JIKUM
JURNAL ILMU KOMPUTER

E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

Implementasi Algoritma Least Significant Bit (LSB) untuk Penyembunyian Pesan Teks pada Media Citra Digital dan Audio Berbasis Python

Zulfathan Akbar[1*], Muhammad Nafis [2], Daniel Elhakim [3], Muhammad Ichsan Faqih[4]
[1,2,3,4]Universitas Malikussaleh, Indonesia
[1]zulfathan.23017200@mhs.unimal.ac.id, [2]muhammad.230170185@mhs.unimal.ac.id,
[3]daniel.230170190@mhs.unimal.ac.id, [4]ichsan.230170188@mhs.unimal.ac.id

ABSTRACT

Information security is a crucial aspect of modern digital data exchange. Conventional security techniques, such as cryptography, often raise suspicion from third parties due to the production of unreadable encrypted data. Therefore, steganography techniques are required to conceal the very existence of secret messages within other media. This study aims to design and build a Python-based steganography application capable of embedding text messages into PNG digital images and WAV audio formats. The method employed is the Least Significant Bit (LSB), which works by manipulating the rightmost bit of image pixel data or audio samples. The application is built using the Tkinter library for the interface, alongside Pillow and Wave for media processing. Black Box testing results indicate that the application successfully performs message encoding and decoding processes accurately without compromising data integrity. Perceptually, both visually and auditorily, the resulting steganographic media suffer no significant quality degradation, making it effective in deceiving human senses and maintaining information confidentiality.

Keywords:
Steganography, Least Significant Bit (LSB), Python, Digital Image, WAV Audio.

PENDAHULUAN

Pesatnya perkembangan teknologi informasi di era digital telah mempermudah proses pertukaran data secara global. Namun, kemudahan ini membawa tantangan baru dalam aspek keamanan dan privasi informasi. Kriptografi sering menjadi solusi utama untuk mengamankan pesan, namun pesan yang terenkripsi sering kali memicu kecurigaan pihak ketiga karena bentuk datanya yang acak dan tidak terbaca. Oleh karena itu, diperlukan metode pengamanan data yang tidak menarik perhatian, yaitu steganografi (Husein et al., 2024).

Steganografi adalah seni dan ilmu menyembunyikan informasi ke dalam media pembawa (*cover object*) sedemikian rupa sehingga keberadaan pesan tersebut tidak terdeteksi oleh indra manusia. Berbeda dengan kriptografi yang menyandikan isi pesan, steganografi menyembunyikan eksistensi pesan itu sendiri (Nugraha, 2023). Teknik ini dapat diterapkan pada berbagai media digital, seperti citra (*image*), audio, dan video (Wijaya, 2024).

Salah satu metode steganografi yang paling populer dan efisien adalah *Least Significant Bit* (LSB). Metode ini bekerja dengan memanipulasi bit paling kanan (bit terkecil) dari data piksel citra atau sampel audio untuk disisipi bit pesan rahasia. Keunggulan utama metode LSB adalah kapasitas penyimpanannya yang relatif besar serta perubahan pada media penampung yang sangat minim, sehingga sulit dibedakan secara visual maupun auditori dari file aslinya (Rahman, 2021).

Penelitian sebelumnya telah banyak membahas penerapan LSB pada citra digital (Yanti & Budayawan, 2023), namun integrasi steganografi yang menggabungkan media citra dan audio dalam satu perangkat lunak berbasis Python masih perlu dikembangkan lebih lanjut untuk kemudahan penggunaan. Penggunaan format audio WAV dalam metode LSB juga terbukti memiliki nilai *Signal-to-Noise Ratio* (SNR) yang lebih baik dibandingkan format audio terkompresi (Marevson, 2024; Ozkan, 2022). Selain itu, implementasi menggunakan bahasa pemrograman Python menawarkan fleksibilitas tinggi dengan dukungan pustaka pengolahan citra dan audio yang mumpuni (Sharma, 2025).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan membangun aplikasi steganografi berbasis Python yang mampu menyisipkan pesan teks ke dalam media citra (PNG) dan audio (WAV) menggunakan algoritma LSB. Penelitian ini diharapkan dapat menghasilkan perangkat lunak yang efektif untuk keamanan privasi digital sederhana dengan antarmuka yang mudah digunakan.

