Research Proposal Title: Multiple Embedding Steganography for RGB Image

---

Submitted by: Dimas Anwar Aziz

NIM: 203022410012

Concentration: Cyber Security

E-mail address:

dimasanwaraziz@student.telkomuniversity.ac.id

Supervisor (I) : Prof. Ari Moesriami Barmawi, Ph.D.

Supervisor (II):

08102024

Topic Summary:

This is a summary and a word length of 300 words. Summary is written briefly from the entire contents of the thesis/proposal and so on until it is finished.

1 INTRODUCTION (assessment 1)

Dalam era digital yang semakin berkembang, keamanan data digital menjadi prioritas utama. Informasi pribadi, data bisnis, dan data kritikal lainnya harus dilindungi dari ancaman kebocoran atau akses yang tidak sah. Dalam menghadapi tantangan ini, kriptografi dan steganografi muncul sebagai solusi penting untuk meningkatkan keamanan data digital.

Steganografi merupakan teknik menyembunyikan pesan di dalam suatu media penyisipan pesan atau cover image, sehingga keberadaan pesan rahasia yang disisipkan tidak dapat dilihat secara langsung [1]. Steganografi memungkinkan pertukaran pesan rahasia melalui penyembunyian informasi pada berbagai media digital seperti gambar, video, dan audio, tanpa menimbulkan kecurigaan [2]. Steganografi dapat dipandang sebagai kelanjutan kriptografi dan dalam prakteknya pesan rahasia dienkripsi terlebih dahulu, kemudian cipherteks disembunyikan di dalam media lain sehingga pihak ketiga tidak menyadari keberadaannya. Pesan rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti aslinya [3].

Di antara berbagai media digital yang dapat digunakan untuk steganografi, gambar menjadi pilihan yang populer karena beberapa alasan. Pertama, gambar digital tersedia secara luas dan pertukaran gambar di internet adalah hal yang umum, sehingga tidak menimbulkan kecurigaan. Kedua, gambar memiliki kapasitas untuk menyembunyikan informasi. Ketiga, manipulasi pixel pada gambar dapat dilakukan dengan berbagai teknik. Namun, tantangan utama

dalam steganografi gambar adalah menemukan keseimbangan antara kapasitas penyembunyian data, kualitas gambar, dan ketahanan terhadap deteksi [4].

Penelitian sebelumnya telah menunjukkan kemajuan signifikan dalam meningkatkan kapasitas penyembunyian data pada gambar grayscale. Kingsley et al. [5] berhasil meningkatkan kapasitas penyembunyian data hingga 450% menggunakan metode code base. Meski begitu, penggunaan gambar grayscale membatasi aplikasi praktis karena hanya memiliki satu channel pencahayaan. Di sisi lain, Fikri et al. [4] mendemonstrasikan bahwa penerapan steganografi pada gambar berwarna memiliki ketahanan yang baik, meskipun gambar dengan dominasi warna hitam menunjukkan kompatibilitas yang kurang optimal. Berdasarkan temuan-temuan tersebut, penelitian ini bertujuan untuk memperluas metode embedding ke gambar berwarna yang memiliki tiga channel warna (Merah, Hijau, dan Biru). Fokus utama adalah untuk meningkatkan kapasitas penyembunyian data sambil mempertahankan kualitas visual dan ketahanan terhadap deteksi. Penelitian ini juga akan mengevaluasi kompatibilitas metode yang diusulkan dengan berbagai jenis gambar berwarna, termasuk yang memiliki area gelap yang luas. Dengan mengoptimalkan penggunaan ketiga channel warna, diharapkan dapat dicapai peningkatan signifikan dalam kapasitas penyembunyian data dibandingkan dengan metode yang hanya menggunakan gambar grayscale, sambil tetap mempertahankan kealamiahan gambar yang dihasilkan.

Aspek pengetahuan dalam penelitian ini berfokus pada teknik steganografi untuk gambar berwarna, metode steganografi berbasis kode, dan analisis ku-

alitas gambar digital. Hal ini mencakup pemahaman mendalam tentang manipulasi channel warna RGB, teknik penyisipan data yang efektif, serta metode evaluasi kualitas gambar stego yang dihasilkan. Penekanan khusus diberikan pada optimasi penggunaan ketiga channel warna untuk memaksimalkan kapasitas penyimpanan data sambil mempertahankan kualitas visual.

Dari sisi pengguna, penelitian ini ditujukan untuk beberapa kelompok utama. Pertama, profesional keamanan informasi yang membutuhkan solusi penyembunyian data yang andal dan aman. Kedua, peneliti di bidang steganografi dan keamanan data yang dapat memanfaatkan metode ini untuk pengembangan lebih lanjut. Ketiga, pengembang aplikasi keamanan data yang memerlukan komponen steganografi yang efisien dan mudah diintegrasikan.

Aspek kegunaan penelitian mencakup berbagai aplikasi praktis dalam sistem data hiding, perlindungan hak cipta, dan watermarking digital. Metode yang dikembangkan dirancang dengan mempertimbangkan kemudahan implementasi, efisiensi komputasi, dan fleksibilitas penggunaan. Fokus utama adalah mencapai keseimbangan optimal antara kapasitas penyembunyian data, kualitas gambar hasil, dan ketahanan terhadap berbagai teknik deteksi steganografi.

Infrastruktur yang digunakan dalam penelitian ini berbasis Mac mini M2 dengan spesifikasi teknis yang mendukung pemrosesan gambar yang efisien. Sistem dilengkapi dengan prosesor Apple M2 chip yang memiliki 8-core CPU dan 10-core GPU, didukung oleh 8GB memori dan penyimpanan SSD 256GB. Penggunaan sistem operasi macOS memberikan platform yang stabil untuk pengembangan dan pengujian metode steganografi yang diusulkan.

2 Preliminary Literature Review (assessment 2)

Kingsley et al (2020) pada papernya membahas tentang rendahnya kapasitas penyisipan dan nilai PSNR (Peak Signal-to-Noise Ratio) pada skema steganografi berbasis kode yang ada, yang hanya mencapai 150% kapasitas dan 48 dB kualitas visual gambar stegano. Untuk mengatasi masalah ini, metode yang diusulkan adalah teknik penyisipan ganda (multiple embedding), yang bertujuan untuk menyisipkan bit rahasia lebih dari sekali pada LSB (Least Significant Bit) dari piksel yang dipilih berdasarkan kunci rahasia. Parameter yang digunakan untuk mengukur keberhasilan metode ini meliputi kapasitas penyisipan dan nilai PSNR dari gambar stego yang dihasilkan. Kelebihan dari metode ini adalah kemampuan untuk mencapai kapasitas penyisipan yang lebih tinggi hingga 450% dan nilai PSNR yang meningkat menjadi 51 dB, serta meningkatkan ketahanan terhadap serangan seperti scratching dan kompresi JPEG. Namun, penelitian ini masih diterapkan pada gambar hitam putih [5].

