Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 400~407
e-ISSN: 2548-6861

Text Data Security Using LCG and CBC with Steganography Technique on Digital Image

Muhammad Wildan [1]*, Wahid Miftahul Ashari [2]*
* Teknik Komputer, Universitas Amikom Yogyakarta
mhmmdwldn@students.amikom.ac.id [1], wahidashari@amikom.ac.id [2]

Article Info

Article history:

Received 2024-09-07
Revised 2024-09-24
Accepted 2024-09-30

Keyword:

AES-256,
Cryptography,
Linear Congurential Generator,
Steganography,
PSNR.

ABSTRACT

This research proposes a text data security method using a combination of Linear Congruential Generator (LCG), Advanced Encryption Standard (AES) Cipher Block Chaining (CBC) mode, and Least Significant Bit (LSB) steganography technique on digital images. The message scrambling process using LCG produces ASCII characters as noise that is inserted in the original message. After that, the message is encrypted using AES-256 CBC to provide additional security. The encryption result is then hidden in the digital image through LSB steganography technique. Tests were conducted on images with JPEG and BMP formats to measure the visual quality after the data insertion process, as measured by PSNR (Peak Signal-to-Noise Ratio). The test results show a PSNR value of 56.60 dB for JPEG images and 70.84 dB for BMP images. In addition, the insertion process in JPEG images degrades the image quality, mainly due to lossy compression, compared to the lossless BMP format. This study concludes that the proposed combination of methods is effective in hiding messages in images, but is susceptible to compression on lossy formats such as JPEG. The use of lossless image formats such as BMP or PNG is recommended to maintain data integrity.

I. PENDAHULUAN

Salah satu aspek krusial dalam keamanan informasi adalah melindungi kerahasiaan data teks, seperti dokumen digital, pesan, atau email. Data ini kerap berisi informasi sensitif yang perlu dijaga dengan baik. Berbagai teknik keamanan digunakan untuk melindungi data teks, dan salah satu teknik yang umum digunakan adalah kriptografi, yang berfungsi untuk mengenkripsi dan mendekripsi data [1]. Dalam upaya meningkatkan keamanan kerahasiaan pesan teks, penelitian ini menggabungkan beberapa metode, yaitu Linear Congruential Generator (LCG), Advanced Encryption Standard (AES) mode Chiper Block Chainning (CBC) dan Least Significant Bit (LSB) guna meningkatkan lapisan perlindungan terhadap data sensitif. Pada kriptografi, penelitian ini menggunakan metode AES-256 CBC, Modifikasi dengan metode Cipher Block Chaining (CBC) menambahkan keamanan karena setiap blok terenkripsi bergantung pada blok sebelumnya, sehingga membuat pola enkripsi lebih sulit ditebak [2].

Penggunaan Linear Congruential Generator (LCG) untuk menghasilkan deretan bilangan pseudo-acak yang kemudian diubah menjadi kode ASCII. Bilangan ASCII tersebut berfungsi sebagai noise untuk menyamarkan pesan sebelum proses enkripsi[3]. LCG dipilih karena efisiensinya dalam menghasilkan bilangan acak semu yang mudah diimplementasikan, meskipun memiliki kelemahan dalam hal prediktabilitas. Alternatif yang lebih aman seperti Cryptographically Secure Pseudo- Random Number Generators (CSPRNG) atau Mersenne Twister dapat dipertimbangkan. CBC digunakan karena kemampuannya untuk meningkatkan keamanan dengan menghubungkan setiap blok cipher dengan blok sebelumnya[4]. Namun, CBC juga rentan terhadap serangan bit-flipping yang dapat dimitigasi dengan penggunaan Message Authentication Code (MAC).

Selain menggunakan metode kriptografi, penelitian ini juga mengimplementasikan teknik steganografi. Steganografi adalah seni menyembunyikan informasi di dalam media lain,

seperti gambar, sehingga informasi tersebut tidak dapat dideteksi secara kasat mata.

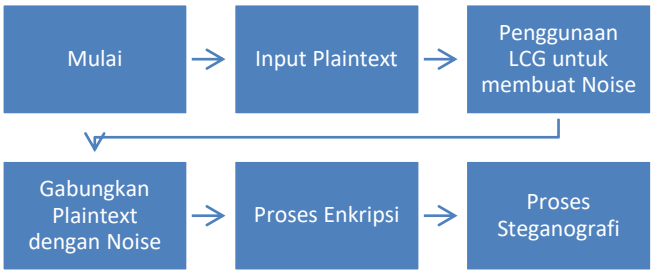
Teknik Least Significant Bit (LSB) digunakan dalam penelitian ini untuk menyisipkan informasi ke dalam gambar digital dengan memodifikasi bit paling tidak signifikan dari setiap piksel[5]. Terdapat beberapa penelitian terkait penggunaan kriptografi, Penelitian oleh [6]. menggunakan algoritma kriptografi DES untuk mengamankan data, di mana ukuran kunci yang digunakan hanya 56 bit. Sementara itu, pada penelitian [7]. membandingkan algoritma kriptografi DES dengan AES dalam mengamankan file dokumen, dan hasilnya menunjukkan bahwa waktu proses enkripsi dengan AES lebih cepat dibandingkan DES. Pada penelitian oleh[8]. hasil data teks yang telah dienkripsi menggunakan algoritma Advanced Encryption Standard (AES) 128-Bit dilanjutkan dengan teknik steganografi pada sebuah citra menggunakan algoritma Discrete Wavelet Transform (DWT) menghasilkan nilai MSE 0.16-0.26 dB dan nilai PSNR nya 46.27-52.2 dB,

