].

Mengompresi pesan dengan menggunakan Teknik Huffman coding membuat pesan rahasia menjadi lebih ringkas, sehingga mengurangi dampak terhadap ukuran file setelah proses steganografi dilakukan. Kombinasi *Least Significant Bit* dan Huffman coding memungkinkan peningkatan efisiensi dan keamanan, sekaligus menurunkan probabilitas deteksi oleh sistem forensik digital, karena ukuran file tidak mengalami kenaikan yang mencolok meskipun mengandung informasi tersembunyi.[7]

Berdasarkan penelitian sebelumnya penelitian yang dilakukan masih menerapakn LSB untuk menyisipkan pesan pada penelitian ini secara eksplisit dilakukan eksplorasi kombinasi metode LSB dengan Huffman coding untuk mengoptimalkan ukuran file hasil steganografi tanpa mengorbankan kualitas citra maupun integritas data tersembunyi. Fokus penelitian ini yaitu menghadirkan pendekatan terintegrasi yang mampu menyembunyikan pesan sekaligus meminimalkan dampak terhadap ukuran file sehingga mendukung distribusi data rahasia yang efisien dan aman.

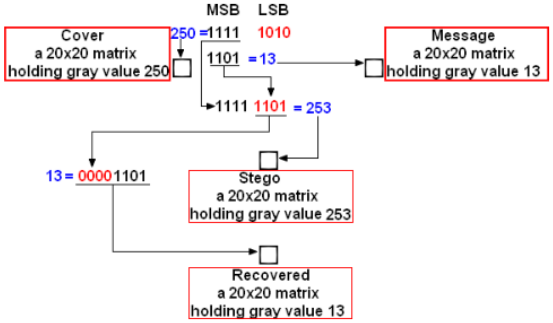
2. Metodologi

Penelitian ini menerapkan metode *Least Significant Bit* serta mengevaluasi efektivitas beberapa teknik kompresi file lossless. Adapun tahapan metodologi pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Metodelogi penelitian

Pada gambar 1 pengumpulan dataset diperoleh secara legal dari USC-SIPI Image Database. Dataset yang digunakan terdiri dari 20 citra grayscale resolusi 512x512 piksel [8]. Gambar dikonversi ke grayscale dan disesuaikan ukurannya secara seragam. Tahap selanjutnya adalah praprosesing, pada tahapan ini pesan teks yang akan disisipkan dikonversi terlebih dahulu kedalam biner. Pada penelitian ini pesan yang akan disisipkan kedalam gambar tidak akan melebihi kapasitas gambar. Selanjutnya penyisipan pesan kedalam gambar menggunakan Least Significant Bit. Mekanisme *Least Significant Bit* dapat dilihat pada gambar 2 [9].



Gambar 2. Mekanisme *Least Significant Bit*

Pada gambar 2 dapat dilihat mekanisme *Least Significant Bit* yang menggunakan media gambar berbasis pixel dengan nilai 8 bit (*gray value*). Setiap pixel yang terdiri dari 8bit akan dibagi menjadi dua bagian yaitu bagian MSB(*most significant bit*) dan LSB (*least significat bit*), untuk 4bit pertama yaitu bagian MSB dan 4bit berikutnya yaitu LSB. Pada penerapan *Least Significant Bit*, bit LSB akan diubah menjadi nilai dari pesan yang akan disisipkan. Setelah media dibubuhi pesan rahasia, setiap pixel dibangun kembali menjadi gambar yang utuh menyerupai dengan media gambar semula. Dalam proses penyembunyian data hal yang harus diperhatikan mulai dari kulaitas citra dari wadah yang tidak jauh berubah dan pesan yang disembunyikan harus dapat diekstrasi kembali. Tahapan penyisipan pesan kedalam gambar menggunakan *Least Significant Bit* dapat dilihat pada table 1 Pseudocode Algorithm: LSB_Embed

Tabel 1. LSB_Embed

Pseudocode Algorithm: LSB_Embed [10]

Input: image (array of pixel values), message (string)
Output: image_with_hidden_message
 	
Convert message to binary string → message_bits
Append '00000000' to message_bits as end-of-message marker
Set index ← 0
 	
For each pixel in image do
  If index ≥ length of message_bits then
  Break
  End If
 	
  bit ← message_bits[index]
  pixel ← (pixel AND 254) OR bit
  Replace pixel in image with new value
 	
  index ← index + 1
 	
End For
 	
Return image_with_hidden_message

Tahap berikutnya mengkompres gambar dalam format lossless. Gambar yang sudah disisipkan pesan akan disimpan dalam tiga format PNG, WebP dan ZIP. Format lossless ini dipilih untuk mempertahankan bit tanpa distorsi [10]. Pseudocode dari algoritma Huffman Coding dapat dilihat pada table 2.