Pada penelitian yang dilakukan oleh Dwi Andika et al (2020), mereka membahas keterbatasan kapasitas penyimpanan dalam teknik steganografi menggunakan algoritma GifShuffle pada citra GIF. Tantangan utama yang dihadapi adalah kesulitan dalam menyisipkan pesan teks dengan ukuran besar. Untuk mengatasi masalah ini, mereka memodifikasi nilai bit dalam penyimpanan pesan, yang bertujuan meningkatkan kapasitas tanpa mengorbankan kualitas visual gambar. Uji coba menggunakan parameter seperti ukuran data yang dapat disisipkan serta kualitas gambar menunjukkan bahwa metode ini ber-

hasil meningkatkan kapasitas penyimpanan hingga lebih dari 256 KB tanpa mengurangi kualitas gambar secara signifikan. Namun, metode ini memerlukan modifikasi tambahan pada algoritma GifShuffle yang dapat meningkatkan kompleksitas implementasi dan pemrosesan [6].

Susanto et al (2020) dalam penelitiannya menggabungkan metode steganografi Least Significant Bit (LSB) dengan algoritma enkripsi RSA untuk meningkatkan keamanan dalam penyisipan pesan terenkripsi pada gambar. Parameter yang digunakan untuk mengukur kualitas gambar setelah penyisipan adalah Peak Signal-to-Noise Ratio (PSNR), dengan hasil yang menunjukkan PSNR tertinggi sebesar 78 dB untuk pesan berukuran 1024 bit. Kelebihan metode ini adalah perubahan kualitas gambar yang hampir tidak terlihat, namun kekurangannya adalah kompleksitas yang meningkat seiring dengan ukuran pesan dan besarnya kunci RSA yang digunakan, yang memengaruhi performa [7].

Basri et al (2021) meneliti penggunaan metode steganografi dengan teknik Least Significant Bit (LSB) untuk menyembunyikan gambar di dalam gambar lainnya, khususnya dalam konteks interaksi sosial melalui media digital. Penelitian ini mengukur rasio ukuran gambar tersembunyi terhadap gambar cover serta kemampuan gambar cover mempertahankan kualitas visualnya. Hasilnya menunjukkan bahwa metode ini efektif menyembunyikan informasi tanpa perubahan signifikan pada gambar cover. Namun, metode ini memiliki keterbatasan dalam menangani gambar dengan transparansi, karena hanya memperhitungkan komponen warna merah, hijau, dan biru (RGB), tanpa memperhitungkan komponen alpha [8].

Penelitian Wiranata et al (2021) berfokus pada penyisipan pesan rahasia dalam gambar dan audio menggunakan metode Least Significant Bit (LSB) yang dikombinasikan dengan enkripsi Caesar Chipper dan Rivest Code 4 untuk menjaga kerahasiaan data. Penelitian ini menguji kemampuan aplikasi untuk menyembunyikan dan mengambil pesan secara utuh, serta perubahan ukuran file gambar dan audio setelah proses penyisipan. Kelebihan metode ini adalah kemampuannya menjaga kerahasiaan tanpa perubahan signifikan pada kualitas gambar atau suara. Namun, ada perubahan ukuran file yang diakibatkan oleh proses enkripsi dan penyisipan pesan [9].

Penelitian yang dilakukan oleh Abdillah et al (2023) membahas tentang perlindungan data dan akses dengan menggunakan steganografi melalui teknik Least Significant Bit (LSB), di mana teks disisipkan dalam gambar melalui perubahan nilai pixel terkecil. Parameter yang diukur mencakup akurasi penyisipan teks dan kualitas gambar yang tetap terjaga setelah proses encoding dan decoding. Kelebihan utama metode ini adalah kemampuannya menyembunyikan informasi tanpa mengubah kualitas visual gambar secara signifikan. Namun, metode ini memiliki keterbatasan dalam kapasitas penyisipan teks dan rentan terhadap serangan manipulasi gambar yang dapat merusak data tersembunyi [10].

Tabel 1 menunjukkan perbandingan hasil penelitian terdahulu.

Table 1: Comparison of Previous Research

Author	**Method**	**Advantages**	**Disadvantages**
Kingsley et al. (2020)	Multiple embedding technique with secret key-based pixel selection	Increased capacity up to 450%, PSNR improved to 51dB	Limited to grayscale images only
Andika et al. (2020)	Modified GifShuffle algorithm for GIF images	Increased storage capacity over 256KB	Complex implementation, limited to GIF format
Susanto et al. (2020)	Combined LSB with RSA encryption	High PSNR (78dB) for 1024-bit messages	Increased complexity with larger messages and RSA keys
Basri et al. (2021)	LSB technique for image-in-image hiding	Effective information hiding with minimal visual changes	Limited handling of transparency (alpha channel)
Wiranata et al. (2021)	LSB combined with Caesar Cipher and RC4	Good security with dual encryption	File size increases after embedding
Abdillah et al. (2023)	LSB technique for text embedding	Maintains visual quality effectively	Limited text capacity, vulnerable to image manipulation

3 Problem Statement (Assessment 2)

Penelitian sebelumnya dalam steganografi berbasis kode telah menunjukkan kemajuan signifikan. Kingsley et al. [5] berhasil meningkatkan kapasitas penyembunyian data dari sekitar 150% menjadi 450%, yang merupakan peningkatan substansial untuk case yang membutuhkan kapasitas tinggi. Meskipun demikian, penelitian pada gambar grayscale [3] masih membatasi pengaplikasiannya karena kurangnya fleksibilitas dalam penggunaan channel warna.

Sementara itu, penerapan steganografi pada gambar berwarna (RGB) mem-

buka peluang untuk peningkatan kapasitas, namun juga menghadirkan tantangan baru. Fikri et al. [4] menemukan bahwa meskipun gambar RGB menawarkan robbustnes yang baik, terdapat masalah kompatibilitas pada gambar dengan dominasi warna gelap [11].

4 Objective and Hypothesis (assessment 3)

Penelitian ini memiliki dua tujuan utama beserta hipotesis terkait. Pertama, mengembangkan teknik steganografi berbasis kode dengan multiple embedding untuk gambar RGB dan meningkatkan kapasitas penyembunyian data minimal 50% dibandingkan dengan metode sebelumnya, dengan harapan bahwa metode ini akan meningkatkan kapasitas penyembunyian data secara signifikan dan penerapan pada gambar RGB akan menghasilkan peningkatan kapasitas lebih besar dibandingkan pada gambar grayscale. Kedua, mempertahankan PSNR di atas 40 dB untuk gambar stego, dengan hipotesis bahwa peningkatan kapasitas melalui multiple embedding tidak akan menurunkan kualitas visual secara signifikan [12].

