LAPORAN UAS
KRIPTOGRAFI (TEMA: ROT13) MENGGUNAKAN STEGANOGRAFI LSB

Tugas ini disusun untuk memenuhi tugas mata kuliah Kriptografi

Dosen Pengampu:

Saiful Nur Budiman, S.Kom., M.Kom

DISUSUN OLEH:

KELOMPOK 2

ARVILANTI DEVANI	(22104410075)
RIZKI RAMADHAN	(22104410088)
NUR CINDY INTAN FANDERELLA	(22104410098)
HANIK HATUL HALIMAH	(22104410101)
WASI'ATUL JANNAH	(22104410121)
SAHRUL RAMADHAN	(22104414002)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS ISLAM BALITAR

2025

DAFTAR ISI

DAFTAR ISI 1
BAB I PENDAHULUAN 2
1.1 Latar Belakang 2
1.2 Rumusan Masalah 3
1.3 Tujuan Proyek 4
BAB II LANDASAN TEORI 5
2.1 Kriptografi 5
2.2 Kriptografi Klasik 5
2.3 Caesar Cipher 6
2.4 ROT13 6
2.5 Steganografi 7
2.6 Least Significant Bit (LSB) 7
2.7 Format Gambar BMP dan JPG 7
2.8 Python 8
BAB III PEMBAHASAN 9
3.1 Cara Kerja Algoritma ROT13 dalam Proses Enkripsi dan Dekripsi Teks 9
3.2 Algoritma Kriptografi Mengenai Steganografi LSB 10
3.3 Implementasi Algoritma ROT13 Menggunakan Bahasa Pemrograman Python 11
3.4 Penjelasan Source Code Aplikasi ROT13 & Outputnya 12
3.5 Petunjuk Menggunakan Aplikasi ROT13 yang Dikembangkan 15
3.6 Kelebihan & Kekurangan Pada Aplikasi 21
BAB IV PENUTUP 23
4.1 Kesimpulan 23
DAFTAR PUSTAKA 24
LAMPIRAN 27

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang sangat pesat telah mendorong peningkatan kebutuhan akan keamanan data dalam berbagai bidang kehidupan manusia. Setiap hari, jutaan data pribadi, transaksi digital, dan komunikasi elektronik berpindah melalui jaringan internet, sehingga risiko penyadapan, peretasan, serta penyalahgunaan informasi menjadi semakin tinggi. Dalam konteks ini, keamanan data menjadi aspek penting yang harus diperhatikan oleh setiap individu maupun organisasi.

Salah satu cara yang umum digunakan untuk melindungi data dari akses yang tidak sah adalah melalui kriptografi. Kriptografi merupakan salah satu teknik dasar dalam keamanan informasi yang mencakup proses-proses penting seperti enkripsi dan dekripsi. Enkripsi adalah prosedur mengubah pesan terbaca (*plaintext*) menjadi bentuk yang tidak dapat dibaca (*ciphertext*), sedangkan dekripsi adalah proses kebalikannya, yaitu mengembalikan ciphertext ke bentuk plainteks sehingga dapat dipahami oleh pihak yang berwenang (Hulu & Putri, 2022). Selain hanya untuk mengamankan pesan, penerapan kriptografi dalam konteks digital modern meliputi autentikasi, tanda tangan digital, penyimpanan data rahasia, serta mekanisme non-repudiasi, yaitu memastikan bahwa pengirim tidak dapat menampik telah mengirim pesan tersebut (Harahap & Salim, 2018).

Secara umum, kriptografi dibagi menjadi dua kategori besar, yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik menggunakan teknik sederhana berbasis penggantian (substitusi) atau pergeseran (transposisi) huruf pada teks, sedangkan kriptografi modern menggunakan operasi matematis dan kunci digital yang lebih kompleks untuk menjaga keamanan data (Dewi, 2024). Meskipun metode klasik dianggap kurang aman terhadap serangan modern, kriptografi jenis ini tetap relevan dalam konteks pembelajaran dasar mengenai konsep enkripsi dan dekripsi karena mudah dipahami serta dapat diimplementasikan menggunakan algoritma sederhana seperti Caesar Cipher atau Vigenère Cipher.

Salah satu algoritma kriptografi klasik yang terkenal dan mudah diterapkan adalah ROT13 (Rotate by 13 Places). Algoritma ini merupakan bentuk modifikasi dari Caesar Cipher, di mana setiap huruf digeser sejauh 13 posisi dalam alfabet. Keunggulan ROT13 terletak pada sifat simetrisnya, yaitu proses enkripsi dan dekripsi dapat dilakukan menggunakan metode yang sama tanpa memerlukan kunci tambahan (Milian & Sulistyo, 2023). Meskipun algoritma ini tidak memberikan tingkat keamanan yang tinggi, ROT13 masih sering digunakan dalam

konteks edukatif dan penyamaran teks ringan, seperti penyembunyian pesan sederhana, teka-teki, maupun pengaburan konten tertentu pada forum daring (Sitompul dkk., 2024).

Namun, dalam menghadapi ancaman keamanan yang semakin kompleks, penggunaan kriptografi saja terkadang belum cukup. Ciphertext hasil enkripsi yang terlihat sebagai kumpulan karakter acak sering kali justru menarik perhatian peretas untuk melakukan serangan brute force atau kriptanalisis. Oleh karena itu, muncul teknik perlindungan tambahan yang disebut steganografi. Berbeda dengan kriptografi yang menyembunyikan isi pesan, steganografi bertujuan untuk menyembunyikan keberadaan atau eksistensi dari pesan itu sendiri (Woźniak dkk., 2025).

Salah satu metode steganografi digital yang paling populer dan efektif dalam menyembunyikan data ke dalam media citra adalah LSB (*Least Significant Bit*). Teknik LSB bekerja dengan cara mengganti bit terakhir (bit yang paling tidak signifikan) pada data piksel sebuah gambar dengan bit pesan rahasia (Utomo dkk., 2023). Karena perubahan pada bit terakhir hanya memberikan dampak visual yang sangat minimal pada warna gambar, mata manusia tidak akan mampu membedakan antara gambar asli dengan gambar yang telah disisipi pesan (stego-image).

Penggabungan antara kriptografi ROT13 dan steganografi LSB menawarkan skema keamanan berlapis (hybrid security). Dalam skema ini, pesan terlebih dahulu diacak menggunakan ROT13 agar maknanya tidak langsung terbaca, kemudian disisipkan ke dalam bit-bit gambar menggunakan metode LSB agar keberadaannya tidak terdeteksi. Dengan pendekatan ini, risiko kebocoran data dapat diminimalisir; bahkan jika seorang peretas berhasil mendeteksi adanya pesan di dalam gambar, mereka masih harus menghadapi lapisan enkripsi untuk memahami isinya.

Oleh karena itu, pembahasan mengenai integrasi algoritma ROT13 dan metode steganografi LSB dalam laporan ini bertujuan untuk memberikan pemahaman mendasar mengenai penerapan keamanan data yang komprehensif. Melalui studi ini, diharapkan pembaca dapat memahami bagaimana teknik substitusi sederhana dan manipulasi bit digital dapat berkolaborasi dalam menciptakan sistem perlindungan informasi yang lebih efektif dan efisien.

1.2 Rumusan Masalah

1. Bagaimana cara kerja algoritma ROT13 dalam melakukan proses enkripsi dan dekripsi pada data teks?

2. Bagaimana cara kerja penyisipan data menggunakan metode Steganografi LSB pada media gambar berformat RGB?
3. Bagaimana cara mengimplementasikan gabungan algoritma ROT13 dan LSB menggunakan bahasa pemrograman Python secara manual tanpa library kriptografi otomatis?
4. Bagaimana hasil uji coba aplikasi dalam melakukan proses encrypt+embedding dan extraction+decrypt terhadap berbagai input teks?

1.3 Tujuan Proyek

1. Untuk memahami dan menjelaskan prinsip kerja algoritma ROT13 dalam proses enkripsi dan dekripsi teks.
2. Untuk memahami teknik penyembunyian data pada level bit menggunakan metode *Least Significant Bit* (LSB).
3. Untuk membangun aplikasi menggunakan Python yang mampu melakukan enkripsi teks sekaligus menyisipkannya ke dalam gambar secara manual, serta mampu melakukan proses ekstraksi dan dekripsi kembali ke bentuk aslinya.
4. Untuk menguji dan memastikan fungsionalitas aplikasi berjalan dengan pesan yang telah disisipkan dapat diambil dan didekripsi kembali menjadi teks asli secara akurat.