Pada penelitian berikutnya [9]. melakukan penelitian dengan melakukan enkripsi menggunakan algoritma DCT Hasil pengujian dengan menggunakan Avalanche effect diperoleh nilai 49.25%. Penelitian ini bertujuan untuk meningkatkan keamanan data dengan menyamarkan pesan asli agar tidak mudah dipahami. Prosesnya melibatkan pengacakan pesan menggunakan algoritma LCG (Linear Congruential Generator) yang kemudian dikonversi ke format ASCII. Setelah itu, pesan dienkripsi menggunakan metode AES-256 CBC, sehingga memberikan lapisan keamanan tambahan. Selanjutnya, pesan yang sudah terenkripsi tersebut disembunyikan dalam gambar digital menggunakan teknik steganografi metode LSB (Least Significant Bit). Dengan demikian, pesan tidak hanya dienkripsi tetapi juga tersamarkan dalam bentuk yang sulit terdeteksi.

II. METODE PENELITIAN

A. Kerangka Dasar Penelitian

Penelitian ini berfokus pada pengamanan pesan teks dengan menggabungkan metode Linear Congruential Generator (LCG), enkripsi Cipher Block Chaining (CBC), dan teknik steganografi Least Significant Bit (LSB).



Gambar 1. Flowchart Proses Encoding

Algoritma LCG digunakan untuk menghasilkan bilangan ASCII yang berfungsi sebagai noise untuk menyamarkan pesan agar lebih sulit dipahami. Setelah itu, pesan yang telah disamarkan dengan noise tersebut dienkripsi menggunakan algoritma Advanced Encryption Standard (AES) dengan mode CBC. Teknik steganografi LSB kemudian diterapkan untuk menyembunyikan pesan yang telah dienkripsi ke dalam gambar digital. Tujuan penelitian ini adalah untuk mengevaluasi efektivitas kombinasi metode LCG, steganografi, dan kriptografi dalam pengamanan data.

B. Tahapan Penelitian

1) Perhitungan Linear Congruential Generator (LCG). Linear Congruential Generator (LCG) merupakan salah satu algoritma pembangkit angka acak semu (pseudo- random number generator) yang paling umum. Algoritma ini memiliki teori yang sederhana dan mudah dimengerti, serta dapat diimplementasikan dengan cepat. Salah satu keunggulan utama dari LCG adalah kemampuannya untuk beroperasi dengan sangat cepat[10]. Algoritma ini bekerja dengan menetapkan nilai awal (seed), kemudian menghasilkan bilangan acak melalui perhitungan dengan angka bulat besar, sehingga kemungkinan munculnya bilangan yang sama sangat kecil. Parameter LCG terdiri dari seed (X_0), yaitu nilai awal yang digunakan untuk memulai proses generasi bilangan acak, multiplier (a), yang merupakan konstanta yang digunakan untuk mengalikan bilangan sebelumnya, increment (c), sebagai konstanta yang ditambahkan setelah proses perkalian, dan modulus (m), yaitu bilangan yang digunakan sebagai pembagi untuk menentukan rentang hasil bilangan acak [11].

Proses menentukan nilai awal (seed) X_0 untuk menghitung bilangan menggunakan rumus LCG. Kemudian, ambil sisa hasil pembagian perhitungan tersebut dengan modulus m, di mana sisa ini akan menjadi bilangan acak yang baru. Setelah itu, bilangan acak dikonversi menjadi karakter ASCII. Agar karakter yang dihasilkan berada dalam rentang ASCII yang dapat dicetak (33-127), tambahkan 33 pada hasil dari operasi modulo 95. Untuk iterasi ke-n, gunakan rumus berikut.

$$Iterasi\ ke-n$$
$$Xn+1 = (a * Xn + c)\ mod\ m$$

$$Konversi\ ke\ ASCII$$
$$Karakter = Hasil\ LCG\ mod\ 95 + 33$$

Perhitungan sebelum melakukan perhitungan, peneliti terlebih dahulu menentukan jumlah karakter ASCII yang dibutuhkan. Pada kasus ini, perhitungan dilakukan sebanyak 5 kali. Hal ini karena semakin banyak angka yang dihasilkan, semakin tinggi kemungkinan noise tercampur dengan teks yang sebenarnya.