5 Research Method (assessment 4,5)

Penelitian ini akan menggunakan pendekatan eksperimental untuk menguji hipotesis yang diajukan. Metode penelitian mencakup beberapa tahap utama, mengintegrasikan konsep dasar steganografi dan mengatasi tantangan khusus steganografi berbasis kode dengan teknik penyisipan ganda.

5.1 Identifikasi Kebutuhan

Berdasarkan analisis literatur dan tujuan penelitian, kebutuhan utama untuk pengembangan metode steganografi berbasis kode dengan penyisipan ganda:

1. Peningkatan kapasitas penyembunyian data minimal 50% dari metode yang diusulkan oleh Kingsley et al. (2020)
2. Teknik mempertahankan kualitas visual (PSNR $>$ 40 dB)
3. Algoritma penyisipan dengan multiple embedding pada gambar RGB

Identifikasi kebutuhan ini akan menjadi dasar untuk proses desain dan implementasi sistem steganografi yang diusulkan.

5.2 Proses Desain

Berdasarkan kebutuhan yang diidentifikasi, penelitian ini akan merancang algoritma untuk steganografi berbasis kode dengan multiple embedding.

Arsitektur sistem umum akan mencakup

1. **Secret Image**

   Secret Image merupakan informasi rahasia yang akan disembunyikan dalam sistem steganografi, yang dapat mencakup gambar dalam berbagai format (JPG, PNG) yang akan dilindungan keamanan dan kerahasiaan. Pemilihan jenis data rahasia ini akan mempengaruhi kapasitas penyimpanan yang diperlukan dan metode penyisipan yang digunakan dalam proses steganografi.

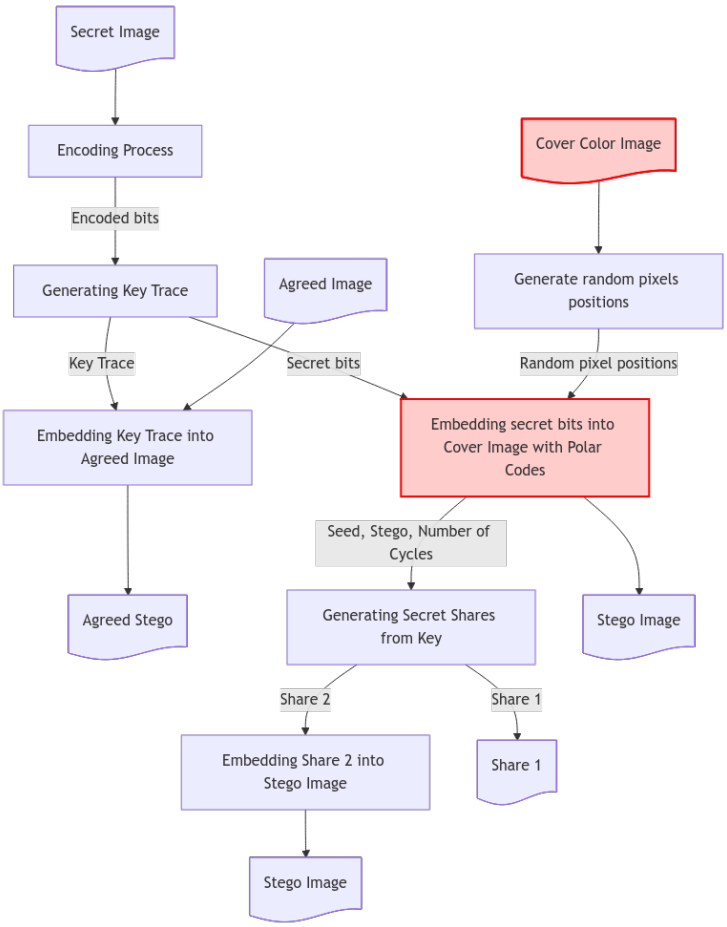

Figure 1: Design method

Gambar ini merupakan inputan dinamis user untuk nanti disisipkan pada cover image yang berwarna, dan nanti akan dicek terkait berapa kapasitas dan noise (kualitas) yang dihasilkan dari hasil penyisipan secret image ini.

2. **Encoding Process**

Proses *encoding* bertujuan untuk menyisipkan pesan (teks) ke dalam gambar dengan menggunakan **Least Significant Bit (LSB)** dari setiap kom-

ponen warna pixel (R, G, B).

Berikut adalah penjelasan langkah-langkah proses encoding:

(a) **Persiapan Data**

- **Input:**
  - Gambar yang akan digunakan sebagai media penyisipan.
  - Pesan teks yang ingin disembunyikan.
- **Tambahan Terminator:** Pesan diberi terminator (\000) di akhir sebagai penanda bahwa pesan telah selesai.
  - Contoh:

    Pesan awal: "Hello, World!"

    Pesan yang akan diproses: "Hello, World!\000"
- **Konversi ke Bit:** Setiap karakter diubah menjadi 8 bit (format byte).
  - Contoh huruf "H" (ASCII 72): 01001000.

(b) **Iterasi Gambar Pixel-per-Pixel**

Gambar diakses pixel-per-pixel, dan untuk setiap pixel:

- Komponen warna R (Red), G (Green), dan B (Blue) diekstrak.
- Setiap komponen warna memiliki nilai 8 bit (0-255).

(c) **Menyisipkan Bit ke LSB**

Dari setiap byte pesan (8 bit), bit pesan disisipkan satu per satu ke dalam LSB (Least Significant Bit) dari komponen R, G, dan B. LSB

adalah bit paling rendah dari nilai komponen warna. Mengubah LSB tidak akan mengubah warna secara signifikan.

- **Contoh Menyisipkan Bit:** Misalkan:
  - Nilai awal komponen warna:

$$R = 10110011, \quad G = 11001001, \quad B = 11111110$$

  - Bit pesan yang akan disisipkan: 0, 1, 1.
- **Langkah Perubahan:**
  - R (Red):
    - LSB sebelum: 1
    - Ganti dengan bit pesan 0.
    - Nilai R menjadi: 10110010.
  - G (Green):
    - LSB sebelum: 1
    - Ganti dengan bit pesan 1.
    - Nilai G tetap: 11001001.
  - B (Blue):
    - LSB sebelum: 0
    - Ganti dengan bit pesan 1.
    - Nilai B menjadi: 11111111.

- **Hasil Pixel:**

$$R = 10110010, \quad G = 11001001, \quad B = 11111111.$$

Proses terus diulang untuk setiap pixel, bergerak dari kiri ke kanan, atas ke bawah. Jika semua bit pesan telah disisipkan, pixel sisanya tidak diubah.