BAB II
LANDASAN TEORI

2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang berarti "penulisan rahasia", yaitu ilmu dan seni pengamanan informasi yang menjadi dasar penting dalam keamanan sistem komputer (Wardana, 2019). Kriptografi merupakan praktik perlindungan informasi baik yang diam (pada *hard drive*), bergerak (komunikasi elektronik), maupun yang sedang digunakan melalui penggunaan teknik matematis, algoritma kode, *hash*, dan tanda tangan digital (Rahman dkk., 2024; Wardana, 2019). Tujuan fundamental dari sistem kriptografi adalah menjaga kerahasiaan data (memastikan hanya pihak berwenang yang dapat mengaksesnya menggunakan kunci yang tepat), mempertahankan integritas data (mencegah manipulasi atau perubahan tidak sah seperti penyisipan atau penghapusan), dan menjamin autentikasi (mengidentifikasi kebenaran pihak-pihak yang berkomunikasi), serta menyediakan ketiadaan penyangkalan (Rahman dkk., 2024; Wardana, 2019). Proses inti dalam kriptografi melibatkan penggunaan algoritma kriptografi, yaitu sebuah fungsi matematika yang bergantung pada nilai kunci untuk mengubah data jelas (*plaintext*) menjadi data sandi (*ciphertext*) melalui enkripsi, dan mengembalikannya melalui dekripsi (Munawar dkk., 2023; Rahman dkk., 2024; Wardana, 2019). Kekuatan algoritma ini diukur dari seberapa sulitnya memperoleh nilai kunci yang benar dari seluruh ruang kunci yang memungkinkan (Munawar dkk., 2023).

2.2 Kriptografi Klasik

Kriptografi klasik adalah teknik kriptografi yang digunakan sebelum era komputerisasi (Sasono dkk., 2023). Algoritma ini umumnya beroperasi pada level karakter atau alfabet. Ciri utamanya adalah penggunaan pena dan kertas atau alat mekanis sederhana. Algoritmanya juga relatif mudah dipecahkan dengan teknik modern.

Metode kriptografi klasik secara umum dibagi menjadi dua (Sasono dkk., 2023):

1. *Cipher* Substitusi: Mengganti setiap karakter *plaintext* dengan karakter lain. Contoh: *Caesar Cipher*, *Vigenere Cipher*.
2. *Cipher* Transposisi: Mengubah urutan (permutasi) karakter dalam *plaintext*. Contoh: *Rail Fence Cipher*.

ROT13 termasuk dalam kategori *cipher* substitusi.

2.3 Caesar Cipher

Caesar Cipher merupakan salah satu metode algoritma yang dipakai untuk sistem kriptografi simetri. Metode ini sering dipakai sebelum adanya sistem kriptografi kunci publik (Hidayah dkk., 2023). Caesar Cipher termasuk jenis metode cipher substitusi yang membuat cipher melalui proses pergantian satu karakter yang diubah dengan karakter lain di beberapa jumlah digit sebelah kanan atau kirinya, sesuai arah pergeserannya (Dwi Putri dkk., 2019). Caesar Cipher memiliki kelebihan untuk menyamarkan suatu pesan yang bisa tersandi karena hanya pengirim dan penerima saja yang bisa mengetahui sandinya. Selain itu, Caesar Cipher juga memiliki kelemahan yaitu tidak dapat melakukan enkripsi ataupun dekripsi terhadap pesan yang tersusun dari beberapa kalimat dan juga rumus yang ditemukan (Hidayah dkk., 2023).

2.4 ROT13

ROT13 merupakan sebuah fungsi yang memakai kode kaisar dengan melakukan pergeseran k=13. ROT13 dirancang untuk keamanan pada sistem operasi UNIX yang banyak terdapat pada forum-forum *online* yang bermanfaat untuk melindungi isi artikel. Proses enkripsi dari ROT13 dengan menggeser maju karakter sebanyak 13 kali terhitung mulai 1 karakter yang ada didepannya, sedangkan untuk deskripsinya dengan menggeser mundur karakter 13 kali yang terhitung 1 karakter dibelakangnya. Pergeseran karakter tersebut dengan berdasar pada urutan karakter pada tabel ASCII.

Formula yang digunakan sebagai berikut (Hondro & Fau, 2018; Milian & Sulistyo, 2023)

$$C = (P + K) \text{ Mod } n$$

$$P = (C - K) \text{ Mod } n$$

Keterangan :

C = *Ciphertext*
P = *Plaintext*
K = *Key* (13)
Mod = Modulo
n = Jumlah karakter *plaintext*

2.5 Steganografi

Steganografi merupakan teknik keamanan informasi yang dilakukan dengan cara menyisipkan pesan ke dalam suatu media penampung (*cover*) sehingga keberadaan pesan tersebut tidak mudah dikenali oleh pihak lain secara kasat mata. Media yang umum digunakan dalam steganografi antara lain gambar, suara, dan teks, dengan media gambar menjadi yang paling banyak dimanfaatkan karena memiliki elemen digital seperti nilai piksel yang memungkinkan proses penyisipan pesan tanpa menimbulkan perubahan visual yang signifikan (Pradeka, 2018). Secara etimologis, istilah steganografi berasal dari bahasa Yunani, yaitu steganos yang berarti tersembunyi dan graphein yang berarti tulisan. Oleh karena itu, steganografi dapat didefinisikan sebagai seni dan teknik menyembunyikan pesan ke dalam data lain sedemikian rupa sehingga media penampung sebelum dan sesudah proses penyisipan tampak hampir sama dan tidak menimbulkan kecurigaan (Syawal dkk., 2016).

2.6 Least Significant Bit (LSB)

Metode *Least Significant Bit* (LSB) merupakan teknik steganografi pada domain spasial yang menyisipkan pesan dengan cara mengganti bit paling tidak signifikan pada nilai piksel citra digital. Perubahan pada bit ini hanya menyebabkan perbedaan nilai piksel yang sangat kecil sehingga tidak menimbulkan perubahan visual yang signifikan dan sulit dikenali oleh penglihatan manusia. Prinsip kerja metode LSB dilakukan dengan mengonversi pesan ke dalam bentuk biner, kemudian menyisipkan bit pesan tersebut ke bit terakhir dari setiap piksel citra. Karena bit LSB memiliki bobot nilai terendah, proses penyisipan tidak mengganggu kualitas citra secara nyata, sehingga metode ini banyak digunakan dalam steganografi citra digital (Laksono dkk., 2024). Namun demikian, metode LSB memiliki keterbatasan pada aspek keamanan dan ketahanan terhadap manipulasi citra. Perubahan piksel akibat penyisipan pesan masih dapat dideteksi oleh sistem komputer melalui analisis statistik atau steganalisis, terutama jika jumlah pesan yang disisipkan cukup besar (Hakim & Sholikhan, 2024).

2.7 Format Gambar BMP dan JPG

Format gambar BMP (*Bitmap Image*) merupakan format penyimpanan citra digital tanpa kompresi (*lossless*) yang menyimpan informasi warna secara eksplisit pada setiap pikselnya (Wildan & Ashari, 2024). Karena sifatnya yang tidak merusak data, format BMP mampu menjaga integritas bit pesan yang disisipkan melalui metode LSB dengan kualitas visual yang sangat tinggi, dibuktikan dengan nilai *Peak Signal-to-Noise Ratio* (PSNR) yang mencapai 70,84 dB (Wildan & Ashari, 2024). Hal ini menjadikan BMP sebagai media penampung yang

paling direkomendasikan untuk steganografi karena tidak adanya risiko kehilangan data akibat algoritma kompresi (Jum'ah & Arifin, 2025).