Tabel 2. Huffman_Encode

Pseudocode Algorithm: Huffman_Encode [11]
Input: image_data (array of pixel values) Output: encoded_bitstream, huffman_tree 1. Count frequency of each pixel value in image_data

2. Create a priority queue of nodes where each node contains:
   - pixel_value
   - frequency
3. While priority queue has more than one node:
   a. Remove two nodes with lowest frequencies
   b. Create new parent node:
      - frequency = sum of two
      - left = node1, right = node2
   c. Insert new parent node into the queue
4. The remaining node is the root of the Huffman Tree
5. Traverse Huffman Tree to generate binary codes for each pixel value
6. Create a dictionary:
   - pixel_value
   - binary_code
7. Initialize empty bitstream
8. For each pixel in image_data: Append its binary_code to the bitstream

Return encoded_bitstream, huffman_tree

Tahap terkahir yaitu tahapan evaluasi. Tahapan evaluasi dilakukan berdasarkan ukuran file, degradasi kualitas citra, kesamaan struktur citra, dan ekstrasi pesan. Hasil evaluasi akan dilakukan analisis dari berbagai format penyimpanan untuk mengetahui metode mana yang paling efisen dalam ukuran file dan kualitas citra setelah penyisipan pesan.

3. Hasil dan Pembahasan

Implementasi penerapana *Least Significant Bit* untuk penyembunyian pesan rahasia dalam gambar dengan optimasi ukuran file menggunakan Bahasa pemrograman python, library utama yang digunakan pada penelitian ini PIL dan numpy.

3.1. Preprocessing

Pada tahap Preprocessing langkah pertama mengambil dan mengumpulkan dataset dari USC-SIPI. Citra pada dataset dikonversi ke format grayscale sehingga penyisipan hanya dilakukan pada 1 chanel. Berikut adalah potongan kode dalam tahap preprocessing:

Tabel 3. dataset hasil praprosesing

No	Nama File	Ukuran Gambar (px)	Panjang Pesan (bit)
1	lena.png	(512, 512)	392
2	peppers.png	(512, 512)	392
3	baboon.png	(512, 512)	392
4	goldhill.png	(512, 512)	392
5	cameraman.png	(512, 512)	392
6	airplane.png	(512, 512)	392
7	boat.png	(512, 512)	392
8	bridge.png	(512, 512)	392
9	truck.png	(512, 512)	392
10	house.png	(512, 512)	392
11	man.png	(512, 512)	392
12	mountain.png	(512, 512)	392
13	woman.png	(512, 512)	392
14	couple.png	(512, 512)	392
15	splash.png	(512, 512)	392
16	surf.png	(512, 512)	392
17	watch.png	(512, 512)	392
18	kids.png	(512, 512)	392
19	bike.png	(512, 512)	392
20	toys.png	(512, 512)	392

Pada Tabel 3 ditunjukkan hasil preprocessing terhadap dataset citra yang diperoleh secara legal dari USC-SIPI Image Database. Setiap citra pada dataset memiliki resolusi 512×512 piksel dengan format grayscale, sehingga kapasitas penyisipan bit menggunakan metode LSB 1-bit adalah sebesar 262.144 bit untuk setiap gambar. Pesan rahasia yang digunakan dalam penelitian ini memiliki panjang 392 bit dan disisipkan ke seluruh citra yang ada pada dataset. Proses preprocessing dilakukan dengan memeriksa kecukupan kapasitas setiap gambar untuk memastikan pesan dapat disisipkan secara utuh tanpa melebihi kapasitas bit yang tersedia. Hasil menunjukkan bahwa seluruh citra dalam dataset memiliki kapasitas yang memadai untuk proses penyisipan pesan menggunakan metode LSB.