Encoding berhenti setelah semua bit dari pesan (termasuk terminator `\000`) berhasil disisipkan. Gambar baru (dengan pesan tersembunyi) disimpan dalam format PNG agar tidak terjadi kompresi lossy.

(d) **Ilustrasi**

Misalkan gambar memiliki pixel dengan:

```
Pixel (R, G, B) = (10110011, 11001001, 11111110)
```

Dan pesan yang akan disisipkan adalah huruf ”A”.

```
"A" dalam ASCII = 65 → 01000001 (8 bit).
```

Proses Iterasi:

Bit Pesan	Komponen Warna	LSB Lama	LSB Baru	Hasil Akhir
0	R (Red)	1	0	10110010
1	G (Green)	1	1	11001001
0	B (Blue)	0	0	11111110
0	R (Red)	0	0	10110010
0	G (Green)	1	0	11001000
0	B (Blue)	0	0	11111110
1	R (Red)	0	1	10110011
0	G (Green)	0	0	11001000

Table 2: Proses Penyisipan Bit Pesan ke dalam Komponen Warna

Output Gambar:

Gambar baru memiliki nilai komponen warna yang dimodifikasi pada LSB, tetapi perubahan warna tidak terlihat oleh mata manusia. Gambar ini berisi pesan yang telah disisipkan secara tersembunyi.

3. Generating Key Trace

Proses pembuatan kunci yang akan menentukan di mana dan bagaimana data akan disembunyikan dalam gambar. Sistem menggunakan cara khusus untuk membuat pola acak yang memastikan data tersebar merata dalam gambar. Cara ini membuat sistem lebih aman karena menambah lapisan pengamanan, sekaligus memudahkan pengambilan data kembali tanpa merusak gambar. Jika terjadi masalah atau ada yang mencoba mengambil data secara ilegal, sistem ini juga membantu mengelola dan memulihkan

data dengan lebih baik.

4. Agreed Image

   Gambar yang telah ditentukan bersama antara pengirim dan penerima sebagai media penyimpanan jejak kunci steganografi. Pemilihan gambar ini dilakukan dengan mempertimbangkan berbagai karakteristik teknis seperti kompleksitas tekstur, distribusi warna, dan noise level yang optimal untuk mendukung proses penyembunyian informasi tanpa menimbulkan kecurigaan. Gambar yang dipilih juga harus memiliki kapasitas yang memadai untuk menampung jejak kunci sambil tetap mempertahankan kualitas visual yang baik setelah proses penyisipan.

5. Embedding Key Trace into Agreed Image

   Tahap kritis dimana key trace disisipkan ke dalam gambar yang telah disepakati sebelumnya menggunakan teknik steganografi. Proses ini memerlukan presisi untuk memastikan integritas data dan imperceptibility optimal. Penyisipan dilakukan dengan mempertimbangkan karakteristik gambar dan distribusi nilai pixel untuk mengoptimalkan keseimbangan antara kapasitas penyimpanan dan kualitas visual. Teknik steganografi yang digunakan juga mempertimbangkan aspek keamanan untuk mencegah deteksi dan ekstraksi unauthorized.

6. Agreed Stego

   Hasil akhir dari proses penyisipan jejak kunci ke dalam gambar yang telah disepakati bersama. Gambar ini memiliki karakteristik visual yang identik

dengan gambar asli untuk menghindari kecurigaan, namun di dalamnya telah tertanam informasi rahasia dalam bentuk jejak kunci yang akan digunakan untuk proses ekstraksi data. Kualitas gambar tetap terjaga meskipun telah melalui proses penyisipan, dengan perubahan nilai pixel yang minimal dan tidak terdeteksi oleh mata manusia. Gambar ini berperan penting sebagai pembawa informasi kontrol yang diperlukan untuk proses steganografi selanjutnya.

7. Embedding secret bits into Cover Image

   Embedding secret bits into cover image merupakan proses penyisipan bit-bit data rahasia ke dalam gambar cover dilakukan dengan memanfaatkan algoritma steganografi yang canggih dan teroptimasi, dikombinasikan dengan polar codes untuk meningkatkan reliabilitas transmisi data. Proses ini melibatkan analisis karakteristik gambar cover untuk menentukan lokasi optimal penyisipan data, kemudian menggunakan teknik transformasi yang presisi untuk menyisipkan bit-bit informasi rahasia.

   Polar codes, yang diperkenalkan oleh Arikan, digunakan untuk mengkodekan data rahasia sebelum penyisipan. Teknik ini memanfaatkan fenomena polarisasi channel untuk mencapai kapasitas Shannon dengan kompleksitas encoding dan decoding yang rendah ($O(N \log N)$).

   Algoritma yang digunakan dirancang khusus untuk memaksimalkan kapasitas penyimpanan data sambil tetap mempertahankan kualitas visual gambar, dengan fokus khusus pada minimalisasi distorsi yang dapat terde-

teksi baik secara visual maupun statistik. Penggunaan polar codes memberikan keuntungan tambahan dalam hal error correction dan ketahanan terhadap noise channel. Proses ini juga mempertimbangkan aspek keseimbangan antara efisiensi penyisipan dan ketahanan terhadap berbagai teknik steganalisis.

8. Cover Image

   Cover image merupakan gambar yang berfungsi sebagai media utama untuk penyembunyian data rahasia, dipilih dengan pertimbangan khusus berdasarkan kompleksitas tekstur, distribusi warna, dan karakteristik statistik yang optimal untuk mendukung proses penyisipan informasi. Pemilihan cover image yang tepat sangat kritis karena akan mempengaruhi kapasitas penyimpanan data, ketahanan terhadap deteksi, dan kualitas visual hasil akhir steganografi. Gambar dengan area tekstur yang kompleks dan variasi nilai pixel yang tinggi umumnya lebih ideal karena dapat menyembunyikan perubahan yang diakibatkan oleh proses penyisipan data dengan lebih efektif.

9. Generate random pixels positions

   Algoritma untuk menghasilkan posisi piksel secara acak namun tetap deterministik berdasarkan seed value yang ditentukan. Proses ini menggunakan pembangkit bilangan pseudo-random yang telah terverifikasi untuk menghasilkan sekuens posisi piksel yang terdistribusi merata di seluruh gambar. Pendekatan deterministik ini penting untuk memastikan

konsistensi dalam proses penyisipan dan ekstraksi data, sekaligus meningkatkan keamanan dengan menciptakan pola penyebaran data yang sulit diprediksi oleh pihak yang tidak berwenang.