Di sisi lain, format JPG atau JPEG (*Joint Photographic Experts Group*) menggunakan algoritma kompresi *lossy* untuk meminimalkan ukuran file, namun proses ini secara inheren merusak bit-bit rahasia yang telah disisipkan pada domain spasial (Wildan & Ashari, 2024). Hasil pengujian menunjukkan bahwa penggunaan LSB pada format JPEG menghasilkan kualitas citra yang lebih rendah dengan nilai PSNR sebesar 56,60 dB yang mengindikasikan adanya degradasi data akibat kompresi (Wildan & Ashari, 2024). Meskipun metode LSB dapat diterapkan pada berbagai resolusi gambar, format BMP tetap terbukti lebih unggul daripada JPEG dalam mempertahankan kekokohan (*robustness*) pesan rahasia terhadap gangguan manipulasi file (A/L Selvarajan & Yusoff, 2024).

2.8 Python

Bahasa pemrograman Python merupakan bahasa pemrograman yang populer dalam bidang analisis data. Hal tersebut karena Python mudah untuk dipelajari dan digunakan di semua kalangan usia. Selain itu, bahasa pemrograman Python memiliki Library yang bervariasi yang memiliki kegunaannya masing-masing dan dapat digunakan oleh siapa saja di berbagai sistem operasi atau dengan kata lain bersifat *open source*. Beberapa contoh Library Python antara lain adalah NumPy, Pandas, Matplotlib, dan Scikit-learn yang masing-masing berguna untuk analisis data, pemodelan statistik, visualisasi data, dan machine learning. Tidak hanya itu, Python juga mudah diintegrasikan dengan teknologi lain, seperti database, big datatools, frameworkweb, dan lain sebagainya yang berhubungan dengan analisis data dengan tujuan untuk mengakses dan mengelola data dari berbagai sumber (Angelina M. T. I. Sambi Ua dkk., 2023).

BAB III
PEMBAHASAN

3.1 Cara Kerja Algoritma ROT13 dalam Proses Enkripsi dan Dekripsi Teks

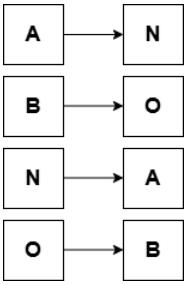
Algoritma ROT13 (*Rotate by 13 places*) merupakan salah satu bentuk sederhana dari Caesar Cipher, yaitu algoritma substitusi yang menggantikan setiap huruf dengan huruf lain yang berjarak 13 posisi setelahnya dalam urutan alfabet. Proses ini bekerja pada huruf-huruf dalam alfabet Latin (A–Z), dan karena total huruf alfabet adalah 26, maka jika dilakukan pergeseran sebanyak 13 posisi dua kali, huruf tersebut akan kembali ke bentuk semula. Hal ini membuat ROT13 bersifat simetris, yaitu proses enkripsi dan dekripsi menggunakan algoritma yang sama sehingga tidak memerlukan kunci tambahan.

Langkah-langkah kerja algoritma ROT13 yang diimplementasikan dalam aplikasi ini adalah sebagai berikut:

1. Sistem membaca setiap karakter masukan dari pesan teks (*plaintext*) secara berurutan
2. Setiap huruf dikonversi menjadi nilai numerik berdasarkan posisi indeksnya (A/a=0, B/b=1, ..., Z/z=25).
3. Nilai posisi digeser sebanyak 13 langkah dengan operasi modulo 26:

   posisi baru = (posisi lama + 13) mod 26
4. Hasil operasi tersebut dikonversi kembali menjadi karakter alfabet dengan tetap mempertahankan format huruf besar atau kecil sesuai input aslinya.
5. Karakter non-huruf seperti spasi, angka, dan simbol tidak mengalami perubahan posisi atau nilai.

Keunggulan utama dari penggunaan pergeseran 13 adalah kemudahan dalam proses pembalikan data.



Gambar 1 operasi modulo

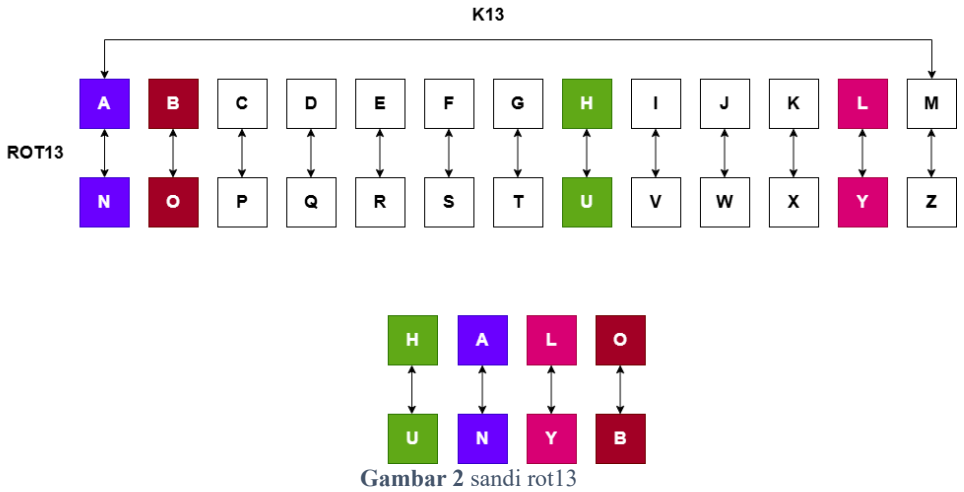
Misalnya, huruf "N" (posisi 13) jika digeser 13 langkah akan menjadi (13 + 13) mod 26 = 0 yang merujuk kembali ke huruf "A". Proses dekripsi dilakukan dengan algoritma yang

identik karena pergeseran total 26 langkah (13 + 13) akan mengembalikan karakter ke posisi awal. Sebagai contoh:

Teks asli : HALO

Enkripsi : UNYB

Dekripsi : HALO



Gambar 2 sandi rot13

3.2 Algoritma Kriptografi Mengenai Steganografi LSB

Metode *Least Significant Bit* (LSB) yang diimplementasikan dalam aplikasi ini bekerja dengan memanfaatkan bit terakhir pada setiap komponen warna piksel gambar (merah, hijau, dan biru) untuk menyimpan data biner dari pesan rahasia. Perubahan pada bit terakhir ini dipilih karena memiliki pengaruh nilai yang paling kecil terhadap warna keseluruhan, sehingga perbedaan antara gambar asli dan gambar stego tidak dapat dideteksi secara visual oleh mata manusia.

Proses kerja algoritma steganografi LSB dalam aplikasi ini dibagi menjadi dua tahap utama, yaitu proses penyisipan (*embedding*) dan proses ekstraksi.

a. Tahap Penyisipan (*Embedding*)

Langkah-langkah yang dilakukan sistem untuk menyembunyikan pesan adalah sebagai berikut:

1. Pesan teks yang telah dienkripsi dengan ROT13 ditambahkan dengan penanda khusus (*signature*) STEGO# di awal pesan dan penanda akhir ||STOP|| di akhir pesan. Penanda ini berfungsi untuk validasi saat proses ekstraksi dilakukan.
2. Seluruh rangkaian pesan (penanda + *ciphertext* + penanda akhir) dikonversi menjadi deretan bit biner, di mana setiap satu karakter diwakili oleh 8 bit.

3. Aplikasi membaca data piksel gambar secara berurutan mulai dari koordinat (0,0). Pada setiap piksel, terdapat tiga saluran warna yaitu *Red* (R), *Green* (G), dan *Blue* (B).
4. Setiap satu bit pesan disisipkan ke dalam bit terakhir (LSB) dari saluran R, kemudian bit berikutnya ke saluran G, dan bit selanjutnya ke saluran B. Proses ini terus berulang piksel demi piksel hingga seluruh bit pesan tersimpan.
5. Setelah seluruh bit tersisip, gambar disimpan dalam format BMP (*Bitmap*). Penggunaan format BMP sangat krusial karena bersifat *lossless*, sehingga data biner yang telah disisipkan pada LSB tidak akan rusak atau berubah akibat kompresi file.

b. Tahap Ekstraksi

Langkah-langkah yang dilakukan sistem untuk mengambil kembali pesan adalah sebagai berikut:

1. Sistem membaca bit terakhir (LSB) dari setiap saluran warna RGB pada setiap piksel gambar secara berurutan.
2. Setiap 8 bit yang terkumpul dikonversi kembali menjadi satu karakter teks.
3. Proses pembacaan bit akan terus berlangsung hingga sistem menemukan rangkaian karakter ||STOP||. Hal ini memastikan bahwa sistem tidak membaca data sampah (*noise*) dari piksel gambar yang tidak mengandung pesan.
4. Setelah pesan didapat, sistem memeriksa apakah pesan diawali dengan STEGO#. Jika benar, maka bagian di antara kedua penanda tersebut diambil sebagai *ciphertext* yang valid untuk didekripsi kembali menjadi pesan asli.

