Implementasi Steganografi dengan Medium Aset Minecraft

Raja Muhammad Arkan H M A - 10123065
Program Studi Matematika
Fakultas Matematika & Ilmu Pengetahuan Alam
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: rajamahma2005@gmail.com , 10123065@mahasiswa.itb.ac.id

Abstrak*—Di beberapa kesempatan, steganografi diterapkan pada gambar-gambar yang mengandung unsur realisme, seperti pemandangan, hewan, atau tumbuhan. Akibatnya, penyisipan pesan pada gambar jenis tersebut lebih mampu menimbulkan kecurigaan dibandingkan gambar aset Minecraft, sebab orang-orang secara psikologis akan cenderung melihatnya hanya sebagai gambar biasa tanpa ada pikiran bahwa terdapat pesan yang disembunyikan pada gambar tersebut. Makalah ini mengusulkan implementasi steganografi Least Significant Bit (LSB) dengan medium berupa aset citra Minecraft, khususnya skin pemain berformat PNG. Kontribusi utama makalah ini adalah skema penyisipan multi-citra (*multi-image steganography*), di mana satu pesan rahasia disisipkan secara terdistribusi ke dalam beberapa citra aset Minecraft sekaligus apabila kapasitas satu citra tidak mencukupi panjang bitstream pesan. Lalu, steganografi dengan multi-citra tentunya akan lebih sulit untuk dipecahkan dibandingkan single-citra, sebab pihak yang tidak berwenang harus mengetahui seluruh kumpulan citra beserta urutannya untuk dapat merekonstruksi pesan secara utuh. Sistem melakukan penelusuran direktori secara rekursif (*depth-first search*) untuk mengumpulkan seluruh citra PNG yang tersedia, menghitung kapasitas penyisipan tiap citra berdasarkan jumlah piksel dan kanal warna, kemudian mendistribusikan bit pesan secara berurutan ke citra-citra tersebut hingga seluruh pesan tersisip. Proses ekstraksi dilakukan dengan membaca kembali bit LSB dari kumpulan citra stego sesuai urutan yang sama untuk merekonstruksi pesan asli. Pengujian dilakukan terhadap sejumlah skin Minecraft dengan variasi panjang pesan untuk mengevaluasi keberhasilan proses penyisipan dan ekstraksi. Hasil pengujian menunjukkan bahwa skema yang diusulkan mampu menyisipkan dan mengekstraksi pesan rahasia secara akurat pada kumpulan citra aset Minecraft.

Kata Kunci: steganografi, Least Significant Bit, multi-image steganography, citra digital

I. PENDAHULUAN

Di beberapa kesempatan, steganografi diterapkan pada gambar-gambar yang mengandung unsur realisme, seperti pemandangan, hewan, atau tumbuhan. Akibatnya, penyisipan pesan pada gambar jenis tersebut lebih mampu menimbulkan kecurigaan dibandingkan gambar aset Minecraft. Orang-orang secara psikologis akan cenderung melihatnya hanya sebagai gambar biasa tanpa ada pikiran terdapat pesan yang disembunyikan pada gambar tersebut. Lalu steganografi dengan multi-image tentunya akan lebih sulit untuk dipecahkan dibandingkan single-image.

Hal ini didasari oleh fakta bahwa aset Minecraft, khususnya skin pemain, memiliki karakteristik visual yang khas berupa citra berukuran kecil, didominasi oleh blok-blok warna datar, dan tersebar dalam jumlah yang sangat banyak di berbagai situs penyedia skin secara publik. Karakteristik ini justru menguntungkan dari sisi steganografi: gambar semacam ini umumnya tidak dianggap sebagai objek yang "berharga" untuk dianalisis secara forensik, berbeda dengan foto pribadi atau gambar yang mengandung informasi sensitif secara visual. Sementara itu, mayoritas penelitian dan praktik steganografi citra yang ada selama ini berfokus pada medium berupa fotografi dunia nyata, sehingga ruang aset digital seperti game relatif belum banyak dieksplorasi sebagai medium penyisipan pesan.