10. Generating Secret Shares from Key

    Proses pembagian kunci rahasia menjadi beberapa bagian (shares) menggunakan skema Shamir's Secret Sharing. Teknik ini memungkinkan pembagian kunci menjadi n bagian dengan threshold k, di mana kunci asli hanya dapat direkonstruksi jika minimal k bagian tersedia. Proses ini meningkatkan keamanan dengan mendistribusikan risiko dan mencegah single point of failure dalam pengelolaan kunci.

11. Embedding Share 2 into Stego Image

    Proses penyisipan bagian kedua (share 2) dari kunci yang telah dibagi ke dalam gambar stego menggunakan teknik steganografi yang telah dioptimasi. Penyisipan dilakukan dengan mempertimbangkan aspek keseimbangan antara keamanan data dan kualitas visual gambar. Proses ini dirancang untuk memastikan share dapat diekstrak kembali dengan akurat tanpa merusak informasi yang tersembunyi lainnya di dalam gambar stego.

12. Share 1

    Bagian pertama (share 1) dari kunci yang telah dibagi disimpan secara terpisah dengan menerapkan protokol keamanan berlapis. Share 1 memiliki peran penting karena diperlukan bersama dengan share 2 untuk merekon-

struksi kunci asli. Penyimpanan dan pengelolaan share 1 mengikuti standar keamanan yang ketat untuk mencegah akses tidak sah.

13. Stego Image

    Gambar hasil akhir setelah proses penyisipan data rahasia, yang dirancang untuk mempertahankan karakteristik visual yang identik dengan gambar cover asli. Meskipun mengandung informasi tersembunyi, gambar stego tidak menunjukkan perbedaan yang dapat terdeteksi secara visual maupun statistik sederhana. Kualitas gambar dijaga melalui optimasi algoritma penyisipan yang mempertimbangkan karakteristik persepsi visual manusia.

14. Final Stego Image

    Versi terakhir dari gambar stego yang telah dilengkapi dengan kunci publik dan semua metadata yang diperlukan untuk proses ekstraksi dan verifikasi yang aman. Gambar ini menyediakan mekanisme pemulihan data yang komprehensif namun tetap mempertahankan aspek keamanan dan kerahasiaan informasi yang tersembunyi. Struktur data tambahan diintegrasikan dengan cara yang tidak mengganggu kualitas visual keseluruhan.

15. **Decoding Process**

    Proses decoding bertujuan untuk mengekstrak pesan tersembunyi yang telah disisipkan ke dalam gambar menggunakan metode **Least Significant Bit (LSB)**. Berikut adalah langkah-langkah decoding:

    (a) **Persiapan Data**

- **Input:** Gambar yang telah di-encode dengan pesan tersembunyi.
- **Tujuan:** Mengekstrak bit dari komponen warna setiap piksel dan menyusun kembali menjadi karakter pesan.

(b) **Iterasi Gambar Pixel-per-Pixel**

Gambar diakses pixel-per-pixel (dari kiri ke kanan, atas ke bawah), lalu dilakukan pembacaan bit dari komponen warna:

- Komponen warna **R (Red)**, **G (Green)**, dan **B (Blue)** diekstrak satu per satu.
- Dari setiap komponen warna, **Least Significant Bit (LSB)** dibaca karena LSB digunakan untuk menyisipkan bit pesan.

Contoh:

$$\begin{aligned} R &= 10110010 \\ G &= 11001001 \\ B &= 11111111 \end{aligned}$$

LSB dari setiap komponen warna:

- LSB $R = 0$
- LSB $G = 1$
- LSB $B = 1$

Bit yang diperoleh dari piksel tersebut adalah: $0, 1, 1$.

(c) **Penyusunan Bit Menjadi Byte**

Setiap 8 bit yang diperoleh dari LSB disusun kembali menjadi 1 byte.

- LSB pertama masuk ke bit paling kiri.

- LSB berikutnya diikuti hingga terbentuk 8 bit (1 byte).

Contoh:

LSB yang dibaca: $0, 1, 0, 0, 1, 0, 0, 0$

Hasil: 01001000, yang dalam ASCII adalah huruf **H**.

Proses pembacaan bit dari LSB dilanjutkan hingga menemukan **null terminator** (\000), yaitu byte dengan nilai `0x00` (00000000 dalam biner).

Contoh:

`H → e → l → l → o → , → W → o → r → l → d → !`

Diikuti oleh null byte \000, proses decoding berhenti.

Byte yang telah dibaca dikonversi kembali menjadi karakter ASCII dan disusun menjadi string pesan yang telah disembunyikan.

Ringkasan Proses Decoding:

- Baca LSB dari komponen warna R, G, dan B secara berurutan.
- Susun LSB menjadi 8 bit untuk membentuk 1 byte.
- Ulangi proses hingga menemukan **null terminator** (\000).
- Konversi byte yang terkumpul menjadi string pesan.
- (Opsional) Validasi pesan dengan menghasilkan **key trace** menggunakan **SHA-256**.

5.3 Proses Implementasi

Algoritma yang dirancang akan diimplementasikan menggunakan Python, memanfaatkan library seperti OpenCV untuk pemrosesan gambar dan NumPy untuk operasi numerik. Implementasi ini dapat mendukung gambar RGB maupun grayscale untuk memungkinkan analisis komparatif.

5.3.1 Persiapan Environment Development

Mulai dengan instalasi Python sebagai bahasa pemrograman utama. Instal library OpenCV yang akan digunakan untuk membaca, menulis, dan memanipulasi gambar digital. Tambahkan NumPy untuk mendukung operasi matematika dan array yang diperlukan dalam pemrosesan gambar.

5.3.2 Pengembangan Modul Utama

1. Implementasi fungsi pembacaan gambar:
    (a) Pembacaan gambar format RGB menggunakan OpenCV
    (b) Pembacaan gambar format grayscale menggunakan OpenCV
    (c) Validasi format input gambar
2. Implementasi fungsi pemrosesan pixel:
    (a) Modifikasi nilai pixel untuk penyisipan data
    (b) Pengecekan kapasitas penyisipan
    (c) Validasi integritas data

3. Implementasi fungsi ekstraksi data:

   (a) Pembacaan nilai pixel termodifikasi

   (b) Ekstraksi data tersembunyi

   (c) Rekonstruksi pesan asli

4. Implementasi fungsi evaluasi kualitas:

   (a) Perhitungan eSNR (Peak Signal-to-Noise Ratio)

   (b) Perhitungan MSE (Mean Square Error)

   (c) Analisis imperceptibility hasil steganografi

5.3.3 Implementasi Algoritma Steganografi

Dalam implementasi algoritma steganografi, pengembangan sistem dimulai dengan implementasi mekanisme penyisipan ganda yang komprehensif. Proses ini mencakup identifikasi area potensial untuk penyisipan data, pengembangan metode penyisipan bertingkat, serta pengelolaan metadata untuk multiple embedding. Koordinasi antar layer penyisipan menjadi aspek krusial untuk memastikan integritas data yang disisipkan.