3.2. Penyisipan pesan Least Significant Bit

Teknik menyisipkan pesan yang dilakukan adalah menyisipkan pesan pada bit yang paling tidak significant, seperti bit paling akhir. Tahapan yang dilakukan dalam menyisipkan pesan yaitu membaca gambar dan pesan, kemudian gambar dan pesan dikonversi kedalam pixel dan bit. Selanjutnya bit disisipkan ke *Least Significant Bit* piksel gambar, dan terakhir gambar stego disimpan. Potongan kode dalam menyisipkan pesan dapat dilihat pada gambar 4.

```python
from PIL import Image
import numpy as np

def lsb_embed(image_path, bit_message, output_path):

    img = Image.open(image_path).convert("L")
    img_np = np.array(img)

    flat_pixels = img_np.flatten()

    if len(bit_message) > len(flat_pixels):
        raise ValueError("Pesan terlalu panjang untuk disisipkan ke gambar.")

    stego_pixels = flat_pixels.copy()

    for i in range(len(bit_message)):
        stego_pixels[i] = (stego_pixels[i] & ~1) | int(bit_message[i])


    stego_image = stego_pixels.reshape(img_np.shape)

    stego_img = Image.fromarray(stego_image.astype(np.uint8))
    stego_img.save(output_path)

    return output_path
```

Gambar 4. Potongan kode LSB

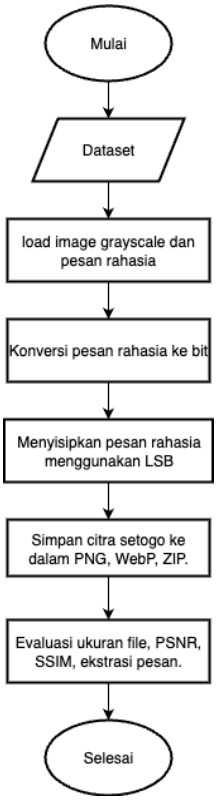
Pada table dapat dilihat gambar dirubah menjadi matriks 1 dimensi untuk dimoddifikasi langsung. Pada potongan code terdapat operasi bitwise untuk menghapus *Least Significant Bit* dari piksel yang kemudian array akan dikembalikan ke bentuk asli gambar, dan gambar hasil penyisipan akan disimpan.

Gambar 5. contoh gambar yang di sisipi pesan

Pada gambar 5 menampilkan dua buah citra yang diantaranya telah disisipi pesan “Ini adalah pesan rahasia untuk diuji pada gambar lena.png” dengan steganografi metode *Least Significant Bit* dan citra yang tidak disisipi pesan. Dapat dilihat bahwa kedua citra tidak terlihat perbedaannya lain hal jika biner dari citra diperlihatkan.

3.3. Kompress Huffman Coding

Teknik kompres Huffman coding yang dilakukan yaitu menghitung frekuensi tiap karakter dalam pasan kemudian akan dibangun pohon Huffman dari frekuensi tiap karakter menjadi karakter kode biner, selanjutnya pesan akan di encode. Dalam Teknik Huffman coding karakter yang sering muncul akan diberikan kode biner lebih pendek dan karakter yang jarang muncul aka diberikan kode biner yang lebih Panjang. Gamabr 6 merupakan alur dari Huffman coding.



Gambar 6. alur Huffman Coding

Pada potongan kode dapat dilihat Teknik kompresi huffman dilakukan dalam fungsi build_huffman_tree, generate_codes, dan huffman_encode. Hasil kompresi adalah bit string yang lebih pendek dari ASCII biasa. Hasil kompresi akan disimpan dalam format PNG, WebP, ZIP.