Selain dari sisi pemilihan medium, pendekatan single-image steganography juga memiliki keterbatasan dari segi kapasitas dan keamanan. Kapasitas penyisipan pesan pada satu citra dibatasi oleh jumlah piksel dan kanal warna yang dimilikinya, sehingga pesan yang panjang dapat melebihi kapasitas citra tunggal yang tersedia. Lebih lanjut, apabila satu citra stego berhasil diidentifikasi dan dianalisis oleh pihak yang tidak berwenang, seluruh pesan rahasia berpotensi terbongkar sekaligus. Pendekatan multi-image steganography menjadi alternatif yang relevan untuk mengatasi kedua permasalahan tersebut, sebab pesan didistribusikan ke dalam beberapa citra sekaligus berdasarkan urutan tertentu, sehingga pihak yang tidak berwenang harus berhasil mengidentifikasi seluruh kumpulan citra beserta urutan penyusunannya untuk dapat merekonstruksi pesan secara utuh.

Berdasarkan kedua pertimbangan tersebut, makalah ini mengusulkan implementasi steganografi LSB dengan skema penyisipan multi-citra menggunakan aset Minecraft, khususnya skin pemain berformat PNG, sebagai medium penyisipan. Sistem yang dibangun melakukan penelusuran direktori secara rekursif untuk mengumpulkan seluruh citra yang tersedia, menghitung kapasitas penyisipan tiap citra, kemudian mendistribusikan bitstream pesan secara berurutan

ke dalam kumpulan citra tersebut hingga seluruh pesan tersisip. Kontribusi makalah ini terletak pada kombinasi antara pemilihan medium yang belum banyak dieksplorasi dengan skema penyisipan multi-citra yang meningkatkan baik kapasitas maupun tingkat keamanan dibandingkan pendekatan single-image konvensional.

II. STUDI PUSTAKA

A. Steganografi

Steganografi merupakan teknik penyembunyian pesan rahasia ke dalam suatu medium sedemikian rupa sehingga keberadaan pesan tersebut tidak disadari oleh pihak yang tidak berwenang. Berbeda dengan kriptografi yang menyamarkan isi pesan, steganografi menyembunyikan keberadaan pesan itu sendiri. Salah satu medium yang umum digunakan dalam steganografi adalah citra digital, sebab citra memiliki redundansi data yang dapat dimanfaatkan untuk menyisipkan informasi tambahan tanpa mengubah tampilan visual secara signifikan.

B. Metode Least Significant Bit (LSB)

Metode LSB merupakan salah satu teknik steganografi citra yang paling umum digunakan karena kesederhanaan implementasinya. Setiap piksel pada citra digital RGB terdiri atas tiga kanal warna, yaitu merah (R), hijau (G), dan biru (B), yang masing-masing direpresentasikan dalam 8 bit. Bit paling tidak signifikan (least significant bit) dari setiap kanal warna dapat diganti dengan satu bit pesan rahasia tanpa menimbulkan perubahan visual yang dapat dipersepsikan oleh mata manusia, sebab perubahan nilai yang ditimbulkan hanya berkisar ±1 dari nilai aslinya.

Secara matematis, citra digital direpresentasikan sebagai kumpulan matriks di mana setiap elemennya menyimpan intensitas piksel. Pada model warna RGB dengan kedalaman 24-bit, setiap piksel dialokasikan 8-bit untuk masing-masing kanal warna, yang direpresentasikan dalam nilai desimal 0 hingga 255. Modifikasi pada posisi bit ke-0 (Least Significant Bit) pada dasarnya beroperasi pada eksponen terkecil dalam representasi biner. Oleh karena itu, pengubahan bit dari 0 menjadi 1 atau sebaliknya hanya akan menyebabkan deviasi nilai intensitas warna maksimum sebesar 1 tingkat. Deviasi mikroskopis ini berada jauh di bawah ambang resolusi kontras sistem visual manusia (*Human Visual System*), sehingga integritas statistik dan visual dari citra medium (*cover image*) tetap terjaga secara utuh. Prinsip fundamental inilah yang menjadikan metode LSB sangat esensial, karena ia mampu memberikan rasio yang optimal antara kapasitas muatan data tersembunyi (*payload capacity*) dan tingkat ketidaktampakan visual (*imperceptibility*)

C. Multi-Image Steganography

Pada skema steganografi konvensional, pesan rahasia disisipkan ke dalam satu citra tunggal (*single-image steganography*). Pendekatan ini memiliki keterbatasan dari segi kapasitas, sebab panjang pesan yang dapat disisipkan dibatasi oleh jumlah piksel dan kanal warna yang dimiliki citra tersebut. Multi-image steganography merupakan pengembangan dari pendekatan tersebut, di mana satu pesan rahasia didistribusikan ke dalam beberapa citra sekaligus. Pendekatan ini tidak hanya meningkatkan kapasitas total penyisipan, tetapi juga meningkatkan tingkat keamanan, sebab pihak yang tidak berwenang harus berhasil mengidentifikasi seluruh kumpulan citra beserta urutan penyusunannya untuk dapat merekonstruksi pesan secara utuh.