Sistem yang dikembangkan dirancang untuk mendukung berbagai format gambar, dengan fokus utama pada penanganan gambar RGB (24-bit) dan grayscale (8-bit). Adaptasi algoritma dilakukan untuk mengakomodasi kedua format tersebut, disertai dengan optimasi performa yang memungkinkan sistem bekerja secara efisien pada berbagai jenis gambar carrier.

Optimasi kapasitas penyisipan menjadi salah satu prioritas utama, dengan target minimal mencapai 50% dari ukuran carrier. Hal ini dicapai melalui implementasi teknik bit-plane slicing yang efisien, didukung oleh mekanisme kompresi data payload, serta pemilihan pixel secara adaptif untuk memaksimalkan kapasitas penyimpanan tanpa mengorbankan kualitas visual.

Preservasi kualitas visual menjadi aspek fundamental dalam pengembangan, dengan target PSNR yang ditetapkan di atas 40 dB. Sistem mengimplementasikan berbagai strategi untuk meminimalisasi distorsi visual, termasuk teknik penyebaran perubahan pixel dan adaptasi terhadap karakteristik spesifik dari gambar carrier.

Implementasi dilengkapi dengan mekanisme validasi yang komprehensif, mencakup verifikasi integritas data, pengukuran kapasitas efektif, evaluasi kualitas visual, serta pengujian robustness. Serangkaian pengujian ini memastikan bahwa sistem yang dikembangkan memenuhi seluruh requirement yang telah ditetapkan, baik dari segi kapasitas, kualitas, maupun keamanan data.

5.3.4 Pengembangan Sistem Pengujian

Buat modul pengujian untuk:

- Mengukur kapasitas maksimum penyisipan data - Mengevaluasi kualitas visual menggunakan PSNR dan SSIM - Menguji ketahanan terhadap teknik steganalisis - Membandingkan dengan metode LSB konvensional

5.3.5 Analisis dan Evaluasi

Lakukan analisis komprehensif meliputi:

- Pengujian statistik menggunakan uji-t dan ANOVA - Evaluasi visual subjektif - Pengukuran metrik kinerja (kapasitas, PSNR, SSIM) - Evaluasi efisiensi komputasi

5.3.6 Dokumentasi

Menyiapkan dokumentasi lengkap yang mencakup:

1. Source code dengan komentar yang jelas
2. Manual penggunaan sistem
3. Hasil pengujian dan analisis
4. Rekomendasi untuk pengembangan lebih lanjut

5.4 Desain Eksperimen dan Pengumpulan Data

Untuk mengevaluasi kinerja metode yang diusulkan dibandingkan dengan metode lain, sebuah eksperimen dilakukan berdasarkan empat ukuran kuantitatif utama. Gambar 2 mengilustrasikan desain eksperimen yang disarankan untuk metode yang diusulkan.

5.4.1 Imperceptibility

Skema steganografi tidak akan terlihat jika mata manusia tidak dapat membedakan antara sampul dan gambar stego. Yang menjadi perhatian dalam bagian

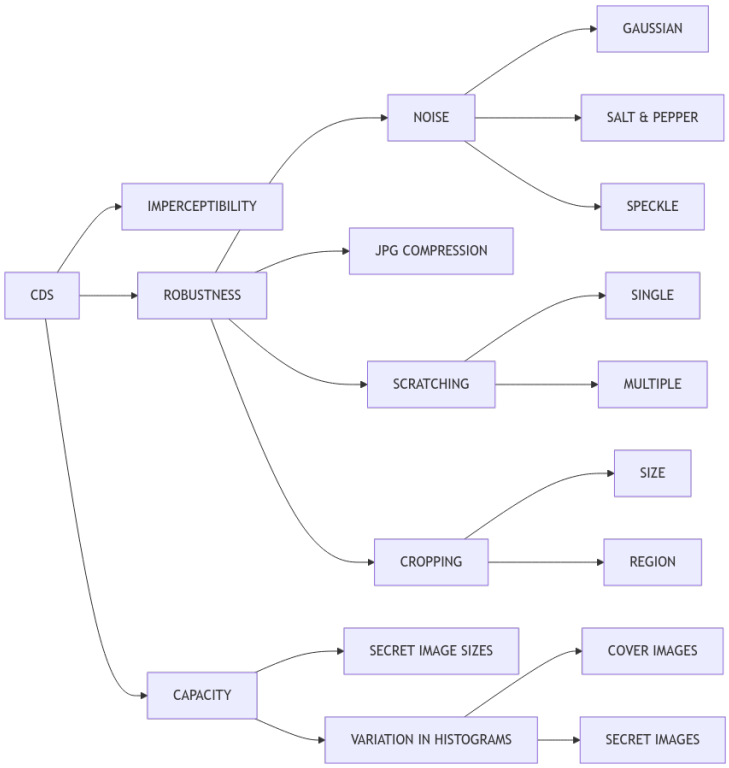

Figure 2: Design experiment

ini adalah kualitas visual gambar stego yang dihasilkan. Kelompok gambar rahasia dengan ukuran gambar yang sama dan distribusi nada yang beragam terbentuk. Setiap gambar rahasia dalam kelompok tertentu disematkan di sampul menggunakan metode yang diusulkan dan metode A. M. Molaei.

Ketidaksadaran kemudian diukur dengan Mean Square Error (MSE) dan Peak to Signal Noise Ratio (PSNR). PSNR yang lebih tinggi menunjukkan bahwa gambar stego lebih mirip dengan gambar asli, yang berarti kualitas visual lebih baik.

5.4.2 Robbustnes

Pada bagian ini menguji kemampuan metode yang diusulkan untuk mengatasi berbagai serangan. Hal ini memungkinkan penerima untuk mengambil kembali pesan rahasia yang telah dihancurkan jika terjadi penghancuran citra stego secara sengaja atau tidak sengaja. Ukuran kuantitatif yang akan ditentukan adalah PSNR dari citra rahasia yang dipulihkan.

Setelah setiap skenario penyematan dilakukan seperti yang dijelaskan, gambar stego yang dihasilkan diekspos di bawah serangan berikut

1. Noise Attack: Gambar stego diserang dengan beberapa metode di bawah dengan intensitas yang berbeda.

   (a) **Gaussian Noise**: Nama ini diambil dari Carl Friedrich Gauss [13]. Gaussian Noise merupakan noise statistik yang memiliki fungsi kerapatan probabilitas (PDF) yang sama dengan distribusi normal, yang juga dikenal sebagai distribusi Gaussian. Dengan kata lain, nilai yang dapat diambil oleh noise tersebut berdistribusi Gaussian. Untuk Gaussian Noise, dua variasi varians dipertimbangkan dalam eksperimen, yaitu Gaussian noise: mean 0 variance 0,01 dan Gaussian noise: mean 0 variance 0,1.