3.3 Implementasi Algoritma ROT13 Menggunakan Bahasa Pemrograman Python

Algoritma ROT13 diimplementasikan dalam aplikasi ini menggunakan bahasa pemrograman Python untuk melakukan proses enkripsi dan dekripsi teks secara efisien. Sebagai algoritma simetris, fungsi yang sama digunakan untuk kedua proses tersebut karena pergeseran ganda sebesar 13 posisi akan mengembalikan karakter ke bentuk aslinya secara otomatis.

Dalam implementasi teknisnya, aplikasi melakukan pemrosesan karakter demi karakter dari pesan masukan dengan ketentuan logika sebagai berikut:

- Sistem melakukan iterasi pada setiap karakter dalam pesan untuk membedakan antara huruf alfabet dan karakter non-huruf.

- Jika karakter merupakan huruf kecil, sistem menghitung posisi baru dengan mengonversi karakter ke nilai ASCII, menggesernya sebanyak 13 posisi dengan operasi modulo 26, dan mengembalikannya ke bentuk karakter alfabet kecil.
- Jika karakter merupakan huruf besar, sistem melakukan proses yang sama dengan basis nilai ASCII untuk huruf kapital agar format penulisan asli tetap terjaga.
- Angka, spasi, dan simbol tidak mengalami modifikasi dan langsung digabungkan kembali ke dalam rangkaian hasil proses.

Hasil akhir dari proses enkripsi ini adalah *ciphertext* yang selanjutnya digunakan sebagai *input* data utama pada tahap penyisipan steganografi LSB. Pada tahap pembalikan, *ciphertext* yang berhasil diekstraksi dari bit-bit gambar stego akan diproses kembali melalui logika Python yang sama untuk mendapatkan kembali pesan asli atau *plaintext*.

3.4 Penjelasan Source Code Aplikasi ROT13 & Outputnya

1. Fungsi ROT13

```python
def rot13(s):
    out=[]
    for c in s:
        # Shift huruf kecil 13 posisi
        if 'a'<=c<='z': out.append(chr((ord(c)-97+13)%26+97))
        # Shift huruf besar 13 posisi
        elif 'A'<=c<='Z': out.append(chr((ord(c)-65+13)%26+65))
        # Karakter non-huruf tidak berubah
        else: out.append(c)
    return ''.join(out)
```

a. Fungsi melakukan pergeseran huruf alfabet sebanyak 13 posisi.
b. Mendukung huruf besar (A–Z) dan kecil (a–z).
c. Karakter non-huruf tidak diubah.
d. Bersifat involutif: ROT13(ROT13(x)) = x.

Contoh Output :

Input	**Cipher (ROT13)**	**Plain (ROT13 lagi)**
Hello	Uryyb	Hello
World	Jbeyq	World
Test123	Grfg123	Test123

2. Proses Embed ROT13 ke LSB

```python
def do_embed(self):
        if not self.file:
            return messagebox.showerror("Error","Belum pilih gambar")
        msg = self.txt_in.get("1.0","end").strip()
        if not msg:
            return messagebox.showerror("Error","Pesan kosong")
        try:
            stego = embed_lsb(self.file, rot13(msg))
            messagebox.showinfo("Sukses",f"Stego disimpan: {os.path.basename(stego)}")
            self.txt_in.delete("1.0","end")
            self.reset()
        except Exception as e:
            messagebox.showerror("Gagal",str(e))
```

a. msg (plaintext) diambil dari input user pada GUI.

b. rot13() mengubah plaintext menjadi ciphertext.

c. Ciphertext diteruskan ke fungsi embed_lsb() untuk disisipkan ke citra.

Output yang dihasilkan:

a. File baru: nama_asli_stego.bmp

b. Isi gambar tampak normal (imperceptible)

c. Payload berada pada LSB piksel RGB.

3. Proses Extract & Dekripsi ROT13

```python
def do_extract(self):
        if not self.file:
            return messagebox.showerror("Error","Belum pilih stego")
        cipher = extract_lsb(self.file)
        self.txt_out.delete("1.0","end")
        if cipher is None:
            self.txt_out.insert("end","Tidak ada pesan tersembunyi")
        else:
            plain = rot13(cipher)
            self.txt_out.insert("end", f"Ciphertext: {cipher}\nPlaintext: {plain}")
```

a. extract_lsb() mengambil ciphertext dari gambar.

b. ROT13 diterapkan kembali untuk mengembalikan ciphertext menjadi plaintext.

c. Hasil ditampilkan pada textbox hasil GUI.

Contoh Output :

Ciphertext: Uryyb Jbeyq

Plaintext: Hello World

4. Mekanisme Validasi Payload

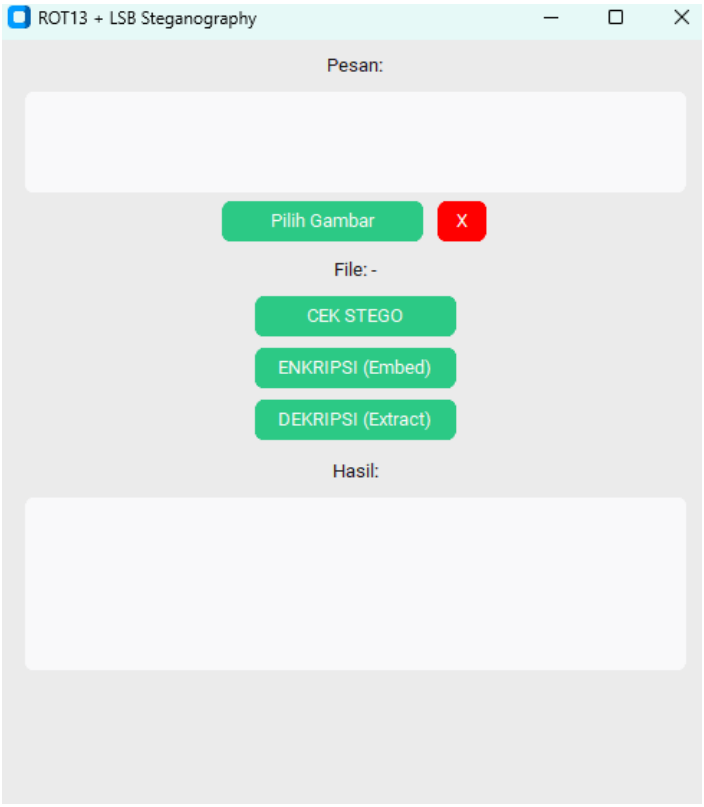
a. SIGN → memastikan gambar adalah stego image.

b. STOP → menentukan batas akhir pesan saat ekstraksi.

c. Mencegah false extract atau pembacaan noise.

5. Antarmuka GUI (CustomTkinter)

Gambar di bawah menunjukkan tampilan antarmuka utama aplikasi ROT13 + LSB Steganography yang dibangun menggunakan library CustomTkinter.



Komponen utama:

a. Textbox Pesan

Digunakan untuk memasukkan plaintext sebelum dilakukan proses enkripsi dan embed.

b. Button “Pilih Gambar”

Memungkinkan pengguna memilih file input dengan format .bmp atau .jpg.

c. Button “X” (Reset)

Menghapus file yang dipilih dan membersihkan hasil proses.

d. Label “File: -”

Menampilkan nama file gambar yang telah dipilih.

e. Button “CEK STEGO”

Berfungsi untuk mendeteksi apakah gambar mengandung payload ciphertext pada LSB.

f. Button “ENKRIPSI (Embed)”

Menjalankan proses ROT13 kemudian embedding ciphertext ke dalam gambar.

g. Button “DEKRIPSI (Extract)”

Menjalankan ekstraksi ciphertext dari gambar kemudian mengembalikan plaintext dengan ROT13.

h. Textbox Hasil

Menampilkan output proses baik berupa ciphertext maupun plaintext.