TINJAUAN PUSTAKA

Steganografi

Steganografi berasal dari bahasa Yunani *steganos* (tersembunyi) dan *graphein* (tulisan). Konsep dasarnya adalah menyisipkan pesan rahasia (*embedded message*) ke dalam media penampung (*cover object*) untuk menghasilkan media baru (*stego object*) yang tampak normal. Tujuan utamanya adalah untuk menghindari kecurigaan, berbeda dengan kriptografi yang tujuannya adalah menyandikan isi pesan agar tidak bisa dibaca (Husein et al., 2024; Nugraha, 2023). Dalam konteks keamanan *cloud* dan transmisi data modern, teknik ini menjadi lapisan keamanan tambahan yang vital

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

(Zhang, 2024).

Metode Least Significant Bit (LSB)

Algoritma *Least Significant Bit* (LSB) merupakan teknik steganografi spasial yang paling dasar namun efektif. Setiap data digital (baik piksel gambar maupun sampel suara) tersusun atas rangkaian bit biner. LSB memanfaatkan kelemahan sistem persepsi manusia yang tidak peka terhadap perubahan kecil pada data.

Pada citra digital RGB 24-bit, setiap piksel terdiri dari komponen Merah, Hijau, dan Biru yang masing-masing bernilai 8 bit. Metode LSB mengganti bit ke-8 (paling kanan) dari setiap komponen warna dengan bit pesan. Perubahan nilai 1 bit ini hanya mengubah nilai desimal warna sebesar 1 poin (misal dari 255 menjadi 254), yang mana perubahan warna tersebut tidak kasat mata bagi mata manusia (Rahman, 2021; Yanti & Budayawan, 2023).

Hal serupa berlaku pada media audio. Pada format audio WAV (*Waveform Audio File Format*), data suara disimpan dalam bentuk sampel amplitudo tanpa kompresi (*lossless*). Modifikasi pada bit terakhir amplitudo suara tidak akan mengubah kualitas audio secara signifikan sehingga tidak terdengar *noise* atau gangguan yang mencolok di telinga pendengar (Marevson, 2024).

Perbandingan Media Audio WAV dan MP3

Dalam steganografi audio, format file sangat menentukan keberhasilan penyisipan pesan. Penelitian menunjukkan bahwa format WAV lebih unggul untuk metode LSB dibandingkan MP3. Hal ini dikarenakan MP3 menggunakan teknik kompresi *lossy* yang menghilangkan frekuensi suara tertentu, sehingga bit pesan yang disisipkan berisiko rusak atau hilang. Sebaliknya, WAV menyimpan data secara mentah sehingga integritas pesan yang disisipkan pada bit LSB lebih terjaga (Ozkan, 2022; Santoso, 2025).

Implementasi dengan Python

Python adalah bahasa pemrograman tingkat tinggi yang populer untuk pengembangan aplikasi keamanan data. Dukungan pustaka (*library*) seperti Pillow (PIL) untuk manipulasi citra dan modul wave untuk pengolahan audio memudahkan implementasi algoritma steganografi secara efisien (Putra et al., 2024; Sharma, 2025). Penelitian oleh Husein et al. (2024) menunjukkan bahwa implementasi LSB menggunakan Python mampu menghasilkan proses enkripsi dan dekripsi yang cepat dan akurat.

METODE PENELITIAN

Perancangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan pendekatan eksperimental. Sistem dirancang untuk memiliki dua fungsi utama: *Encoding* (penyisipan pesan) dan *Decoding* (pembacaan pesan). Alur logika sistem dimulai dengan input pengguna berupa file (citra/audio) dan pesan teks, kemudian diproses menggunakan algoritma LSB, dan menghasilkan output file stego (Akmal et al., 2023).