   (b) **Salt and Pepper Noise**: Ini adalah bentuk noise yang juga terlihat pada gambar. Ini juga dikenal sebagai noise impuls. Noise ini dapat disebabkan oleh gangguan tajam dan tiba-tiba pada sinyal gambar. Noise ini muncul sebagai piksel putih dan hitam yang jarang muncul

[14]. Noise Salt and Pepper hadir dalam berbagai bentuk dengan kepadatan yang berbeda. Dua variasi kepadatan dipertimbangkan dalam percobaan yaitu noise Salt and Pepper: kepadatan 0,05 dan noise Salt and Pepper: kepadatan 0,5.

(c) **Speckle Noise**: Speckle noise adalah noise yang muncul akibat pengaruh kondisi lingkungan terhadap sensor pencitraan selama akuisisi citra. Speckle noise paling banyak terdeteksi pada citra medis, citra Radar aktif, dan citra Synthetic Aperture Radar (SAR) [15]. Dua variasi varians dipertimbangkan dalam eksperimen yaitu Speckle noise: varians 0,04 dan Speckle noise: varians 0,4.

Setelah stego terkena serangan Noise ini, proses ekstraksi mengambil gambar rahasia dan melakukan koreksi kesalahan. PSNR dari gambar rahasia yang diambil ditentukan dan ditabulasi. PSNR dari semua gambar rahasia yang diambil ditentukan dan dibandingkan dengan Metode A. M. Molaei.

2. Cropping Attack: Citra stego dengan rentang nilai PSNR yang sama dikelompokkan dan diekspos ke dalam serangan pemotongan. Serangan pemotongan dilakukan berdasarkan rasio pemotongan dan wilayah pemotongan yang berbeda. Setiap citra stego dalam kelompok tertentu diekspos ke dalam rasio pemotongan dan wilayah pemotongan yang sama dengan citra stego lain dalam kelompok yang sama. Citra stego akan menjalani proses ekstraksi untuk mengambil citra rahasia dengan melak-

ukan koreksi kesalahan. Untuk setiap set atau kelompok, nilai PSNR dari citra rahasia yang diambil ditentukan dan ditabulasi serta dibandingkan dengan Metode A. M. Molaei.

3. Scratching Attack: Serangan goresan muncul dalam dua bentuk berbeda yaitu perbedaan panjang, lebar sama dan perbedaan lebar, panjang sama. Setelah setiap bentuk serangan goresan pada gambar stego dilakukan, koreksi kesalahan digunakan untuk mengambil gambar rahasia. PSNR dari gambar rahasia yang diambil ditentukan, ditabulasi dan dibandingkan dengan metode sebelumnya.

   Scratching single merupakan Scratching yang dilakukan dengan cara menggores satu garis pada gambar stego. Scratching multiple merupakan Scratching yang dilakukan dengan cara menggores beberapa garis pada gambar stego.

4. JPEG Compression Attack: Singkatan dari Joint Photographic Experts Group. JPEG adalah metode kompresi lossy yang umum digunakan untuk gambar digital, terutama untuk gambar yang dihasilkan oleh fotografi digital. JPEG biasanya mencapai kompresi 10:1 dengan sedikit kehilangan kualitas gambar yang terlihat.[16] Kompresi JPEG digunakan dalam sejumlah format berkas gambar. JPEG adalah format gambar yang paling umum digunakan oleh kamera digital dan format yang paling umum untuk menyimpan dan mengirimkan gambar fotografi di Web.

   Untuk melakukan percobaan, semua gambar stego yang disematkan dengan

ukuran berbeda dari pesan rahasia yang sama diekspos ke serangan JPG. Setelah mengekspos stego ke serangan kompresi JPEG, ekstraksi gambar rahasia dilakukan dengan menerapkan koreksi kesalahan untuk memulihkan gambar rahasia. PSNR dari gambar rahasia yang dipulihkan ditentukan, ditabulasi, dan dibandingkan dengan Metode A. M. Molaei.

5.4.3 Kapasitas

Pada bagian ini, kapasitas penyisipan dari metode yang diusulkan dan metode yang digunakan diperiksa. Rasio jumlah bit rahasia yang disematkan dalam gambar Cover terhadap jumlah piksel gambar Cover dievaluasi.

Hasil untuk setiap kelompok ditabulasi dan nilai PSNR ditentukan dan dibandingkan dengan Metode sebelumnya.

5.5 Metode Analisis dan Evaluasi

Penelitian ini akan merancang serangkaian eksperimen untuk mengevaluasi kinerja metode yang diusulkan:

5.5.1 Imperceptibility

Mean Square Error (MSE) adalah metrik yang digunakan untuk mengukur perbedaan antara dua gambar. MSE dihitung dengan rumus berikut:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - K(i,j)]^2 \tag{1}$$

Di mana:

- M dan N adalah dimensi gambar.
- $I(i, j)$ adalah nilai piksel pada posisi (i, j) dari gambar asli.
- $K(i, j)$ adalah nilai piksel pada posisi (i, j) dari gambar stego.

MSE yang lebih rendah menunjukkan bahwa gambar stego lebih mirip dengan gambar asli, yang berarti kualitas visual lebih baik.

Peak Signal-to-Noise Ratio (PSNR) adalah metrik yang digunakan untuk mengukur kualitas gambar stego dibandingkan dengan gambar asli. PSNR dihitung dengan rumus berikut:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{2}$$

Di mana:

- MAX_I adalah nilai maksimum intensitas gambar (misalnya, 255 untuk gambar 8-bit).
- MSE adalah Mean Square Error antara gambar asli dan gambar stego.

PSNR yang lebih tinggi menunjukkan bahwa gambar stego lebih mirip dengan gambar asli, yang berarti kualitas visual lebih baik.

5.5.2 Robbustnes

Ukuran kuantitatif yang akan ditentukan adalah PSNR dari citra rahasia yang dipulihkan. Hal ini ditentukan menggunakan persamaan 3 dan 4:

Kode Reed Muller RM (1, 4) telah diterapkan dalam metode ini karena kemampuan koreksi kesalahannya yang lebih tinggi. Angka 1 menunjukkan order/tingkat dari kode Reed-Muller Angka 4 menunjukkan dimensi atau jumlah variabel input.