Contoh Hasil Output GUI

a. Tanpa Stego

Status: Tidak ada pesan tersembunyi

b. Setelah Embed

Sukses: Stego disimpan: sample_stego.bmp

c. Setelah Extract

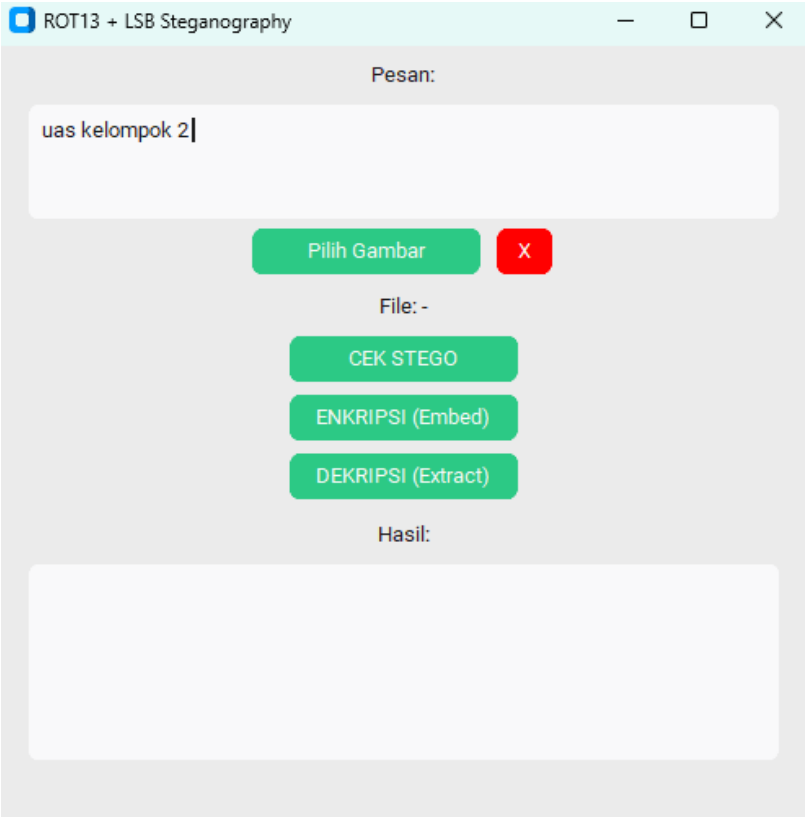
Ciphertext: Uryyb Jbeyq

Plaintext: Hello World

3.5 Petunjuk Menggunakan Aplikasi ROT13 yang Dikembangkan

1. Langkah Pertama

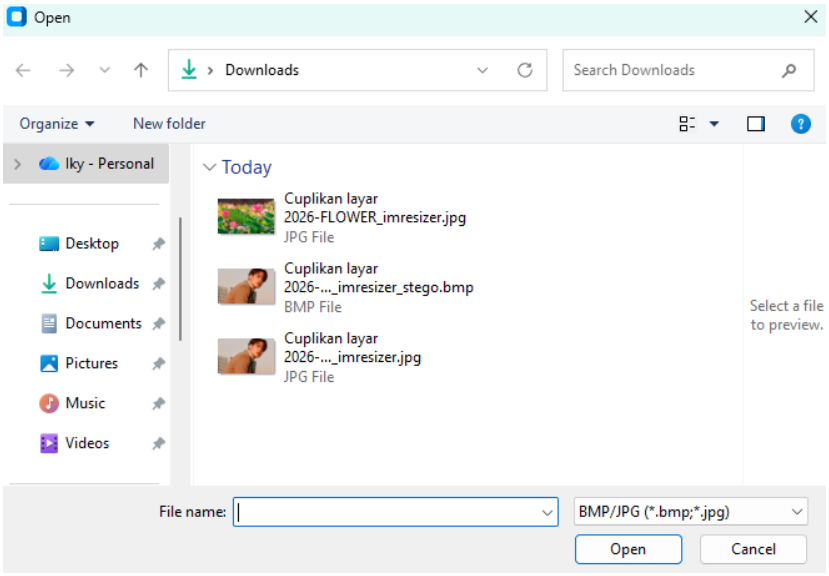
Langkah pertama dalam penggunaan aplikasi ROT13 + LSB Steganography adalah menjalankan aplikasi hingga tampil antarmuka utama (GUI) seperti yang ditunjukkan pada gambar. Pada tahap ini, pengguna akan melihat beberapa komponen utama, yaitu textbox pesan, tombol pemilihan gambar, serta tombol proses enkripsi dan dekripsi.



Pengguna terlebih dahulu memasukkan pesan teks (plaintext) yang ingin diamankan ke dalam textbox “Pesan”. Pesan ini merupakan data awal yang nantinya akan diproses menggunakan algoritma ROT13 sebelum disisipkan ke dalam media gambar. Contoh pesan yang dimasukkan pada tahap ini adalah “uas kelompok 2”.

2. Langkah Kedua (Memilih Gambar Cover)

Pada langkah kedua, pengguna memilih file gambar yang akan digunakan sebagai media penampung (cover image) untuk proses steganografi. Setelah tombol “Pilih Gambar” pada antarmuka utama ditekan, sistem akan menampilkan jendela dialog pemilihan file seperti yang ditunjukkan pada gambar.

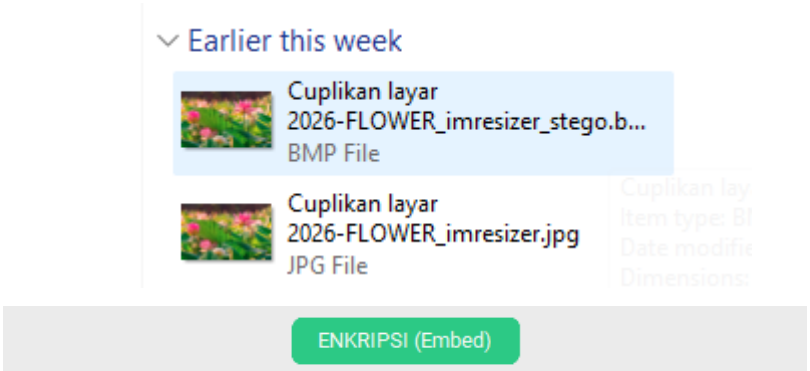


Aplikasi mendukung pemilihan file gambar dengan format BMP dan JPG, yang ditandai melalui filter ekstensi file (*.bmp; *.jpg) pada bagian bawah jendela dialog. Pengguna kemudian menelusuri direktori penyimpanan, misalnya folder Downloads, dan memilih salah satu file gambar yang diinginkan.

Setelah file gambar dipilih dan tombol “Open” ditekan, nama file gambar akan ditampilkan pada label “File:” di antarmuka aplikasi. Hal ini menandakan bahwa gambar telah berhasil dimuat ke dalam sistem dan siap digunakan untuk proses selanjutnya, yaitu enkripsi pesan dan penyisipan (embedding) menggunakan algoritma ROT13 dan metode Least Significant Bit (LSB).