III. METODOLOGI

Skema yang diusulkan pada makalah ini terdiri atas dua proses utama, yaitu proses penyisipan (*embedding*) dan proses ekstraksi (*extraction*), dengan medium berupa citra aset Minecraft berformat PNG, khususnya skin pemain.

A. Proses Penyisipan

Proses penyisipan terdiri atas beberapa tahap berurutan sebagai berikut.

1) Penelusuran Direktori (Folder Traversal). Sistem melakukan penelusuran direktori secara rekursif menggunakan algoritma *depth-first search* (DFS) untuk mengumpulkan seluruh berkas citra berformat PNG yang tersedia di dalam suatu direktori beserta seluruh subdirektorinya. Setiap citra PNG yang ditemukan dicatat *path*-nya, dan jumlah total citra yang ditemukan turut dilaporkan kepada pengguna.

2) Pengurutan Citra. Kumpulan *path* citra yang telah ditemukan kemudian diurutkan secara konsisten berdasarkan nama berkas, untuk memastikan urutan penyisipan dan ekstraksi pesan dapat direproduksi secara deterministik pada tahap ekstraksi.

3) Konversi Pesan menjadi Bitstream. Pesan rahasia berupa teks dikonversi menjadi rangkaian bit (*bitstream*) dengan menambahkan header berukuran 32 bit pada bagian awal yang menyatakan panjang pesan dalam satuan byte. Header ini diperlukan pada tahap ekstraksi untuk menentukan secara presisi jumlah bit pesan yang perlu dibaca, sehingga proses ekstraksi tidak membaca bit residu dari kapasitas citra yang tidak digunakan.

4) Pemeriksaan Kapasitas. Sebelum penyisipan dilakukan, sistem menghitung kapasitas total penyisipan dari seluruh citra yang tersedia, yaitu jumlah piksel dikalikan tiga kanal warna untuk setiap citra, kemudian dijumlahkan ke seluruh citra. Apabila panjang bitstream pesan melebihi kapasitas total yang tersedia, proses penyisipan dibatalkan dan sistem mengembalikan pesan kesalahan.

5) Penyisipan Bit Secara Berurutan ke Multi-Citra. Bit pesan disisipkan ke kanal LSB tiap piksel pada citra pertama dalam urutan yang telah ditentukan. Apabila kapasitas citra pertama telah terpakai seluruhnya namun bitstream pesan belum habis tersisip, proses penyisipan dilanjutkan ke citra berikutnya dalam urutan yang sama, dan begitu seterusnya hingga seluruh bit pesan berhasil tersisip atau seluruh citra telah digunakan. Dengan demikian, satu pesan rahasia dapat

tersebar ke dalam beberapa citra stego sekaligus, bergantung pada panjang pesan dan kapasitas tiap citra.

B. Proses Ekstraksi

Proses ekstraksi dilakukan dengan urutan yang berkebalikan dari proses penyisipan. Sistem membaca kumpulan citra stego sesuai urutan yang sama dengan urutan penyisipan, kemudian mengekstraksi bit LSB dari setiap kanal warna tiap piksel secara berurutan menggunakan fungsi generator. Sebanyak 32 bit pertama yang berhasil diekstraksi diinterpretasikan sebagai header panjang pesan, yang kemudian digunakan untuk menentukan jumlah bit pesan aktual yang perlu dibaca selanjutnya. Bit-bit pesan yang telah terkumpul kemudian dikelompokkan per 8 bit dan dikonversi kembali menjadi rangkaian byte, lalu didekode menjadi teks pesan asli.

Sistem turut menyediakan mekanisme validasi untuk menentukan apakah suatu kumpulan citra merupakan citra stego atau bukan, dengan menangkap kemungkinan kegagalan yang dapat terjadi selama proses ekstraksi, yaitu kapasitas bit yang tidak mencukupi panjang header, hasil ekstraksi bit yang tidak membentuk rangkaian byte UTF-8 yang valid, dan kegagalan lain pada saat penguraian (*parsing*) bitstream.