Tahapan Implementasi Algoritma

Proses implementasi algoritma LSB dalam penelitian ini dibagi menjadi dua modul utama:

Proses Penyisipan Pesan (Encoding)

1. Konversi Pesan: Pesan teks dikonversi menjadi deretan biner (ASCII 8-bit).
2. Penambahan Delimiter: Menambahkan karakter khusus (misalnya "#####") di akhir pesan sebagai penanda batas akhir data (Putra et al., 2024).
3. Pembacaan Media:
   - Untuk Citra: Membaca nilai piksel RGB menggunakan pustaka Pillow.
   - Untuk Audio: Membaca *frame byte* audio WAV menggunakan pustaka wave (Marevson, 2024).
4. Substitusi Bit: Mengganti bit terakhir (LSB) dari setiap byte media penampung dengan bit pesan secara berurutan.
5. Penyimpanan: Menyimpan hasil modifikasi ke dalam file baru (PNG untuk gambar, WAV untuk audio) guna menghindari kompresi data (Yanti & Budayawan, 2023).

Proses Ekstraksi Pesan (Decoding)

1. Pembacaan Stego Object: Membaca data biner dari file stego yang diunggah.
2. Ekstraksi LSB: Mengambil bit terakhir dari setiap byte data (piksel atau sampel audio).
3. Rekonstruksi Pesan: Menggabungkan bit-bit yang diekstrak menjadi karakter (byte) kembali.
4. Terminasi: Proses pembacaan berhenti ketika sistem mendeteksi karakter *delimiter* yang telah ditentukan sebelumnya (Husein et al., 2024).

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

Alat dan Bahan

Penelitian ini dilakukan menggunakan perangkat keras laptop standar dan perangkat lunak sebagai berikut:

1. Bahasa Pemrograman yang digunakan adalah Python 3.10.4
2. Editor Kode yang digunakan adalah Visual Studio Code.
3. Pustaka Pendukung yang digunakan adalah Tkinter (untuk antarmuka GUI), Pillow (pengolahan citra), wave (pengolahan audio), dan os (manajemen file) (Sharma, 2025).
4. Data Uji menggunakan sampel citra format .PNG dan sampel audio format .WAV dengan variasi ukuran untuk pengujian kapasitas.

Metode Pengujian

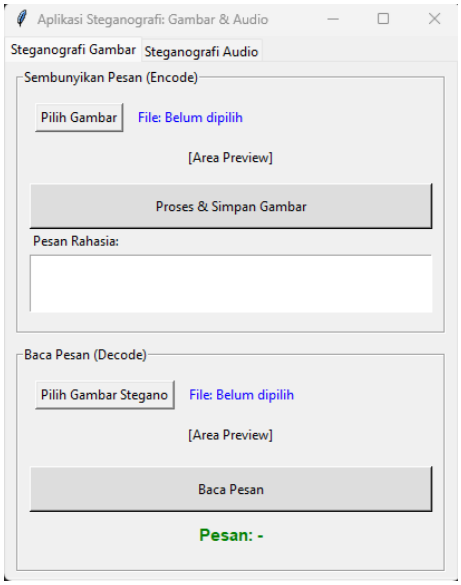
Untuk memastikan perangkat lunak berjalan sesuai spesifikasi dan menghasilkan media steganografi yang berkualitas, penelitian ini menerapkan dua tahapan pengujian:

1. Pengujian Fungsional (*Black Box Testing*): Pengujian ini berfokus pada fungsionalitas fitur tanpa melihat struktur kode internal. Skenario pengujian meliputi validasi input format file (harus PNG/WAV), proses penyisipan pesan dengan variasi panjang karakter, dan akurasi hasil dekripsi pesan. Keberhasilan ditandai jika pesan yang diekstraksi sama persis dengan pesan asli (Setiawan & Wardana, 2021).
2. Pengujian Kualitas (*Perceptual & Statistical Analysis*):
   - Analisis Persepsi: Melibatkan pengamatan visual (pada citra) dan pendengaran (pada audio) untuk memastikan tidak ada *noise* atau distorsi yang mencolok (Pradana, 2023).
   - Analisis Statistik (Opsional): Mengukur nilai *Peak Signal-to-Noise Ratio* (PSNR) dan *Mean Square Error* (MSE) untuk menghitung tingkat kemiripan antara media asli dan media steganografi secara matematis (Rahman, 2021).