3. Langkah Ketiga (Enkripsi dan Penyisipan Pesan)

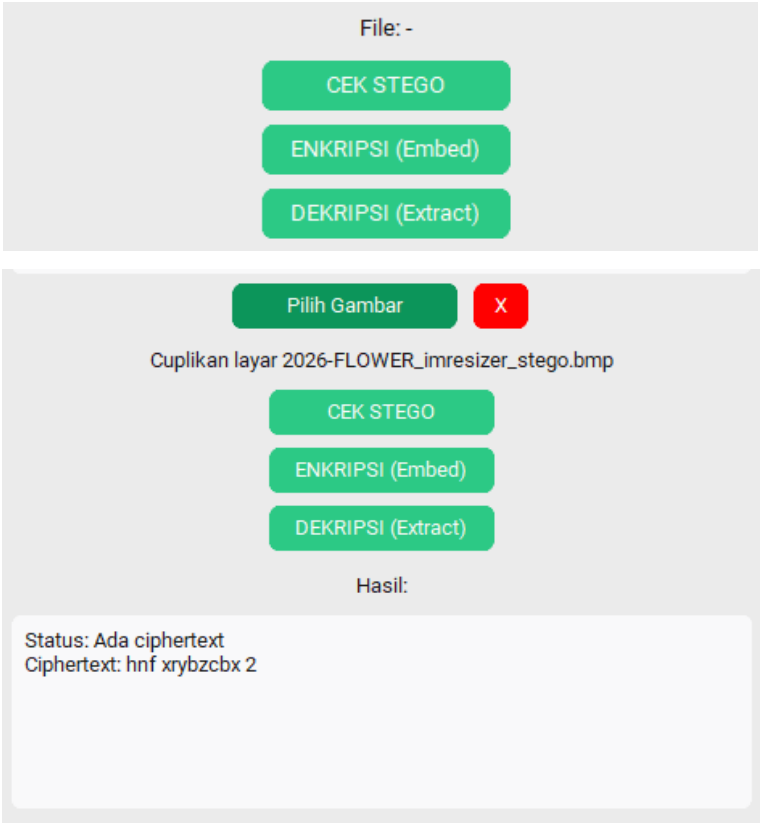
Pada langkah ketiga, pengguna melakukan proses enkripsi dan penyisipan pesan (embedding) dengan menekan tombol “ENKRIPSI (Embed)”. Pada tahap ini, pesan teks yang telah dimasukkan akan terlebih dahulu dienkripsi menggunakan algoritma ROT13, kemudian hasil enkripsi tersebut disisipkan ke dalam gambar menggunakan metode Least Significant Bit (LSB).



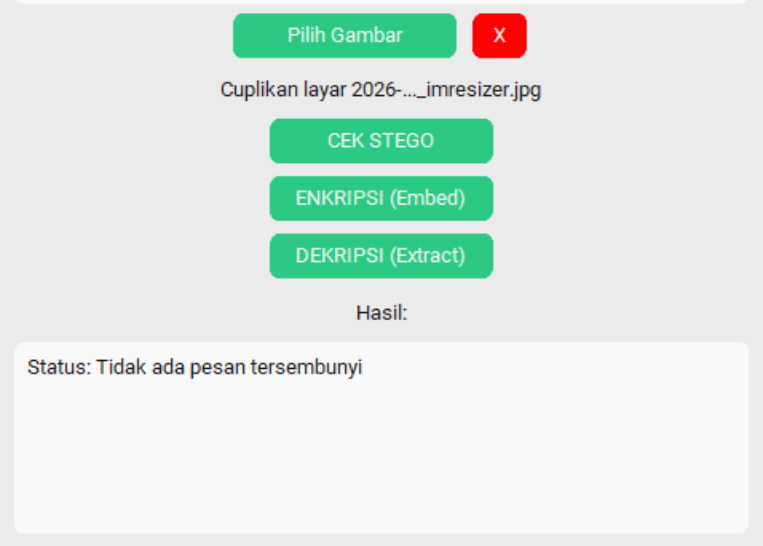
Setelah proses berhasil, aplikasi secara otomatis menyimpan file hasil steganografi dengan nama tambahan _stego pada direktori yang sama dengan gambar asli. Proses ini menandakan bahwa pesan telah berhasil disembunyikan ke dalam citra digital.

4. Langkah Keempat Cek Stego

Pada langkah keempat, pengguna melakukan pengecekan keberadaan pesan tersembunyi (stego detection) pada sebuah file gambar dengan menekan tombol “CEK STEGO” pada antarmuka utama aplikasi.



Setelah tombol ditekan, sistem akan menampilkan jendela dialog pemilihan file. Pengguna memilih gambar yang diduga mengandung pesan tersembunyi, biasanya ditandai dengan nama file yang mengandung kata stego, misalnya FLOWER_imresizer_stego.bmp. Aplikasi mendukung format file BMP dan JPG sesuai dengan filter ekstensi yang tersedia.



Aplikasi kemudian melakukan proses validasi dengan membaca penanda khusus (signature) yang tersimpan pada bit LSB gambar. Jika gambar tidak mengandung pesan tersembunyi, sistem akan menampilkan informasi "Status: Tidak ada pesan tersembunyi". Sebaliknya, jika gambar terdeteksi sebagai stego-image, pengguna dapat melanjutkan ke tahap berikutnya.

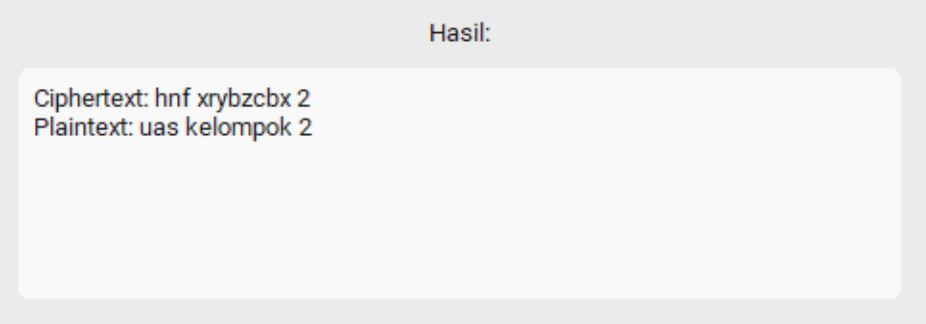
5. Langkah Kelima Deskripsi File (Dekripsi dan Menampilkan Hasil)

Pada langkah kelima, setelah gambar berhasil terdeteksi sebagai stego-image, pengguna dapat melanjutkan proses dekripsi pesan tersembunyi dengan menekan tombol "DEKRIPSI (Extract)" pada antarmuka aplikasi.

Aplikasi akan membaca data tersembunyi yang terdapat pada bit Least Significant Bit (LSB) dari file gambar stego, kemudian mengekstrak pesan yang telah disisipkan sebelumnya. Pesan hasil ekstraksi masih berada dalam bentuk ciphertext, karena sebelumnya telah melalui proses enkripsi menggunakan algoritma ROT13.

Selanjutnya, sistem secara otomatis melakukan proses dekripsi ROT13 dengan menggeser kembali setiap karakter sebanyak 13 posisi untuk mengembalikan pesan ke bentuk plaintext. Hasil proses ini ditampilkan pada kolom Hasil, yang memuat dua informasi, yaitu:

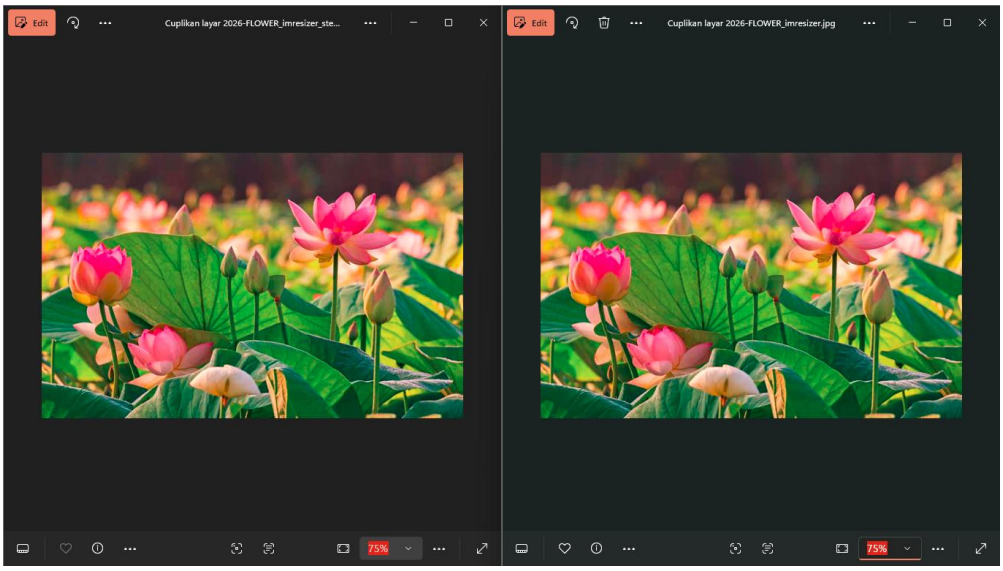
- Ciphertext: hasil enkripsi ROT13 dari pesan asli,
- Plaintext: pesan asli yang berhasil dikembalikan.



Berdasarkan hasil pengujian, ciphertext “nhf xrybcxb 2” berhasil didekripsi kembali menjadi plaintext “uas kelompok 2”. Hal ini menunjukkan bahwa proses ekstraksi steganografi dan dekripsi ROT13 berjalan dengan baik dan sesuai dengan perancangan sistem.