C. Implementasi Algoritma

Implementasi manipulasi *Least Significant Bit* (LSB) dieksekusi secara *bitwise* pada setiap piksel. Berikut adalah cuplikan kode inti yang menangani proses penyisipan *bitstream* ke dalam kanal warna RGB secara iteratif melintasi berbagai citra:

```
def embed_bitstream_to_images(sorted_paths: list[str], bitstream: str) -> None:
    check_total_capacity(sorted_paths, len(bitstream))

    bit_index = 0
    total_bits = len(bitstream)

    for path in sorted_paths:
        if bit_index >= total_bits:
            break  # bitstream udah abis, sisa file gak usah disentuh

        with Image.open(path) as img:
            img = img.convert("RGB")
            pixels = img.load()
            width, height = img.size

            for x, y in itertools.product(range(width), range(height)):
                if bit_index >= total_bits:
                    break
                r, g, b = pixels[x, y]
                channels = [r, g, b]

                for c in range(3):
                    if bit_index >= total_bits:
                        break
                    bit = int(bitstream[bit_index])
                    channels[c] = (channels[c] & 0b11111110) | bit # ganti LSB doang
                    bit_index += 1

                pixels[x, y] = tuple(channels)

            img.save(path)  # overwrite PNG asli jadi stego-image

            print(f"[EMBED] {path} -> selesai (bit_index sekarang: {bit_index})")

    print(f"[EMBED] Total bit tersisip: {bit_index}/{total_bits}")
```

Sedangkan untuk proses rekonstruksi pesan, sistem membaca kembali bit terakhir menggunakan operasi *bitwise* AND (& 1) yang diimplementasikan dalam bentuk *generator* untuk menghindari pembacaan memori yang berlebihan: *(Masukkan fungsi extract_bits_from_images dan extract_message_from_images di sini)*

```
def extract_bits_from_images(sorted_paths: list[str]):
    for path in sorted_paths:
        with Image.open(path) as img:
            img = img.convert("RGB")
            pixels = img.load()
            width, height = img.size

            for x, y in itertools.product(range(width), range(height)):
                r, g, b = pixels[x, y]
                for value in (r, g, b):
```

```
                yield value & 1


    def extract_message_from_images(sorted_paths: list[str]) -
> str:
        bit_gen = extract_bits_from_images(sorted_paths)


          header_bits = "".join(str(next(bit_gen)) for _ in
range(32))
        message_length_bytes = int(header_bits, 2)


        message_bit_count = message_length_bytes * 8
          message_bits = "".join(str(next(bit_gen)) for _ in
range(message_bit_count))


        message_bytes = bytes(
              int(message_bits[i:i + 8], 2) for i in range(0,
len(message_bits), 8)
        )
        return message_bytes.decode("utf-8")
```

IV. SKEMA PENGUJIAN

Pengujian terhadap sistem steganografi *multi-image* ini berfokus pada validasi fungsionalitas penyisipan dan ekstraksi, pengujian batas kapasitas, ketahanan penanganan galat (*error handling*), serta analisis visual medium aset Minecraft.

A. Lingkungan dan Skenario Pengujian

Skenario pengujian menggunakan kumpulan citra aset *skin* Minecraft berformat PNG yang ditempatkan dalam suatu struktur direktori. Sistem diuji menggunakan pesan *plaintext* dengan variasi panjang karakter untuk menyimulasikan dua kondisi utama: pesan yang dapat ditampung oleh satu citra (*single-image*) dan pesan panjang yang membutuhkan distribusi ke beberapa citra (*multi-image*).

B. Pengujian Fungsionalitas DFS dan Kapasitas

Pengujian ini bertujuan memvalidasi modul penelusuran (*folder traversal*) dan perhitungan kapasitas.

1. Validasi Depth-First Search (DFS*):* Sistem diuji kemampuannya menelusuri direktori dan subdirektori untuk menemukan seluruh berkas .png, kemudian mengurutkan *path* secara alfabetis absolut agar urutan penyisipan dan ekstraksi deterministik.
2. Pemeriksaan Kapasitas (Pre-flight Check*):* Sistem diuji dengan ukuran *bitstream* pesan yang sengaja dibuat melebihi akumulasi kapasitas *Least Significant Bit* (LSB) seluruh citra PNG yang ditemukan. Sistem yang berjalan dengan benar harus membatalkan proses dan mengeluarkan pesan kesalahan (*ValueError*) sebelum manipulasi piksel terjadi.

C. Pengujian Penyisipan dan Ekstraksi Pesan

Tahap ini memvalidasi inti algoritma steganografi.