Seed (X_0): 123
Multiplier (a): 1664525
Increment (c): 1013904223
Modulus (m): 232 = 4294967296

TABEL I.
PERHITUNGAN LCG

Iterasi Ke-	Perhitungan LCG	Hasil LCG (mod 22^32)	Konversi ke ASCII	Karakter
1	(1664525 * 123 + 1013904223) mod 2^32	1218640798	1218640798 mod 95 + 33 = 116	t
2	(1664525 * 1218640798 + 1013904223) mod 2^32	1868869221	1868869221 mod 95 + 33 = 89	Y
3	(1664525 * 1868869221 + 1013904223) mod 2^32	166005888	166005888 mod 95 + 33 = 71	G
4	(1664525 * 166005888 + 1013904223) mod 2^32	948671967	948671967 mod 95 + 33 = 100	d
5	(1664525 * 948671967 + 1013904223) mod 2^32	1543727538	1543727538 mod 95 + 33 = 86	V

2) Penggabungan Pesan dan Hasil LCG. Pada tahap lanjutan dalam proses penggabungan plainteks dan karakter acak yang dihasilkan melalui metode Linear Congruential Generator (LCG), penggabungan ini melibatkan pendekatan berbasis interval yang secara bergantian mencampurkan kedua string (plainteks dan noise) dengan pola yang terorganisir. Dalam konteks ini, interval berfungsi sebagai parameter kunci yang menentukan berapa banyak karakter yang diambil secara bergantian dari masing-masing string sebelum menambahkan karakter dari string lainnya.

TABEL II.
PLAINTEKS DAN NOISE

No	Teks	Isi Teks
1	Plainteks	Amikom
2	Noise	tYGdV

Proses, membuat variabel hasil untuk menyimpan teks gabungan dari proses penggabungan plainteks dan noise. Dua indeks, indeks1 (P) dan indeks2 (N), digunakan untuk melacak posisi karakter pada masing-masing string. Loop utama akan berjalan selama masih terdapat karakter yang belum digabungkan dari salah satu teks. Ambil sejumlah karakter dari plainteks berdasarkan nilai interval yang dimulai dari indeks1, lalu tambahkan karakter tersebut ke dalam variabel hasil. Setelah karakter diambil, tambahkan interval tersebut ke indeks1. Jika indeks1 telah mencapai akhir dari plainteks, maka tidak ada lagi karakter yang diambil dari plainteks. Kemudian, ambil karakter dari noise berdasarkan interval yang dimulai dari indeks2, lalu tambahkan ke hasil. Setelah mengambil karakter, tambahkan interval ke indeks2. Jika indeks2 telah mencapai akhir dari karakter noise, maka tidak ada lagi karakter yang diambil dari noise.

Perhitunga, teks plainteks (P): "Amikom" Teks ASCII (A) : "tYGdV" dan Interval (I) : 1.

TABEL I.
PROSES INTERVAL

Iterasi	Plainteks (P)	Noise (N)	Hasil penggabungan Sementara
1	A	t	At
2	m	Y	AtmY
3	i	G	AtmYiG
4	k	d	AtmYiGkd
5	o	V	AtmYiGkdoV
6	m	-	AtmYiGkdoVm

Hasil Akhir: AtmYiGkdoVm

3) Proses Enkripsi. Proses enkripsi ini bertujuan untuk melindungi pesan asli (plainteks) yang telah digabungkan dengan noise menggunakan algoritma enkripsi simetris AES-256. Proses enkripsi dilakukan dengan membagi plainteks menjadi beberapa blok, di mana setiap blok dienkripsi menggunakan vektor inisialisasi (IV) dan kunci (key). Plainteks hasil penggabungan dengan noise dipecah menjadi blok-blok yang siap dienkripsi, seperti P1, P2, hingga Pn[12]. Blok pertama plainteks (P1) dienkripsi dengan rumus beikut.

$$C1 = Ek\ (P1 \oplus IV)$$

Di mana C1 adalah chiperteks blok pertama, Ek adalah enkripsi dengan kunci k, $\oplus$ adalah operasi XOR. Untuk blok kedua hingga blok ke-n (i = 2 hingga n):

$$Ci = Ek\ (Pi \oplus Ci - 1)$$

Blok plainteks di-XOR dengan blok chiperteks sebelumnya sebelum dienkripsi. Padding ditambahkan ke blok pertama agar sesuai dengan panjang blok AES (16 byte). Padding yang digunakan berupa 5 byte dengan nilai heksadesimal 05. Hasil enkripsi diperoleh dengan proses enkripsi menggunakan AES-256 untuk menghasilkan blok chiperteks.

TABEL IV.
KARAKTER DAN KUNCI ENKRIPSI

No	Deskripsi	Nilai
1	Plainteks	AtmYiGkdoVm
2	IV	e71aafcf1126f296cd0695739dd51722
3	Key	e152feebd44c097cc2ee5d9fdf2d89873d 60d3f29577b63e5ebe0e70eb9c222f
4	Plainteks Heksadesimal	41 74 6d 59 69 47 6b 64 6f 56 6d

Plainteks dalam bentuk teks "AtmYiGkdoVm" dikonversi ke nilai heksadesimal sebagai: 41 74 6d 59 69 47 6b 64 6f 56 6d . Padding ditambahkan: 05 05 05 05 05.