6. Analisis Visual Hasil Steganografi

Secara visual, tidak terdapat perbedaan yang signifikan antara gambar asli dan gambar stego. Perubahan nilai piksel hanya terjadi pada bit paling rendah (LSB) sehingga tidak memengaruhi kualitas visual gambar secara kasat mata.



Pengujian ini menunjukkan bahwa pesan yang telah dienkripsi menggunakan algoritma ROT13 dapat disisipkan ke dalam media gambar dengan baik, serta tidak menimbulkan distorsi

visual yang dapat dikenali oleh pengguna biasa. Dengan demikian, gambar stego tetap terlihat alami dan tidak mencurigakan, namun tetap mampu menyimpan informasi rahasia.

3.6 Kelebihan & Kekurangan Pada Aplikasi

1. Kelebihan

a. Antarmuka Sederhana dan Mudah Digunakan

Aplikasi memiliki tampilan GUI yang sederhana dan intuitif dengan tombol utama seperti Pilih Gambar, CEK STEGO, ENKRIPSI (Embed), dan DEKRIPSI (Extract). Berdasarkan hasil screenshot, pengguna dapat memahami alur penggunaan aplikasi tanpa memerlukan dokumentasi tambahan.

b. Implementasi Pengamanan Berlapis

Aplikasi menerapkan dua lapisan keamanan, yaitu:

- Kriptografi ROT13 untuk mengenkripsi pesan teks, dan
- Steganografi LSB untuk menyembunyikan pesan ke dalam citra digital.

Pendekatan ini meningkatkan tingkat keamanan karena pesan tidak hanya disamarkan, tetapi juga disembunyikan di dalam media gambar.

c. Hasil Visual Stego Tidak Berubah Signifikan

Berdasarkan perbandingan citra asli dan citra stego pada screenshot, tidak terlihat perbedaan visual yang signifikan. Hal ini membuktikan bahwa metode LSB berhasil menjaga kualitas citra dengan hanya memodifikasi bit paling rendah pada nilai RGB.

d. Penanganan Error Dasar Sudah Tersedia

Aplikasi sudah dilengkapi dengan validasi dan notifikasi kesalahan, seperti:

- Peringatan jika gambar belum dipilih,
- Peringatan jika pesan kosong,
- Validasi ukuran minimum gambar (128×128 piksel), dan
- Validasi kapasitas penyisipan pesan.

2. Kekurangan

a. Algoritma ROT13 Kurang Aman untuk Data Sensitif

ROT13 merupakan algoritma kriptografi klasik yang sangat sederhana dan mudah dipecahkan. Siapa pun yang mengetahui metode ROT13 dapat dengan mudah mengembalikan ciphertext ke plaintext tanpa kunci tambahan.

b. Tidak Menggunakan Kunci (Key-Based Encryption)

Aplikasi tidak menggunakan kunci rahasia (secret key) pada proses enkripsi maupun steganografi. Akibatnya, tingkat keamanan masih terbatas dan belum memenuhi standar kriptografi modern.

c. Rentan terhadap Manipulasi Gambar

Metode LSB sangat sensitif terhadap:

- kompresi ulang,
- resize,
- atau editing warna.

Jika gambar stego mengalami proses tersebut, pesan tersembunyi berpotensi rusak atau hilang.

d. Format Output Terbatas pada BMP

Hasil stego selalu disimpan dalam format BMP, yang memiliki ukuran file relatif besar. Aplikasi belum mendukung format lain seperti PNG secara penuh atau JPEG untuk output stego.

e. Tidak Ada Indikator Kapasitas Maksimum Pesan

Aplikasi belum menampilkan informasi kapasitas maksimum pesan yang dapat disisipkan sebelum proses embedding dilakukan. Pengguna baru mengetahui batasan saat muncul pesan error.

f. Tampilan GUI Masih Bersifat Dasar

Meskipun fungsional, tampilan GUI masih sederhana dan belum menyediakan fitur seperti:

- preview gambar sebelum dan sesudah embed,
- indikator proses (loading/progress bar),
- atau statistik penyisipan (jumlah karakter, kapasitas terpakai).

BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan seluruh tahapan analisis, perancangan, implementasi, dan pengujian yang telah dilakukan pada proyek ini, dapat disimpulkan bahwa penggabungan algoritma kriptografi ROT13 dengan metode steganografi Least Significant Bit (LSB) mampu membentuk sebuah sistem pengamanan data berlapis yang sederhana namun fungsional. Algoritma ROT13, sebagai salah satu cipher substitusi klasik turunan dari Caesar Cipher dengan pergeseran tetap 13 karakter, memiliki sifat simetris sehingga proses enkripsi dan dekripsi dapat dilakukan menggunakan fungsi yang sama. Hal ini mempermudah implementasi serta memberikan pemahaman konseptual yang kuat mengenai prinsip dasar kriptografi.

Implementasi algoritma ROT13 dan metode LSB menggunakan bahasa pemrograman Python berhasil dilakukan tanpa memanfaatkan library kriptografi otomatis. Proses enkripsi dilakukan dengan menggeser karakter alfabet menggunakan operasi modulo, sedangkan proses steganografi dilakukan dengan menyisipkan bit pesan terenkripsi ke dalam bit paling tidak signifikan pada kanal warna RGB citra digital. Penggunaan format gambar BMP terbukti efektif karena bersifat lossless sehingga mampu mempertahankan integritas data pesan yang disisipkan.

Hasil pengujian menunjukkan bahwa aplikasi mampu melakukan proses enkripsi, penyisipan, ekstraksi, dan dekripsi pesan secara konsisten dan akurat. Pesan teks yang telah dienkripsi menggunakan ROT13 dapat disembunyikan ke dalam citra digital tanpa menimbulkan perubahan visual yang signifikan, serta dapat diekstraksi kembali dan dikembalikan ke bentuk plaintext semula. Selain itu, mekanisme validasi menggunakan penanda khusus (signature dan stop marker) berhasil mencegah kesalahan pembacaan data dan meningkatkan keandalan proses ekstraksi.

Meskipun demikian, sistem yang dibangun masih memiliki keterbatasan, terutama dari sisi keamanan kriptografi karena ROT13 bukanlah algoritma yang dirancang untuk melindungi data sensitif, serta metode LSB yang rentan terhadap manipulasi citra seperti kompresi dan pengubahan ukuran. Oleh karena itu, aplikasi ini lebih tepat digunakan sebagai media pembelajaran dan demonstrasi konsep kriptografi dan steganografi dasar. Secara keseluruhan, proyek ini telah berhasil mencapai tujuan pembelajaran dengan memberikan pemahaman komprehensif mengenai penerapan kriptografi klasik, steganografi citra, serta integrasi keduanya dalam sebuah aplikasi keamanan data sederhana berbasis Python.

DAFTAR PUSTAKA

A/L Selvarajan, K., & Yusoff, Y. (2024). Steganography Algorithms in Computed Tomography (CT) Scan Images. *International Journal of Innovative Computing*, *14*(2), 65–71. https://doi.org/10.11113/ijic.v14n2.472

Angelina M. T. I. Sambi Ua, Diandra Lestriani H, Elizabeth Sonia Kristanty Marpaung, Jesslyn Ong, Michelle Savinka, Putri Nurhaliza, & Rahmi Yulia Ningsih. (2023). Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru. *Jurnal Publikasi Teknik Informatika*, *2*(2), 88–99. https://doi.org/10.55606/jupti.v2i2.1742

Dewi, S. K. (2024). Perbandingan Cryptography Klasik Vigenere Cipher Dengan Cryptography Modern RC4 Dalam Tingkat Keamanan Jaringan Komputer. *JoMMiT : Jurnal Multi Media dan IT*, *8*(2), 130–137. https://doi.org/10.46961/jommit.v8i2.1302

Dwi Putri, Y., Rosihan, R., & Lutfi, S. (2019). Penerapan Kriptografi Caesar Cipher Pada Fitur Chatting Sistem Informasi Freelance. *JIKO (Jurnal Informatika dan Komputer)*, *2*(2), 87–94. https://doi.org/10.33387/jiko.v2i2.1319

Hakim, F. N., & Sholikhan, M. (2024). *Enhancing Data Security through Digital Image Steganography : An Implementation of the Two Least Significant Bits (2LSB) Method*. *02*(November), 222–235.