1. Keberhasilan Penyisipan LSB*:* Pengujian dilakukan dengan memverifikasi bahwa *bitstream* pesan, yang diawali dengan *header* ukuran 32-bit, berhasil menimpa nilai bit terakhir (melalui operasi *bitwise* & 0b11111110 | bit) pada kanal R, G, dan B secara berurutan menembus batas file citra jika diperlukan.
2. Akurasi Ekstraksi*:* Sistem diuji kemampuannya membaca kembali 32-bit pertama sebagai penentu panjang pesan, lalu mengekstrak sisa bit secara iteratif dari kumpulan citra stego. Validasi berhasil apabila *bitstream* yang diekstrak membentuk rangkaian *byte* UTF-8 yang sama persis dengan pesan asli.

D. Pengujian Penanganan Galat (Error Handling)

Karena sistem dirancang untuk memproses direktori secara buta, perlu diuji bagaimana sistem merespons berkas PNG yang tidak dimodifikasi (bukan citra stego). Pengujian dilakukan dengan memberikan citra *skin* Minecraft normal ke fungsi ekstraktor. Sistem dievaluasi berdasarkan kemampuannya menangkap eksepsi dengan tepat, yaitu:

1. Menangkap galat StopIteration (melalui fungsi *helper* khusus agar tidak bereskalasi menjadi RuntimeError bawaan PEP 479) apabila kapasitas bit habis sebelum *header* atau pesan selesai dibaca.
2. Menangkap galat UnicodeDecodeError apabila bit LSB acak yang terekstrak tidak membentuk format teks UTF-8 yang valid, membuktikan bahwa citra tersebut bersih dari pesan rahasia.

E. *Analisis Visual Citra*

Modifikasi LSB secara teoretis hanya mengubah nilai kanal warna maksimal sebesar 1 tingkat. Pengujian visual dilakukan dengan membandingkan *skin* Minecraft asli dan citra stego secara berdampingan. Mengingat karakteristik aset Minecraft yang didominasi blok warna datar, evaluasi ini krusial untuk memastikan tidak adanya *noise* atau degradasi warna yang kasat mata, sehingga aspek *imperceptibility* dari steganografi tetap terjaga.

V. SKEMA PENGUJIAN

Pengujian skema steganografi *multi-image* dilakukan dengan mengeksekusi program berbasis Python yang telah dirancang. Pembahasan hasil pengujian dibagi ke dalam beberapa aspek sesuai dengan skema yang telah ditetapkan.

A. Hasil Penelusuran Direktori dan Kapasitas

Berdasarkan pengujian, fungsi *Depth-First Search* (DFS) terbukti secara konsisten menemukan seluruh berkas citra berformat PNG di dalam struktur direktori. Pengurutan *path*

secara absolut alfabetis berjalan sesuai ekspektasi, memastikan urutan berkas tidak berubah antara fase penyisipan dan ekstraksi. Sebagai contoh representatif, sebuah *skin* Minecraft standar beresolusi 64 x 64 piksel terbukti menyediakan kapasitas penyisipan teoretis sebesar 12.288 bit per kanal, atau 36.864 bit per citra (menggunakan kanal RGB). Fitur *pre-flight check* berhasil mengakumulasi total kapasitas seluruh berkas dan secara akurat mengizinkan atau menolak eksekusi berdasarkan panjang *bitstream* pesan.

B. Hasil Penyisipan (Embedding)

Proses penyisipan terbukti mampu mendistribusikan *bitstream* pesan melintasi batas citra tunggal. Modifikasi *Least Significant Bit* (LSB) dieksekusi menggunakan operasi *bitwise*, menyisipkan 32-bit *header* penanda panjang *byte* yang diikuti oleh *bitstream* karakter UTF-8. Sistem berhasil berhenti secara presisi ketika iterasi *bitstream* habis, membiarkan sisa piksel dan sisa berkas citra lainnya dalam kondisi orisinal.

C. Hasil Ekstraksi dan Penanganan Galat

Sistem ekstraksi yang menelusuri ulang kumpulan citra berhasil mengumpulkan kembali bit LSB. Penguraian 32-bit pertama secara akurat memberikan informasi batas panjang pesan, mencegah sistem membaca *noise* atau ruang kosong yang tidak perlu. Ekstraksi menghasilkan teks *plaintext* yang 100% identik dengan pesan asli tanpa adanya kerusakan karakter (*data loss*).