HASIL DAN PEMBAHASAN

Implementasi Antarmuka Pengguna

Aplikasi steganografi ini dikembangkan menggunakan bahasa pemrograman Python dengan pustaka Tkinter sebagai antarmuka pengguna (*Graphical User Interface*). Antarmuka dirancang sederhana dengan model *tabbed-view* untuk memisahkan fungsi steganografi gambar dan audio, sehingga memudahkan interaksi pengguna. Sharma (2025) menyatakan bahwa penggunaan pustaka standar seperti Tkinter sangat efektif untuk pengembangan purwarupa aplikasi keamanan yang ringan dan portabel.



Gambar 1. Tampilan Antarmuka Utama Aplikasi Steganografi

Pada Gambar 1, terlihat aplikasi memiliki dua panel utama: panel atas untuk proses penyisipan pesan (Encode) yang dilengkapi dengan fitur pratinjau (*preview*) citra, dan panel bawah untuk proses pembacaan pesan (*Decode*).

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

Pengujian Steganografi pada Citra Digital

Pengujian pertama dilakukan pada media citra digital berformat .PNG. Format PNG dipilih karena sifat kompresinya yang *lossless*, sehingga data piksel tidak berubah saat disimpan. Hal ini krusial karena menurut Yanti dan Budayawan (2023), integritas data piksel mutlak diperlukan agar bit pesan yang disisipkan menggunakan metode LSB tidak rusak.

Proses Penyisipan Pesan (Encoding)

Pengguna memilih file citra asli, kemudian memasukkan pesan rahasia pada kolom teks yang tersedia. Pada pengujian ini, pesan yang disisipkan adalah *"Ini adalah data rahasia perusahaan"*. Sistem kemudian memproses citra tersebut dengan mengubah bit terakhir dari setiap komponen warna piksel.

Gambar 2. Proses Input Citra dan Pesan Rahasia

Setelah tombol "Proses & Simpan" ditekan, aplikasi menghasilkan file baru. Secara visual, tidak terdapat perbedaan yang dapat dilihat oleh mata manusia antara citra asli (*cover image*) dengan citra hasil steganografi (*stego image*). Hal ini membuktikan teori Rahman (2021) bahwa perubahan 1 bit LSB tidak mempengaruhi persepsi visual manusia.

Gambar 3. Perbandingan Visual Citra Asli (Kiri) dan Citra Steganografi (Kanan)

Seperti yang diilustrasikan pada Gambar 3, implementasi algoritma LSB pada aplikasi ini menghasilkan citra *stego* yang memiliki fidelitas tinggi terhadap citra aslinya. Meskipun data biner dari pesan teks telah disisipkan ke dalam struktur piksel gambar sebelah kanan, tidak terdapat perbedaan visual yang signifikan dibandingkan gambar referensi di sebelah kiri. Kualitas persepsi (*imperceptibility*) yang tinggi ini membuktikan efektivitas metode LSB dalam menyembunyikan informasi tanpa memancing kecurigaan pihak ketiga.

Proses Ekstraksi Pesan (Decoding)

Pengujian dekripsi dilakukan dengan mengunggah kembali file *stego image* ke dalam aplikasi pada panel *Decode*. Sistem membaca bit LSB dari citra dan menyusunnya kembali menjadi karakter. Hasil pengujian menunjukkan pesan berhasil diekstraksi secara utuh tanpa kesalahan karakter.

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

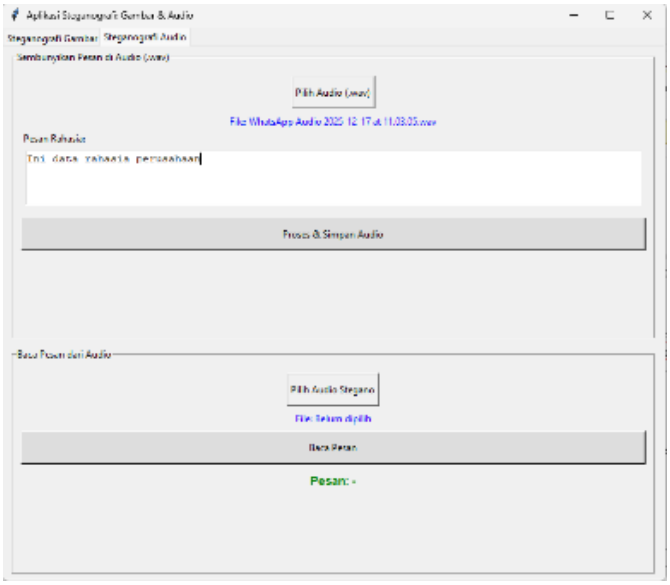
Gambar 4. Hasil Ekstraksi Pesan dari Citra Steganografi