Karena r = 1 dan m = 4, maka ukuran pesan k dihitung dengan:

$$k = \sum_{i=0}^{r} mi = 40 + 41 = 5bits \tag{3}$$

Dan kode berukuran N dihitung dengan:

$$N = 2^m = 16bits \tag{4}$$

$$ECC = \frac{numberofcorrectablebits}{totalbits} = \frac{3}{16} = 18.75\% \tag{5}$$

Setelah setiap skenario penyematan dilakukan seperti yang dijelaskan, gambar stego yang dihasilkan diekspos di bawah serangan berikut

1. Noise Attack
2. Cropping Attack
3. Scratching Attack
4. JPEG Compression Attack

5.5.3 Kapasitas

Kapasitas penyisipan, EC kemudian ditentukan oleh persamaan berikut:

$$EC = \frac{|S|}{H \cdot W}(bpp) \tag{6}$$

Di mana —S— adalah jumlah bit rahasia yang disematkan, H dan W masing-masing adalah tinggi dan panjang gambar. Muatan yang dihasilkan dinyatakan sebagai persentase. Untuk mengevaluasi kinerja metode yang diusulkan dalam hal kapasitas, dua skenario utama berikut dipertimbangkan:

1. Variasi Nilai Piksel: Gambar sampul dan gambar rahasia dianalisa berdasarkan histogram gambarnya.
    - Digunakan 40 gambar sampul skala berwarna berukuran 512x512 piksel.
    - Digunakan 40 gambar rahasia skala berwarna dengan berbagai ukuran.
    - Setiap gambar rahasia dengan distribusi tonal unik disematkan pada setiap gambar sampul dengan tonal yang unik pula.
2. Ukuran Secret Image: Untuk setiap gambar rahasia, dilakukan 33 variasi ukuran yang berbeda.
    - Setiap variasi ukuran disematkan pada gambar sampul yang berbeda.
    - Kapasitas yang dihasilkan dari setiap penyematan dicatat.

Hasil untuk setiap kelompok ditabulasi dan nilai PSNR ditentukan dan dibandingkan dengan Metode sebelumnya.

6 Work Plan and Time Schedule

Write a work plan along with the schedule for completion. The following is the example. You may adjust the activities and time schedule according to the problem.

Table 3: Activity Schedule

Activity		**SEMESTER**											
		1			**2**			**3**			**4**		
1	Literature study												
2	Problem identification												
3	Contribution formulation												
4	Hypothesis formulation												
5	Proposal												
6	Design Encoding Process												
7	Implement embedding schema												
8	Design decoding process												
9	Design sharing schema												
10	Imperceptibility testing												
11	Robbustnes testing												
12	Capacity testing												
13	Thesis draft												

Supervisor (I)'s Comments:

Comments about the title
Comments about the research method
Sign **Date:** (_____)

Supervisor (II)'s Comments:

Comments about the title
Comments about the research method
Sign **Date:** (_____)

References

[1] Ikwan Pujianto. “Uji Ketahanan Citra Digital Terhadap Manipulasi Robustness Pada Steganography”. In: *Jurnal Informatika dan Rekayasa Perangkat Lunak* 2.1 (2021), pp. 16–27.

[2] Siaulhak Siaulhak, Safwan Kasma et al. “Sistem Pengiriman File Menggunakan Steganografi Pengolahan Citra Digital Berbasis Matriks Laboratory”. In: *BANDWIDTH: Journal of Informatics and Computer Engineering* 1.2 (2023), pp. 75–81.

[3] Anggya ND Soetarmono. “Studi Mengenai Aplikasi Steganografi Camouflage”. In: *Teknika* 1.1 (2012), pp. 55–65.

[4] Muhammad Alfin Fikri and FX Ferdinandus. “Optimasi Teknik Steganografi Amelsbr Pada Empat Bit Terakhir Dengan Cover Image Berwarna”. In: *Antivirus: Jurnal Ilmiah Teknik Informatika* 16.1 (2022), pp. 25–38.

[5] Katandawa Alex Kingsley and Ari Moesriami Barmawi. “Improving Data Hiding Capacity in Code Based Steganography using Multiple Embedding.” In: ().

[6] Dwi Andika and Dedi Darwis. “Modifikasi Algoritma Gifshuffle Untuk Peningkatan Kualitas Citra Pada Steganografi”. In: *Jurnal Ilmiah Infrastruktur Teknologi Informasi* 1.2 (2020), pp. 19–23.

[7] Ajib Susanto and Ibnu Utomo Wahyu Mulyono. “Kombinasi LSB-RSA untuk Peningkatan Imperceptibility pada Kripto-Stegano Gambar RGB”. In: (2020).

[8] Muh Basri and Muhammad Fadhlil Gushari. “Penerapan Steganografi Gambar Berwarna pada Delapan Image Cover Menggunakan Metode LSB”. In: *Jurnal Sintaks Logika* 1.3 (2021), pp. 153–158.

[9] Ade Davy Wiranata and Rima Tamara Aldisa. “Aplikasi Steganografi Menggunakan Least Significant Bit (LSB) dengan Enkripsi Caesar Chipper dan Rivest Code 4 (RC4) Menggunakan Bahasa Pemrograman JAVA”. In: *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)* 5.3 (2021), pp. 277–281.

[10] Muhammad Oemar Abdillah, Ogie Ariansah Pane and Farhan Rusdy Asyhary Lubis. “Implementasi Keamanan Aset Informasi Steganografi Menggunakan Metode Least Significant Bit (LSB)”. In: *Jurnal Sains dan Teknologi (JSIT)* 3.1 (2023), pp. 40–46.

[11] Mamta Juneja and Parvinder S Sandhu. “An improved LSB based steganography technique for RGB color images”. In: *International journal of computer and communication engineering* 2.4 (2013), p. 513.

[12] AB Nasution, S Efendi and S Suwilo. “Image steganography in securing sound file using arithmetic coding algorithm, triple data encryption standard (3DES) and modified least significant bit (MLSB)”. In: *Journal of Physics: Conference Series*. Vol. 1007. 1. IOP Publishing. 2018, p. 012010.

[13] Ameen Mohammed Abd-Alsalam Selami and Ahmed Freidoon Fadhil. “A Study of the Effects of Gaussian Noise on Image”. In: ().

[14] Shakair Kaisar and Jubayer AI Mahmud. “Salt and Pepper Noise Detection and removal by Tolerance based selective Arithmetic Mean Filtering Technique for image restoration”. In: *International Journal of computer science and network security* 8.6 (2008), pp. 271–278.

[15] Mostafa Mansourpour, MA Rajabi and JAR Blais. “Effects and performance of speckle noise reduction filters on active radar and SAR images”. In: *Proc. Isprs*. Vol. 36. 1. 2006, W41.

[16] Mr Venugopal Reddy CH et al. “Medical image watermarking schemes against salt and pepper noise attack”. In: *International Journal of Bio-Science and Bio-Technology* 7.6 (2015), pp. 55–64.