TABEL V.
PERHITUNGAN OPERASI XOR (BLOK 1 DENGAN IV)

Blok 1 Plainteks	IV	Hasil XOR
41 74 6D 59 69 47 6B 64 6F 56 6D 05 05 05 05 05	e7 1a af cf 11 26 f2 96 cd 06 95 73 9d d5 17 22	a6 6e c2 96 78 61 99 f2 a2 50 f8 76 98 d0 12 27

Hasil XOR di atas kemudian dienkripsi menggunakan algoritma AES-256, menghasilkan blok cipher C1 dengan nilai: 97 39 0f ee a1 a1 ad 25 bf 65 94 de 38 e9 5f b9

Blok chiperteks C1 dikodekan dalam format Base64, menghasilkan: lzkP7qGhrSW/ZZTeOOlfuQ==

4) Proses Encoding. Pesan disisipkan ke dalam gambar menggunakan metode Least Significant Bit (LSB). Metode ini bekerja dengan menggantikan bit paling tidak signifikan dari setiap piksel gambar dengan bit plainteks yang telah dikonversi ke format biner[13]. Dengan cara ini, pesan dapat disembunyikan tanpa mengubah tampilan visual gambar secara signifikan, karena perubahan hanya terjadi pada bit- bit terkecil dari tiap piksel. Setiap piksel dalam gambar terdiri dari tiga saluran warna, yaitu Red (R), Green (G), dan Blue (B), di mana masing-masing saluran memiliki 8 bit. Artinya, setiap piksel dapat menyimpan 3bit pesan, dengan menyisipkan satu bit pesan di masing-masing saluran warna.

Dengan jumlah total 576.225 piksel, maka kapasitas total penyisipan pesan dalam gambar adalah 576.225 $piksel$ × 3 $bit/piksel$ = 1.728.675 bit.

Panjang chiperteks yang akan disisipkan adalah string "lzkP7qGhrSW/ZZTeOOlfuQ==", yang memiliki 24 karakter. Karena setiap karakter ASCII membutuhkan 8 bit, panjang chiperteks dalam bit adalah: 24 $karakter$ × 8 $bit/karakter$ = 192 bit.

Peneliti kemudian melakukan penyisipan bit dari chiperteks yang telah dikonversi ke biner ke dalam bit paling tidak signifikan dari tiap saluran warna RGB di gambar. Berikut adalah contoh proses penyisipan bit ke piksel gambar.

Pertama, kita konversi tiga karakter chiperteks "l", "z", dan "k" menjadi biner:

l: 01101100 z: 01111010 k: 01101011.

Selanjutnya, kita pilih piksel di koordinat (850, 585) dengan nilai RGB:

R: 64 → 01000000 G: 172 → 10101100 B: 148 → 10010100

Proses penyisipan bit dilakukan dengan mengganti bit terakhir (LSB) dari tiap channel RGB dengan bit dari chiperteks.

TABEL VI.
PROSES PENYISIPAN BIT CHIPERTEKS KE PIKSEL

Piksel (RGB)	Sebelum Penyisipan	Bit Chiperteks	Setelah Penyisipan
R (Red)	64 → 01000000	0	64 → 01000000
G (Green)	172 → 10101100	1	173 → 10101101
B (Blue)	148 → 10010100	1	149 → 10010101

5) Proses Decoding. Proses dekode ini bertujuan untuk mengekstrak chiperteks yang telah disisipkan ke dalam gambar dan mengembalikan pesan ke bentuk aslinya[14]. Proses dekode dimulai dengan mengambil nilai RGB dari piksel yang digunakan saat proses encoding. Sebagai contoh, digunakan piksel pada koordinat (850, 585) dengan nilai RGB sebagai berikut:

R = 64, G = 173, dan B = 149.

Langkah berikutnya adalah mengekstraksi bit chiperteks dari bit paling tidak signifikan (Least Significant Bit atau LSB) dari masing-masing saluran warna. LSB dari nilai RGB yang telah dikonversi ke biner adalah:

LSB dari R (01000000) = 0
LSB dari G (10101101) = 1
LSB dari B (10010101) = 1

Dengan demikian, bit chiperteks yang diekstraksi dari piksel ini adalah 011. Proses yang sama dilakukan pada piksel-piksel lain yang digunakan dalam proses encoding. Setelah beberapa piksel, bit-bit yang diekstraksi membentuk urutan biner lengkap. Sebagai contoh, bit chiperteks yang diperoleh dari beberapa piksel adalah: 01101100 01111010 01101011. Setelah semua bit chiperteks terkumpul, urutan biner tersebut dikonversi ke dalam karakter ASCII. Urutan biner 01101100 01111010 01101011 dikonversi menjadi karakter ASCII: lzk. Proses ini diulang hingga seluruh chiperteks yang disembunyikan di dalam gambar berhasil diekstraksi. Hasil akhirnya adalah string chiperteks yang lengkap: lzkP7qGhrSW/ZZTeOOlfuQ==

6) Proses Dekripsi. Setelah chiperteks diperoleh dari proses decoding, langkah selanjutnya adalah mendekripsi chiperteks tersebut untuk memulihkan plainteks asli[15-17]. Proses dekripsi dilakukan menggunakan algoritma AES 256 dengan beberapa tahapan berikut.