Pengujian Steganografi pada Audio

Pengujian kedua dilakukan pada file audio berformat .WAV. Berbeda dengan MP3 yang membuang frekuensi suara tertentu, WAV menyimpan data audio secara utuh sehingga sangat cocok untuk metode LSB (Siregar & Panggabean, 2024).

Proses Encode dan Decode Audio

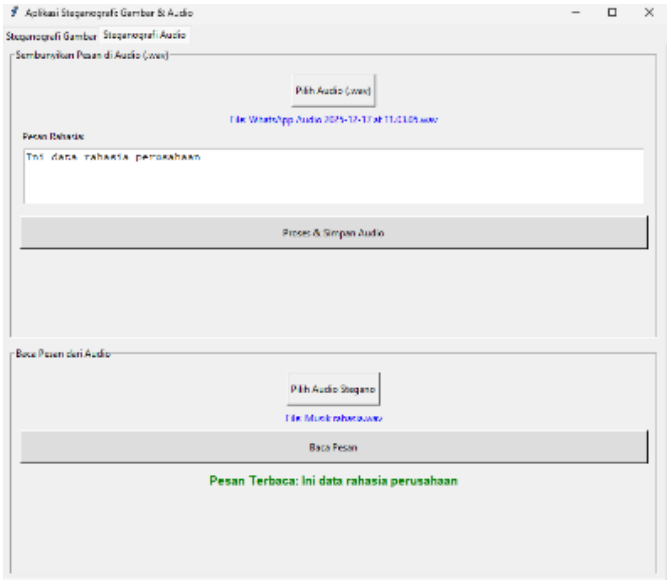
Pada tahap ini, pesan rahasia disisipkan ke dalam sampel amplitudo audio. Setelah proses penyimpanan, file audio hasil steganografi diputar menggunakan pemutar media standar. Hasil pendengaran subjektif menunjukkan tidak ada gangguan suara (*noise*) atau *glitch* yang terdengar; kualitas audio terdengar sama persis dengan aslinya. Hal ini sejalan dengan temuan Marevson (2024) yang menyatakan bahwa modifikasi amplitudo pada level bit terendah tidak merusak kualitas audio.



Gambar 5. Antarmuka Steganografi Audio

Selanjutnya, proses pembacaan pesan dilakukan untuk memvalidasi integritas data. Aplikasi berhasil mendeteksi *delimiter* akhir pesan dan menampilkan teks tersembunyi dengan akurat.

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026



Gambar 6. Keberhasilan Dekripsi Pesan pada File Audio

Analisis Kinerja

Berdasarkan hasil pengujian di atas, metode LSB yang diimplementasikan dengan Python terbukti efektif. Penggunaan pustaka Pillow dan wave memungkinkan manipulasi data hingga tingkat bit dengan efisiensi waktu yang tinggi (Putra et al., 2024).

Kelebihan utama dari sistem ini adalah ketahanan persepsi (*imperceptibility*). Modifikasi bit terkecil (LSB) hanya mengubah nilai warna atau amplitudo sebesar 1 unit, perubahan yang sangat kecil sehingga tidak terdeteksi oleh indra manusia. Hal ini sejalan dengan penelitian Akmal et al. (2023) yang menyatakan bahwa metode LSB sangat unggul dalam menyembunyikan keberadaan pesan tanpa merusak kualitas media penampung.

Namun, sistem ini memiliki keterbatasan pada jenis file. File hasil steganografi tidak boleh dikonversi ke format *lossy* (seperti JPG atau MP3) atau dikompresi (misalnya dikirim via WhatsApp), karena proses tersebut akan menghancurkan susunan bit LSB yang berisi pesan rahasia (Ozkan, 2022; Santoso, 2025).