Dalam pengujian ketahanan (*error handling*), pemberian citra *skin* orisinal (bukan citra stego) ke dalam fungsi ekstraktor berhasil memicu penanganan eksepsi secara tepat. Perbaikan arsitektur kode dengan memisahkan fungsi generator secara mandiri berhasil mencegah galat terselubung akibat PEP 479 (RuntimeError), sehingga program secara spesifik mampu menangkap galat ketiadaan kapasitas (StopIteration) atau kegagalan dekode string (UnicodeDecodeError) dengan mulus, membuktikan bahwa citra tersebut bersih dari sisipan pesan.

D. Analisis Visual

Aset *skin* Minecraft memiliki karakteristik unik, yakni beresolusi rendah dan didominasi oleh perpaduan blok-blok warna (*flat colors* / *pixelated*). Hasil observasi visual menunjukkan bahwa modifikasi pada LSB secara praktis tidak memicu degradasi visual yang dapat dipersepsikan oleh mata manusia. Citra stego tampak sama persis dengan aset *skin* orisinal, memastikan syarat utama steganografi, yakni *imperceptibility* (ketidaktampakan keberadaan pesan), berhasil dipenuhi dengan sangat baik.

VI. Kesimpulan

Makalah ini telah berhasil memaparkan rancangan dan implementasi steganografi LSB dengan pendekatan *multi-image* menggunakan medium citra aset permainan Minecraft. Berdasarkan hasil pengujian, dapat disimpulkan bahwa:

1. Skema penyisipan lintas citra (*multi-image steganography*) berhasil menyelesaikan permasalahan keterbatasan kapasitas pada citra tunggal dengan cara mendistribusikan *bitstream* pesan ke dalam urutan direktori secara deterministik melalui metode *Depth-First Search*.
2. Pemilihan aset *skin* Minecraft sebagai medium penyisipan terbukti sangat efektif. Karakteristik visualnya yang unik membuat citra stego terlihat natural, meminimalkan kecurigaan analisis forensik visual dibandingkan penggunakan medium fotografi konvensional.
3. Mekanisme keamanan internal program, termasuk penggunaan 32-bit *header* untuk presisi ekstraksi dan *error handling* yang kokoh dalam memilah citra orisinal dan citra stego, telah memastikan keutuhan pesan rahasia saat direkonstruksi tanpa merusak stabilitas program.

LINK SOURCE CODE

https://github.com/rajamahma2005-bot/Steganografi-Dengan-Medium-Aset-Minecraft

Acknowledgment

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan kekuatan yang diberikan, sehingga penulis dapat menyelesaikan makalah ini tepat pada waktu yang telah ditentukan. Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada Bapak Dr. Ir. Rinaldi Munir, M.T., atas bimbingan yang tak ternilai, perkuliahan yang sangat bermanfaat, serta kesempatan yang diberikan untuk menulis makalah penelitian ini. Penulis juga menyampaikan apresiasi yang setulus-tulusnya kepada keluarga atas dukungan dan dorongan semangat yang tiada henti di sepanjang perjalanan akademik ini. Akhir kata, penulis mengucapkan terima kasih kepada seluruh rekan-rekan mahasiswa kelas mata kuliah II4021 Kriptografi atas kerja sama yang baik dan semangat saling mendukung selama semester ini.

References

[1] A. Alenizi, M. S. Mohammadi, A. A. Al-Hajji, and A. S. Ansari, "A review of image steganography based on multiple hashing algorithm," *Computers, Materials & Continua*, 2024, doi: 10.32604/cmc.2024.051826.

[2] M. S. Abuali, C. B. M. Rashidi, R. A. A. Raof, K. N. F. K. Azir, S. S. Hussein, and A. Q. Abd-Alhasan, "Enhancing Security with Multi-level Steganography: A Dynamic Least Significant Bit and Wavelet-Based Approach," *Mathematical Modelling of Engineering Problems*, vol. 11, no. 6, pp. 1403-1416, Jun. 2024, doi: 10.18280/mmep.110602.

[3] D. R. I. M. Setiadi, "Payload enhancement on least significant bit image steganography using edge area dilation," *International Journal of Electronics and Telecommunications*, vol. 65, no. 2, pp. 287-292, 2019, doi: 10.24425/ijet.2019.126312.

[4] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K. H. Jung, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication*, vol. 65, pp. 46-66, Jul. 2018, doi: 10.1016/j.image.2018.03.012.

[5] A. D. Ker, "Batch steganography and pooled steganalysis," in *Information Hiding*, vol. 4437, Berlin, Heidelberg: Springer, 2006, pp. 265-281, doi: 10.1007/978-3-540-74124-4_18.

[6] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313-336, 1996, doi: 10.1147/sj.353.0313.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026

Raja M Arkan, 10123065