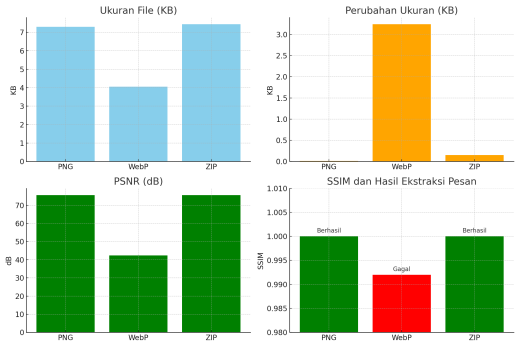
3.4. Evaluasi

Evaluasi yang dilakukan yaitu evaluasi terhadap ukuran file, Evaluasi Degradasi Kualitas Citra, evaluasi kesamaan struktur citra, dan evaluasi ekstrasi pesan. Berikut merupakan evaluasi yang dilakukan pada setiap parameter.

1. Ukuran file
   Pada parameter ukuran file metode yang dilakukan adalah membandingkan ukuran file gamar sebelum dan sesudah penyisipan besar, dalam ukuran byte maupun kilobyte dengan format yang diuji yaitu PNG, WebP dan ZIP.
2. Degradasi Kualitas Citra
   Pada parameter degradasi kualitas citra metode yang dilakukan yaitu menghitung Peak Signal to Noise Ratio antara gambar original dan gambar stego. Ukuran peak signal to noise ratio yang tinggi diatas 40 dB Dimana perbedaan tidak terlihat oleh mata manusia.
3. Kesamaan Struktur Citra
   Pada parameter kesamaan struktur citra metode yang dilakukan menghitung structural similarity index. Nilai structural similarity index yang

mendekati 1.0 artinya gambar sangat mirip dengan aslinya.

4. Ekstrasi Pesan
   Pada parameter ekstrasi pesan metode yang digunakan yaitu mengekstrasi bit dari gambar stego dan merekonstruksi pesan. Hasil dari evaluasi ini yaitu pesan harus indentik dengan pesan asli yang disisipkan.



Gambar 7. hasil evaluasi PSNR, SSIM

Pada Gambar 7 dapat dilihat hasil dari PSNR dan SSIM untuk setiap jenis kompresi. PNG dan ZIP unggul dalam menjaga kualitas citra stego (PSNR tinggi, SSIM 1,0000) serta integritas pesan. WebP meskipun mengurangi ukuran file paling signifikan, namun menurunkan kualitas (PSNR rendah) dan mengakibatkan kegagalan ekstraksi pesan.

3.5. Analisis Komperatif

Analisis komperatif dilakukan dengan melakukan pengujian terkait aspek evaluasi dan format pengujian yang dilakukan. Table 3 merupakan hasil dari pengujian yang dilakukan untuk contoh citra pada gambar 8 dan dengan pesan "Ini adalah pesan rahasia untuk diuji pada gambar lena.png".

Gambar 8. Contoh gambar yang disisipi pesan

Pada tabel 3 dapat dilihat pesan dengan format PNG penambahan bit *Least Significant Bit* meningkatkan ukuran hanya 0.01 KB, tidak ada degradasi visual Dimana nilai PSNR dangat tinggi dan SSIM adalah 1 dan pesan dapat diekstrasi sempurna. Begitupun dengan format ZIP, walaupun ukuran sedikit lebih besar karena overhead ZIP, namun isi file tetap aman dan tidak ada perubahan kualitas atau struktur sehingga pesan yang disisipkan aman. Sedangkan untuk format WebP ukuran file mengecil karena kompresi yang dilakukan lossy, kemudian Bit *Least Significant Bit* berubah sehingga menyebabkan gagal ekstrasi, meskipun visual bagus Dimana SSIM yaitu 0.9 namun fungsi steganografi sudah rusak

Table 3 pengujian LSB dan Huffman Coding

Aspek Evaluasi	PNG	WebP	ZIP
Ukuran File	7.30 KB	4.05 KB	7.44 KB
Perubahan Ukuran	0.01 KB	3.24 KB	0.15 KB
PSNR terhadap Original	75.58 dB	42.38 dB	75.58 dB
SSIM terhadap Original	1.0000	0.9920	1.0000
Ekstraksi Pesan	Berhasil	Gagal	Berhasil