Harahap, A. R., & Salim, T. A. (2018). *Sistem Kriptografi pada Pengamanan Autentikasi Dokumen Elektronik: Systematic Literature Review*. *16*(2), 203–220.

Hidayah, V. M., Mulyana, D. I., & Bachtiar, Y. (2023). Algoritma Caesar Cipher atau Vigenere Cipher pada Pengenkripsian Pesan Teks. *Journal on Education*, *5*(3), 8563–8573. https://doi.org/10.31004/joe.v5i3.1647

Hondro, R. K., & Fau, A. (2018). *PERANCANGAN APLIKASI PENYANDIAN TEKS DENGAN ALGORITMA ROT13 DAN TRIANGLE CHAIN CIPHER (TCC)*. *3*(2).

Hulu, F. N., & Putri, M. (2022). Metode Analitis Enkripsi Dan Dekripsi Dengan Penerapan Algoritma Kriptografi Klasik Ke Dalam Cipher. *Jurnal Elektro dan Telkomunikasi*, 26–34.

Jum'ah, M. N. Al, & Arifin. (2025). Analisis Pengaruh Kompresi File Pada Media Sosial Terhadap Ketahanan Image Steganografi Pada Metode Least Significant Bit (LSB). *CyberSecurity dan Forensik Digital*, *8*(1), 97–106.

Laksono, A. W., Suhada, S., & Zakaria, A. (2024). *IMPLEMENTASI METODE LEAST SIGNIFICANT BIT (LSB) DALAM TEKNIK STEGANOGRAFI PADA CITRA DIGITAL*. *4*(1).

Milian, Y. C., & Sulistyo, W. (2023). Model Pengembangan Keamanan Data dengan Algoritma ROT 13 Extended Vernam Cipher dan Stream Cipher. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *7*(2), 208–216. https://doi.org/10.35870/jtik.v7i2.716

Munawar, Z., Putri, N. I., Kharisma, I. L., Sid, S. S. K., Insany, P. G., Mogi, I Komang Ari, N., Ma'sum, H., Sastradipraja, C. K., Sukmana, R. N., & Barus, O. P. (2023). *KEAMANAN INFORMASI : Prinsip Dasar, Teori, dan Rekayasa Penerapan Konsep* (R. Komalasari (ed.)). Kaizen Media Publishing.

Pradeka, D. (2018). *PENYEMBUNYIAN INFORMASI DENGAN METODE CRYPTO-STEGANOGRAPHY MENGGUNAKAN MEDIA GAMBAR BERBASIS MOBILE*.

Rahman, R., Ariantini, M. S., Hadi, A., Hayati, N., Gunawan, P. W., Mandowen, S. A., Widiyasono, N., Gede Arna Jude Saskara, I. K., Salim, B. S., & Thantawi, A. M. (2024). *Buku Ajar Keamanan Jaringan Komputer* (E. Efitra & N. Safitri (ed.)). PT. Sonpedia Publishing Indonesia.

Sasono, D., Thir, M., Angel, F., Azizah, M., Utami, L., & Septiana, N. (2023). Perbandingan Kriptography Klasik Caesar Cipher Dengan Kriptography Modern Aes Dalam Tingkat Keamanan Jaringan Komputer. *Jurnal Informasi, Sains dan Teknologi*, *6*(1), 72–77.

Sitompul, D. S., Lubis, M. G. R., & Indra, Z. (2024). Implementasi ROT13 sebagai Algoritma Enkripsi Sederhana dalam Python untuk Keamanan Komunikasi Digital. *Journal of Accounting Law Communication and Technology*, *2*(1), 41–45. https://doi.org/10.57235/jalakotek.v2i1.4030

Syawal, M. F., Fikriansyah, D. C., & Agani, N. (2016). *Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher Dan Metode LSB*. *4*(3).

Utomo, I., Mulyono, W., Kusumawati, Y., & Ningrum, N. K. (2023). *Analisa Visual Citra Hasil Kombinasi Steganografi dan Kriptografi Berbasis Least Significant Bit Dalam Cipher*. *14*(1), 16–28.

Wardana. (2019). *Belajar Pemrograman dan Hacking Menggunakan Python* (Wardana (ed.)). Elex Media Komputindo.

Wildan, M., & Ashari, W. M. (2024). Text Data Security Using LCG and CBC with Steganography Technique on Digital Image. *Journal of Applied Informatics and Computing*, *8*(2), 400–407. https://doi.org/10.30871/jaic.v8i2.8457

Woźniak, K., Ogiela, M. R., & Ogiela, L. (2025). A Two-Phase Embedding Approach for Secure Distributed Steganography. *Sensors*, *25*(5), 1–19. https://doi.org/10.3390/s25051448

LAMPIRAN

A. *Source Code*

```python
def rot13(teks):
    hasil = ""

    teks = teks.upper()

    for huruf in teks:
        if huruf == " ":
            hasil += " "
        elif huruf.isalpha():
            pos = ord(huruf) - ord('A')
            pos = (pos + 13) % 26
            hasil += chr(pos + ord('A'))
    return hasil

# Program utama
if __name__ == "__main__":
    print("=== PROGRAM ROT13 ===")
    print("Ketik exit untuk berhenti.\n")

    riwayat = []

    while True:
        teks = input("Masukkan teks (exit untuk selesai): ").upper()

        if teks == "EXIT":
            print("\n=== RIWAYAT ENKRIPSI & DEKRIPSI ===")
            for i, r in enumerate(riwayat, 1):
                print(f"{i}. {r['input']} -> {r['encrypt']} -> {r['decrypt']}")
            print("\nProgram selesai. Terima kasih!")
            break

        if not all(c.isalpha() or c == " " for c in teks):
            print("Masukkan Inputan huruf!\n")
            continue

        hasil_encrypt = rot13(teks)
        hasil_decrypt = rot13(hasil_encrypt)

        print("Hasil Enkripsi :", hasil_encrypt)
        print("Hasil Dekripsi :", hasil_decrypt, "\n")

        riwayat.append({
            "input": teks,
            "encrypt": hasil_encrypt,
            "decrypt": hasil_decrypt
        })
```

B. Pembagian Tugas (*Jobdesk*) Kelompok 2

NIM	Nama Lengkap	*Jobdesk*
22104410075	ARVILANTI DEVANI	• Menyusun Bab I (Latar Belakang) dan Bab III bagian 3.1 (Alur Algoritma ROT13) • Mengatur logika pergeseran karakter dan pemisahan huruf besar-kecil pada fungsi rot13.
22104410088	RIZKI RAMADHAN	• Menyusun Bab II (Landasan Teori Kriptografi, Kriptografi Klasik, Caesar Cipher) dan Bab III bagian 3.5 (Petunjuk Penggunaan). • Menangani logika validasi penanda khusus (SIGN & STOP) untuk mencegah kesalahan ekstraksi.
22104410098	NUR CINDY INTAN FANDERELLA	• Menyusun Bab II (Landasan Teori ROT13), Bab III bagian 3.4 (Penjelasan Fungsi GUI), Bab IV (Kesimpulan), dan Lampiran. • Melakukan desain *layout* dan penataan elemen antarmuka menggunakan library CustomTkinter.
22104410101	HANIK HATUL HALIMAH	• Menyusun Bab I (Rmusan Masalah dan Tujuan), Bab II (Landasan Teori Format Gambar BMP dan JPG) dan Bab III bagian 3.2 & 3.3 (Alur Algoritma LSB & Implementasi). • Mengoptimalkan iterasi piksel pada kanal RGB untuk proses penyisipan bit LSB.
22104410121	WASI'ATUL JANNAH	• Menyusun Bab II (Landasan Teori Steganografi, LSB, Python). • Melakukan pengujian (*testing*) fungsionalitas *input* pesan dan pemilihan gambar pada sistem.
22104414002	SAHRUL RAMADHAN	• Menyusun Bab III bagian 3.6 (Analisis Kelebihan-Kekurangan), Final Editing laporan, Daftar Pustaka, dan penyesuaian format PDF. • Merancang arsitektur utama program, integrasi seluruh fungsi, dan sistem *error handling*.