Chiperteks yang akan didekripsi adalah lzkP7qGhrSW/ZZTeOOlfuQ==, yang telah dikonversi menjadi blok chiperteks dalam bentuk heksadesimal sebagai berikut:

Ciphertext Blok C1:

97 39 0f ee a1 a1 ad 25 bf 65 94 de 38 e9 5f b9 Vektor inisialisasi (IV) yang digunakan untuk proses enkripsi dan dekripsi adalah

IV: e7 1a af cf 11 26 f2 96 cd 06 95 73 9d d5 17 22 Kunci enkripsi/dekripsi yang digunakan adalah

Key:
e152feebd44c097cc2ee5d9fdf2d89873d60d3f29577b63e5
ebe0e70eb9c222f

Langkah pertama adalah mendekripsi blok chiperteks C1. Proses dekripsi dengan AES 256 menggunakan kunci yang telah ditentukan menghasilkan hasil dekripsi berikut dalam format heksadesimal:

Hasil Dekripsi Blok C1:
a6 6e c2 96 78 61 99 f2 a2 50 f8 76 98 d0 12 27

Setelah mendapatkan hasil dekripsi, langkah berikutnya adalah melakukan operasi XOR dengan vektor inisialisasi (IV) untuk mendapatkan plainteks yang sebenarnya. Proses XOR dilakukan antara hasil dekripsi dan IV.

TABEL VII.
PROSES DEKRIPSI

Blok C1 (Hasil Dekripsi)	IV	Hasil XOR
a6 6e c2 96 78 61 99 f2 a2 50 f8 76 98 d0 12 27	e7 1a af cf 11 26 f2 96 cd 06 95 73 9d d5 17 22	41 74 6d 59 69 47 6b 64 6f 56 6d 05 05 05 05 05

Hasil XOR ini memberikan plainteks dalam bentuk heksadesimal:

41 74 6d 59 69 47 6b 64 6f 56 6d 05 05 05 05 05

Blok hasil ini memiliki padding di bagian akhir dengan nilai heksadesimal 0x05 yang menunjukkan bahwa terdapat 5byte padding. Padding ini ditambahkan selama proses enkripsi untuk melengkapi ukuran blok agar sesuai dengan ukuran blok AES (16 byte). Padding dihapus menggunakan metode PKCS#7, yang memberikan blok plainteks asli:

41 74 6d 59 69 47 6b 64 6f 56 6d

Setelah padding dihapus, plainteks asli diperoleh dengan mengonversi nilai heksadesimal ke teks ASCII: AtmYiGkdoVm

Proses dekripsi ini berhasil memulihkan plainteks asli dari chiperteks yang telah disembunyikan dan dienkripsi.

7) Pemisahan Plainteks. Metode Regular Expression (RegEx), yang sangat efektif untuk membedakan atau memfilter teks berdasarkan pola tertentu. Metode ini memungkinkan untuk secara mudah mengidentifikasi dan memisahkan bagian plainteks yang valid dari noise yang ada dalam string.

Proses pemisahan menggunakan Regular Expression (RegEx) dilakukan dengan pola yang menangkap pasangan karakter berurutan dalam string. Simbol (.) digunakan untuk menangkap satu karakter, sedangkan (.)? menangkap karakter kedua dalam pasangan, jika ada. Metode ini diterapkan untuk mengelompokkan karakter ke dalam dua grup: grup karakter pada posisi ganjil dan grup karakter pada posisi genap. Jika pasangan tidak lengkap (misalnya, string berakhir dengan satu karakter), karakter yang tersisa dimasukkan ke dalam posisi ganjil.

Pada iterasi pertama, pasangan "At" diproses, di mana A dimasukkan ke dalam grup posisi ganjil dan t dimasukkan ke grup posisi genap. Pada iterasi kedua, pasangan "mY" dipisahkan menjadi m (posisi ganjil) dan Y (posisi genap). Iterasi ketiga, pasangan "iG" memberikan i (posisi ganjil) dan G (posisi genap). Proses ini berlanjut hingga iterasi terakhir, di mana karakter tunggal m (tanpa pasangan) ditempatkan pada posisi ganjil.

Hasil akhir dari proses ini menghasilkan dua kelompok karakter:

Karakter di posisi ganjil: ['A', 'm', 'i', 'k', 'o', 'm']
Karakter di posisi genap: ['t', 'Y', 'G', 'd', 'V']

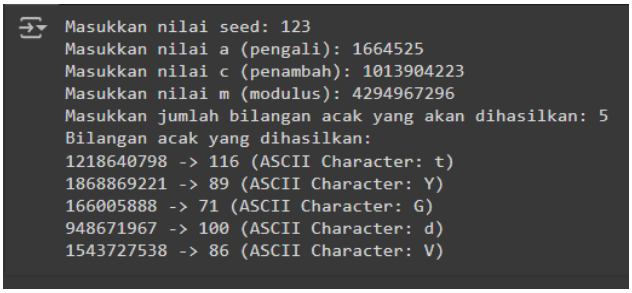
III. HASIL DAN PEMBAHASAN

A. *Implementasi Sistem*

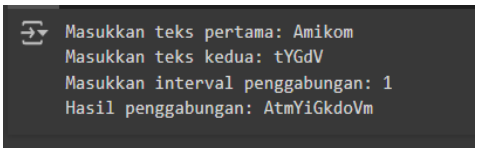
Pada bagian ini, setelah perancangan, implementasi adalah tahap berikutnya. enkripsi AES mode CBC dan encoding LSB akan ditunjukkan, dan hasilnya akan menunjukkan antarmuka sistem.