4. Kesimpulan

Metode *Least Significant Bit* dengan Huffman Coding berhasil menyisipkan pesan rahasia kedalam gambar digital. Penggunaan Huffman coding memperpendek Panjang bit pesan yang disisipkan, sehingga mengurangi beban pada media penyimpanan dan meminimalkan gangguan visual pada gambar. Gambar dengan format PNG terbukti paling ideal untuk steganografi *Least Significant Bit*, Dimana format PNG mampu mempertahankan setiap bit informasi tanpa merusak pesan yang disipkan. Evaluasi kualitas menunjukan PSNR tinggi di 75.58 dB dan SSIM sempurna di 1 yang menandakan perubahan pada citra tidak terdeteksi secara visual. Dalam konteks pengiriman pesan rahasia yang disisipkan format ZIP sebagai media mendukung dalam distribusi gambar stego meskipun ada penambahan sedikit ukuran file yaitu 0.15KB. Namun format WebP tidak direkomendasikan untuk penyimpanan gambar stego karena kompresi yang digunakan mengubah struktur bit gambar termasuk bit bit *Least Significant Bit* tempat pesan disisipkan, meskipun ukuran file lebih kecil yaitu 4.05KB sehingga terdapat perubahan ukuran pada 3.24KB dan masih layak secara visual namun pesan yang disisipkan tidak dapat diekstrasi dengan benar.

5. Saran

Penelitian selanjutnya disarankan untuk mengembangkan teknik kompresi pesan yang lebih adaptif seperti Arithmetic Coding atau LZW guna meningkatkan efisiensi ukuran pesan serta mengeksplorasi format citra lossless lain seperti TIFF atau BMP sebagai alternatif PNG. Selain itu perlu dilakukan pengujian dengan kapasitas pesan yang lebih besar untuk mengetahui batas optimal penyisipan tanpa menurunkan kualitas citra, serta mengkaji ketahanan pesan terhadap berbagai transformasi citra seperti rotasi, cropping, dan scaling.

Daftar Pustaka

[1] N. F. Johnson, Z. Duric, and S. Jajodia, Information Hiding: Steganography and Watermarking—Attacks and Countermeasures. Boston, MA, USA: Kluwer Academic Publishers, 2001.

[2] J. C. Kurniawan, A. Nugraha, A. I. Prayogo, and T. F. Novanto, "Improving Data Embedding Capacity in LSB Steganography Utilizing LSB2 and Zlib Compression," Sinkron: Jurnal dan Penelitian Teknik Informatika, vol. 9, no. 1, pp. 174–181, Jan. 2024, doi:10.33395/sinkron.v9i1.13185.

[3] F. Baso, "Performance Analysis of the Last Significant Bit (LSB) Method in Steganography for Data Hiding in Image Data," Scientist — J. Security, Computer, Information, Embedded, Network, and Intelligence System, vol. 1, no. 2, pp. 58–62, Dec. 2023, doi: 10.61220/scientist.v1i2.20234.

[4] Y. Y. Demircan and S. Ozekes, "A Novel LSB Steganography Technique Using Image Segmentation," *Journal of Universal Computer Science (JUCS)*, vol. 30, no. 3, pp. 308–332, Mar. 2024, doi: 10.3897/jucs.105702.

[5] S. A. Author, "Comparative Analysis of Huffman Coding Implementations for Efficient Data Communication Using Greedy and Divide-and-Conquer Techniques," *Enigma: Journal of Information Security*, vol. 2, no. 1, pp. xx–xx, Oct. 2024, doi: 10.62123/enigma.v2i1.24.

[6] D. Kapgate, "A Review on Lossless Image Compression Techniques and Algorithms," International Journal of Computing and Technology (IJCAT), vol. 1, no. 9, pp. 457–460, Oct. 2014.

[7] K. Sayood, Lossless Compression Handbook. San Diego, CA, USA: Academic Press, 2003.

[8] USC-SIPI Image Database, Signal and Image Processing Institute, University of Southern California. [Online]. Available: http://sipi.usc.edu/database/

[9] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital Image Steganography: Survey and Analysis of Current Methods," Signal Processing, vol. 90, no. 3, pp. 727–752, Mar. 2010, doi: 10.1016/j.sigpro.2009.08.010.

[10] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952, doi: 10.1109/JRPROC.1952.273898.

[11] K. Sayood, Introduction to Data Compression, 5th ed. San Francisco, CA, USA: Morgan Kaufmann, 2017.