KESIMPULAN

Berdasarkan perancangan, implementasi, dan pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Aplikasi steganografi berbasis Python berhasil dibangun dan mampu mengimplementasikan algoritma *Least Significant Bit* (LSB) untuk menyisipkan pesan teks ke dalam media citra digital (.PNG) dan audio (.WAV).
2. Pengujian *Black Box* menunjukkan bahwa fitur *encoding* dan *decoding* berjalan 100% sesuai fungsi. Pesan yang disisipkan dapat diekstraksi kembali secara utuh tanpa pengurangan atau perubahan karakter.
3. Secara kualitas persepsi, media hasil steganografi tidak mengalami penurunan kualitas yang signifikan. Gambar tidak terlihat mengalami distorsi warna, dan audio tidak terdengar memiliki *noise*, sehingga memenuhi prinsip utama steganografi yaitu menyembunyikan keberadaan pesan dari kecurigaan pihak ketiga.

REFERENSI

Akmal, R. A., Setiawan, D., & Rahmad, I. F. (2023). Implementasi metode Least Significant Bit dalam teknik steganografi pada berkas audio dengan stego citra digital. *G-Tech: Jurnal Teknologi Terapan*, *7*(1), 1-10.

Husein, A., Ramadhani, F., & Saputra, A. (2024). Steganografi: Keamanan data dengan metode Least Significant Bit menggunakan Python. *Arus Jurnal Sosial dan Humaniora*, *4*(2), 120-130.

Marevson, R. (2024). Implementasi audio steganografi dalam penyembunyian pesan rahasia. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, *3*(2), 45-52.

Nugraha, A. (2023). *Implementasi fragile watermarking dan steganografi Least Significant Bit pada file citra*. Makalah IF4020 Kriptografi, Program Studi Informatika ITB.

Ozkan, M. (2022). Comparative analysis of audio steganography methods. *International Journal of Information Security Science*, *11*(3), 88-97.

Pradana, A. (2023). Penggunaan fungsi Hitzl-Zele pada implementasi gabungan Secret Sharing Shamir's (t, w)-Threshold Scheme dan steganografi audio Least Significant Bit. *Jurnal UPI (JEM)*, *3*(1), 15-22.

JIKUM
JURNAL ILMU KOMPUTER

E-ISSN: 3090-739X
Volume 2, **Number** 2, November 2026

Putra, D. S., Wijaya, K., & Santoso, H. (2024). Analisis proses hasil enkripsi steganografi file citra gambar dengan metode Least Significant Bit menggunakan Python. *Jurnal Pendidikan Inovatif (Journal Versa)*, *5*(1), 22-30.

Rahman, S. (2021). A comparative analysis of LSB, MSB and PVD based image steganography. *International Journal of Research and Review*, *8*(5), 112-119.

Santoso, B. (2025). Analisis kualitas audio steganografi MP3 menggunakan teknik masking pada spectrogram. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi (Jurnal SNATI)*, *4*(1), 15-24.

Setiawan, R., & Wardana, L. (2021). Implementasi steganografi image processing dan enkripsi AES menggunakan OpenStego. *Akrab Juara: Jurnal Ilmu-ilmu Sosial*, *6*(2), 45-55.

Sharma, R. (2025). Steganography using Python: A practical approach. *International Journal of Research Publication and Reviews*, *6*(1), 200-205.

Siregar, B., & Panggabean, J. (2024). Implementasi metode LSB dan Playfair untuk mengamankan file teks ke dalam file audio WAV. *Jurnal Teknik Informatika (JUTIF)*, *5*(3), 500-510.

Wijaya, T. (2024). Penerapan steganografi file teks pada video menggunakan metode Least Bit Significant (LSB). *Jurnal Sistemasi: Jurnal Sistem Informasi*, *13*(2), 300-310.

Yanti, F., & Budayawan, K. (2023). Implementasi steganografi menggunakan metode Least Significant Bit (LSB) dalam pengamanan informasi pada citra digital. *Jurnal Vocational Teknik Elektronika dan Informatika*, *11*(1), 50-58.

Zhang, L. (2024). A novel LSB steganography technique for enhancing cloud security. *Journal of Information Security*, *15*(2), 150-165.