1) LCG dan interval

Penggunaan Google Colab sebagai perantara untuk menjalankan algoritma LCG dan menentukan interval. Output yang dihasilkan setelah memasukkan rumus LCG, serta interval plainteks dan noise, ditampilkan pada gambar berikut.



Gambar 2. Output LCG



Gambar 3. Output Interval

2) AES Mode CBC

Aplikasi ini adalah aplikasi berbasis web yang menggunakan enkripsi AES dengan mode CBC. Pengguna dapat mengakses menu enkripsi, di mana mereka bisa memasukkan kunci (key) untuk menghasilkan cipherteks dalam format base64. Selain itu, aplikasi ini juga menyediakan fitur dekripsi yang dapat mengembalikan cipherteks ke bentuk plainteks aslinya.

Gambar 4. Tampilan Menu CBC

3) Encoding LSB

Aplikasi ini adalah aplikasi steganografi berbasis web. Pengguna dapat memilih file gambar sebagai cover image melalui menu "Enkripsi Steganografi". Pada tahap ini, pengguna dapat menyisipkan pesan ke dalam gambar menggunakan fungsi "Encode Message". Setelah proses encoding selesai, gambar yang berisi pesan tersebut akan otomatis diunduh. Selain itu, dengan fitur "Decode Message", pesan yang telah disisipkan sebelumnya dapat diekstraksi dengan mengunggah kembali gambar yang sudah melalui proses encoding.

Gambar 5. Tampilan Menu LSB

B. Pengujian Citra

Pengujian citra ini bertujuan untuk menilai kualitas gambar sebelum dan sesudah proses penyisipan data digital. Kualitas gambar diukur menggunakan teori PSNR (Peak Signal-to-Noise Ratio) dan analisis histogram. Pengujian ini melibatkan gambar dengan ukuran piksel yang sama namun menggunakan format yang berbeda, sehingga memungkinkan perbandingan kualitas antara kedua format tersebut. Berikut adalah langkah-langkah dalam pengujiannya.

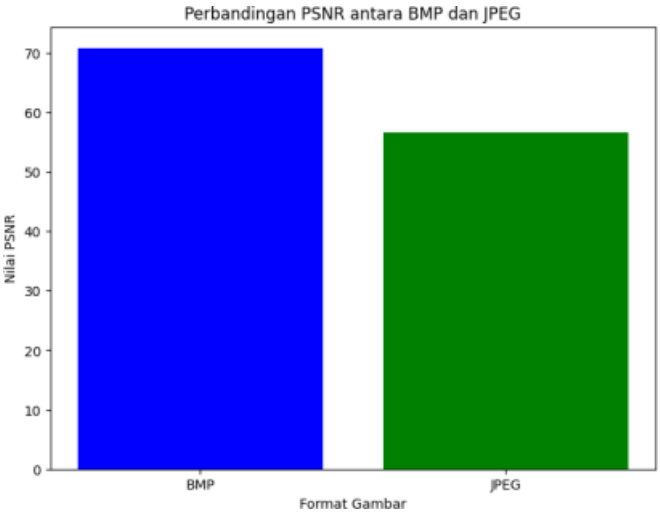
TABEL VIII.
HASIL PENGUJIAN ENCODING JPEG

No	Nama File	Ukuran File	Nilai PSNR (db)
1	Pantai.jpeg	87.0 KB	56.60
2	Pinguin.jpeg	34.2 KB	70.82

TABEL IX.
HASIL PENGUJIAN ENCODING BMP

No	Nama File	Ukuran File	Nilai PSNR (db)
1	Pantai.bmp	2.19 MB	70.84
2	Pinguin.bmp	337 KB	56.03

Berdasarkan tabel 8 dan 9 hasil pengujian PSNR yang dilakukan pada citra JPEG dan BMP berhasil dengan baik dengan rata-rata nilai PSNR yang cukup baik. Berikut grafik nilai PSNR dari kedua format citra dapat dilihat pada gambar6.



Gambar 6. Grafik Perbandingan PSNR

Berdasarkan grafik pada gambar 6, dapat disimpulkan bahwa baik file gambar berformat JPG maupun BMP memiliki tingkat kualitas citra yang cukup baik. Namun, ukuran pesan yang dapat disembunyikan dalam gambar melalui teknik steganografi tergantung pada format gambar, metode yang digunakan, dan ukuran gambar itu sendiri. Format gambar lossless seperti PNG atau BMP memungkinkan kapasitas penyembunyian data yang lebih besar dibandingkan format lossy seperti JPEG, yang lebih mudah terdeteksi karena proses kompresinya. Metode LSB (Least Significant Bit) adalah teknik yang paling umum digunakan, di mana bit terakhir dari setiap piksel diubah untuk menyimpan data tanpa mengurangi kualitas visual secara signifikan. Dalam gambar 24-bit RGB, biasanya 1-

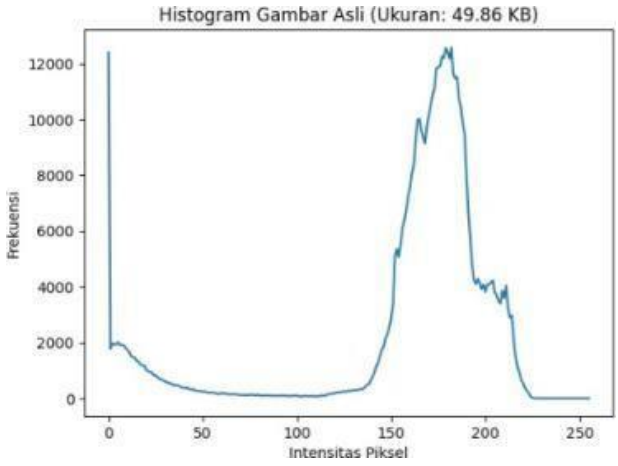
3 bit per piksel dapat disembunyikan tanpa terlihat. Gambar BMP 24-bit (lossless) tanpa kompresi, memungkinkan penyembunyian pesan sekitar 300-500 kB pada gambar berukuran 1 MB menggunakan teknik LSB tanpa penurunan kualitas visual. Sebaliknya, dalam gambar JPEG 24-bit (lossy), kompresi membatasi kapasitas penyembunyian, hanya memungkinkan sekitar 20-50 kB pesan dalam gambar ukuran yang sama, karena perubahan lebih mudah terdeteksi dan memengaruhi kualitas gambar.

TABEL X.
CITRA DIGITAL

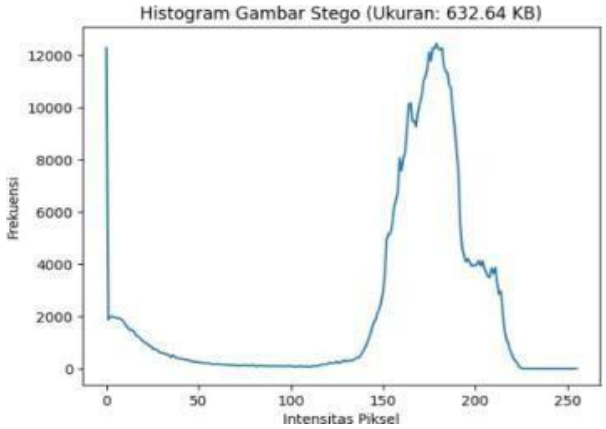
Cover Citra	Stego Citra
	

C. Pengujian Histogram

Pengujian histogram ini bertujuan untuk mengukur tingkat perubahan pada gambar setelah ditambahkan pesan dibandingkan dengan gambar asli sebelum diberi pesan. Hasil dari pengujian histogram ditampilkan pada gambar 7 dan 8.



Gambar 7. Histogram gambar asli



Gambar 8. Histogram gambar Stego

Perbedaan antara histogram gambar asli dan gambar stego tidak terlalu terlihat, meskipun kedua gambar tersebut memiliki ukuran yang berbeda. Gambar stego umumnya berukuran lebih besar dibandingkan dengan gambar asli. Metode LSB memiliki kelemahan signifikan ketika diterapkan pada gambar dengan format kompresi lossy, seperti JPEG. Kompresi lossy cenderung mengubah bit-bit terakhir dari setiap piksel yang digunakan untuk menyisipkan data, yang berpotensi merusak pesan tersembunyi. Sebaliknya, pada format gambar lossless seperti PNG atau BMP, LSB lebih stabil karena kompresi tidak mempengaruhi bit-bit gambar. Untuk mempertahankan integritas data selama kompresi gambar, disarankan menggunakan format gambar lossless atau mengadopsi metode steganografi berbasis transformasi domain seperti Discrete Cosine Transform (DCT) atau Discrete Wavelet Transform (DWT), yang lebih tahan terhadap kompresi lossy.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa metode LCG + CBC + LSB efektif untuk menyembunyikan pesan dalam gambar digital, namun memiliki beberapa kelemahan. Pertama, metode LSB tidak tahan terhadap kompresi lossy seperti JPEG, yang dapat merusak pesan tersembunyi. Pengujian menunjukkan penurunan nilai PSNR pada gambar JPEG, seperti Pantai.jpeg yang memiliki PSNR 56,60 dB, dibandingkan dengan gambar BMP seperti Pantai.bmp yang mencapai 70,84 dB. Oleh karena itu, format gambar lossless seperti PNG atau BMP lebih disarankan untuk menjaga integritas data. Steganografi berbasis transformasi domain seperti DCT atau DWT lebih tahan terhadap kompresi lossy, sehingga lebih sesuai untuk aplikasi dengan risiko kompresi ulang. Kedua, penggunaan Linear Congruential Generator (LCG) sebagai pembangkit bilangan acak kurang aman karena prediktabilitasnya. Sebagai alternatif, CSPRNG atau Mersenne Twister lebih direkomendasikan untuk keamanan yang lebih baik. Dari segi kinerja, metode ini efisien untuk gambar beresolusi tinggi tanpa penurunan kualitas visual yang signifikan, meskipun enkripsi AES-256 dalam mode CBC meningkatkan waktu pemrosesan. LSB lebih cepat dalam menyisipkan pesan, tetapi kurang tahan terhadap kompresi dibandingkan metode berbasis DCT atau DWT, yang lebih cocok untuk aplikasi yang melibatkan kompresi gambar ulang.

DAFTAR PUSTAKA

[1] Putra Rahmadi and Hilda Dwi Yunita, “Implementasi Pengamanan Basis Data Dengan Teknik Enkripsi,” J. Cendikia, vol. 19, no. 1, pp. 413–418, 2020.

[2] P. A. Rizky and S. Soim, “Implementasi Algoritma Kriptografi AES CBC Untuk Keamanan Komunikasi Data Pada Hardware,” vol. 3, no. 1, pp. 71–78, 2024.

[3] D. N. Triwibowo, P. -, I. A. Ashari, A. S. Sandi, and Y. F. Rahman, “Enkripsi Pesan Menggunakan Algoritma Linear Congruential Generator (LCG) dan Konversi Kode Morse,” Bul. Ilm. Sarj. Tek. Elektro, vol. 3, no. 3, pp. 194–201, 2022, doi: 10.12928/biste.v3i3.5546.

[4] A. Nugrahantoro, A. Fadlil, and I. Riadi, “Optimasi Keamanan Informasi Menggunakan Algoritma Advanced Encryption Standard (AES) Mode Chiper Block Chaining (CBC),” J. Ilm. FIFO, vol. 12, no. 1, p. 12, 2020, doi: 10.22441/fifo.2020.v12i1.002.

[5] M. Afsari, D. I. Mulyana, A. Damaiyanti, and N. Sa’adah, “Implementasi Mode Operasi Kombinasi Cipher Block Chaining dan Metode LSB-1 Pada Pengamanan Data text,” J. Pendidik. Sains dan Komput., vol. 2, no. 01, pp. 70–82, 2022, doi: 10.47709/jpsk.v2i01.1381.

[6] M. Siahaan and J. Manurung, “Perancangan Aplikasi Penyandian Teks Menggunakan Algoritma Triple DES,” J. Ilmu Komput. dan Sist. vol. 3, no. 3, pp. 197–201, 2021, [Online]. Available: http://ejournal.sisfokomtek.org/index.php/jikom/a rticle/view/116

[7] M. Suwarni, J. Wahyudi, and K. Khairil, “Comparison of the DES Cryptographic Algorithm and the AES Algorithm in Securing Document Files,” J. Media Comput. Sci., vol. 2, no. 1, pp. 41–48, 2023, doi: 10.37676/jmcs.v2i1.3348.

[8] C. Umam and M. Muslih, "Enkripsi Data Teks Dengan AES dan Steganografi DWT," InComTech J. Telekomun. dan Komput., vol. 13, no. 1, p. 28, 2023, doi: 10.22441/incomtech.v13i1.15059.

[9] W. Diaztary, D. Atmajaya, F. Umar, Purnawansyah, Harlinda, and S. M. Abdullah, "Tiny Encryption Algorithm on Discrete Cosine Transform Watermarking," 3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021, pp. 415–420, 2021, doi: 10.1109/EIConCIT50028.2021.9431930.

[10] B. O. Sinaga, S. Sinurat, and T. Zebua, "Modifikasi Algoritma XTEA dengan Pembangkitan Kunci Menggunakan Metode Linear Congruential Untuk Pengamanan File Dokumen," J. Informatics …, vol. 1, no. 4, pp. 144–152, 2021.

[11] M. Furqan, Sriani, and D. Vita, "Application Of RSA-CRT Cryptographic Methods And LSB- LCG Steganography Method In Message Security Systems With Video Media," J. Infokum, vol. 10, no. 3, pp. 93–103, 2022.

[12] A. Sudrajat, Y. H. Prasetyo, and M. Kusumawardani, "Implementasi Enkripsi Advanced Encryption Standard (AES-128) Mode Cipher Block Chaining (CBC) sebagai Keamanan Komunikasi Pergerakan Robot Humanoid KRSBI," J. Jartel J. Jar. Telekomun., vol. 11, no. 1, pp. 6–11, 2021, doi: 10.33795/jartel.v11i1.16.

[13] N. A. Ramadhani and I. Susilawati, "Penerapan Steganografi untuk Penyisipan Pesan Teks pada Citra Digital dengan Menggunakan Metode Least Significant Bit," J. Multimed. Artif. Intell., vol. 4, no. 1, pp. 21–27, 2020.

[14] A. P. Ratnasari and F. A. Dwiyanto, "Metode Steganografi Citra Digital," Sains, Apl. Komputasi dan Teknol. Inf., vol. 2, no. 2, p. 52, 2020, doi: 10.30872/jsakti.v2i2.3300.

[15] W. Mahmud and E. Mintorini, "Pengamanan Data Kombinasi Metode Cipher Block Chaining dan Modifikasi LSB," Techno.Com, vol. 19, no. 1, pp. 97–102, 2020, doi: 10.33633/tc.v19i1.3375

[16] D. E. Kurniawan and N. Narupi, "Teknik Penyembunyian Data Menggunakan Kombinasi Kriptografi Rijndael dan Steganografi Least Significant Bit (LSB)", JuTISI, vol. 2, no. 3, Dec. 2016.

[17] D. E. Kurniawan, N. R. Hartadi, and P. Prasetyawan, "Analisis Hasil Teknik Penyembunyian Hak Cipta Menggunakan Transformasi DCT dan RSPPMC pada Jejaring Sosial," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 5, no. 3, Art. no. 3, Aug. 2018, doi: 10.25126/jtiik.201853692.