Jurnal EurekaMatika

Journal homepage: https://ejournal.upi.edu/index.php/JEM

Penggabungan *Affine Cipher* dan *Least Significant Bit-2* untuk Penyisipan Pesan Rahasia pada Gambar

Firda Kurniasih, Rini Marwati, dan Ririn Sispiyati*

Program Studi Matematika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Indonesia

*Correspondence: E-mail: firda.k@upi.edu

A B S T R A K

Seiring berkembangnya teknologi, peningkatan keamanan pesan menjadi sangat penting. Salah satu pendekatan yang dapat dilakukan untuk meningkatkan keamanan pesan adalah dengan menggabungkan kriptografi dan steganografi. Pada penelitian ini, penulis melakukan penggabungan antara teknik kriptografi Affine Cipher dan steganografi Least Significant Bit-2 (LSB-2). Dalam metode penggabungan ini, keamanan pesan dilakukan dengan menerapkan kriptografi *Affine Cipher* terlebih dahulu. Kemudian, pesan yang telah dienkripsi disisipkan ke dalam citra menggunakan LSB-2. Lebih spesifiknya, LSB-2 merupakan modifikasi dari metode Least Significant Bit (LSB) dengan menukarkan bit ke enam dari setiap elemen warna pixel pada gambar dengan bit-bit dari pesan rahasia yang ingin disembunyikan. LSB-2 yang digunakan merupakan LSB-2 secara acak. Bilangan acak yang diperlukan dalam proses ini dihasilkan melalui *Pseudo Random Number Generator* (PRNG). Hasil penelitian ini berupa aplikasi komputer dengan bahasa pemrograman Python versi 3.11 yang dapat menyamarkan pesan dan menyisipkannya dalam gambar. Berdasarkan implementasi, diperoleh hasil bahwa penggabungan *Affine Cipher* dan LSB-2 dapat mempersulit kriptanalisis karena harus meretas dua algoritma dan pesan disembunyikan secara acak keberadaannya di dalam gambar.

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima 2 September 2023
Direvisi 7 Oktober 2023
Disetujui 18 Oktober 2023
Tersedia online 1 November 2023
Dipublikasikan 1 Desember 2023

Kata Kunci:
Affine Cipher,
Keamanan Pesan Kriptografi,
LSB-2,
Steganografi.

A B S T R A C T

As technology develops, improving message security is very important. One approach that can be taken to improve message security is to combine cryptography and steganography. In this study, the authors combined Affine Cipher cryptographic and Least Significant Bit-2 (LSB-2) steganography. In this merge method, the Affine Cipher cryptography is applied first, then, the encrypted message is inserted into an image using LSB-2. Specifically, LSB-2 is a modification of the Least Significant Bit (LSB) method which works by exchanging the sixth bits of each pixel color element in the image with the bits of the secret message. The LSB-2 used is a random LSB-2. In this case, the random numbers are generated by the Pseudo Random Number Generator (PRNG). The results of this research are a computer application with the Python language version 3.11 which can disguise messages and insert them in an image. Based on the implementation, the combination Affine Cipher and LSB-2 can complicate cryptanalysis because it has to hack two algorithms and the message is hidden randomly in an image.

Keywords:
Affine Cipher,
Cryptography,
LSB-2,
Message Security,
Steganography.

1. PENDAHULUAN

Seiring perkembangan zaman, semakin meningkat pula kebutuhan manusia akan informasi. Di tengah kemajuan teknologi informasi yang pesat, internet tidak lagi dapat menjamin keamanan dalam penyediaan informasi (Syawal, et al., 2016). Beragam ancaman dalam dunia digital, seperti peretasan dan pencurian data, menimbulkan kecemasan mengenai keamanan informasi yang sedang dikirim terutama apabila informasi tersebut bersifat rahasia. Kecemasan ini berdampak pada keterlambatan dalam proses pengiriman informasi, sementara informasi tersebut memiliki nilai penting bagi pihak-pihak tertentu (Irvando, et al., 2014).

Teknik kriptografi merupakan salah satu cara untuk menjaga keamanan informasi dengan mengubah data ke dalam bentuk lain yang tidak dapat dibaca maknanya (Humaira, et al., 2023). Salah satu algoritma kriptografi yang memberikan solusi untuk permasalahan keamanan informasi adalah algoritma *Affine Cipher*. Namun, informasi masih dapat diretas dengan mudah karena pada proses enkripsi dan dekripsi menggunakan kunci yang sama (Juliadi & Kusumastuti 2013). Peningkatan keamanan dapat dilakukan dengan menggabungkan dua teknik kriptografi seperti yang dilakukan oleh Fadlan et al. (2017) yang menggabungkan algoritma *Knapsack Merkle Hellman* dan *Affine Cipher*. Meskipun demikian, teknik kriptografi masih memiliki kekurangan di mana dapat dengan mudah menimbulkan kecurigaan (Diana, 2020).

Berbeda dengan kriptografi, fokus utama steganografi adalah menyamarkan keberadaan informasi daripada mengubah isi informasi itu sendiri. Steganografi adalah seni menyembunyikan informasi ke dalam informasi lain (media *cover*). Ada berbagai media yang dapat dipakai sebagai penampung (*cover*) untuk steganografi. *Least Significant Bit* (LSB) merupakan salah satu teknik steganografi. Cara kerja metode LSB adalah dengan menyisipkan bit pesan rahasia ke bit terakhir dari setiap elemen citra digital (Ricky, et al., 2018). Selain untuk menyisipkan pesan rahasia ke dalam citra digital, LSB juga dapat digunakan untuk menyembunyikan pesan ke dalam file audio seperti pada penelitian yang dilakukan oleh Humaira et al. (2022). Namun, LSB ini rentan terhadap serangan karena pesan tersembunyi dapat diungkap dengan mengumpulkan bit terakhir. Oleh karena itu perlu dilakukan modifikasi pada metode ini. Adapun salah satu modifikasi LSB adalah *Least Significant Bit-2* (LSB-2). Penelitian mengenai LSB-2 telah dilakukan oleh Zebua (2015). Menurut Zebua, LSB-2 merupakan hasil modifikasi metode LSB yang bekerja dengan konsep menukarkan bit ke 8-2 (bit ke-6) dari setiap elemen warna *pixel* citra digital yang menjadi citra penampung dengan setiap bit pesan rahasia yang akan disembunyikan.

Keamanan informasi dapat ditingkatkan melalui penggabungan antara teknik kriptografi dan steganografi. Saat teknik kriptografi dan steganografi digabungkan, hasilnya akan menciptakan efek keamanan yang tinggi (Humaira, et al., 2023). Di sini, penulis menggabungkan kriptografi *Affine Cipher* dengan steganografi *Least Significant Bit-2* untuk mengamankan pesan. Hasil penggabungan tersebut dikonstruksi program aplikasi komputer menggunakan bahasa pemrograman Python versi 3.11.

2. METODE

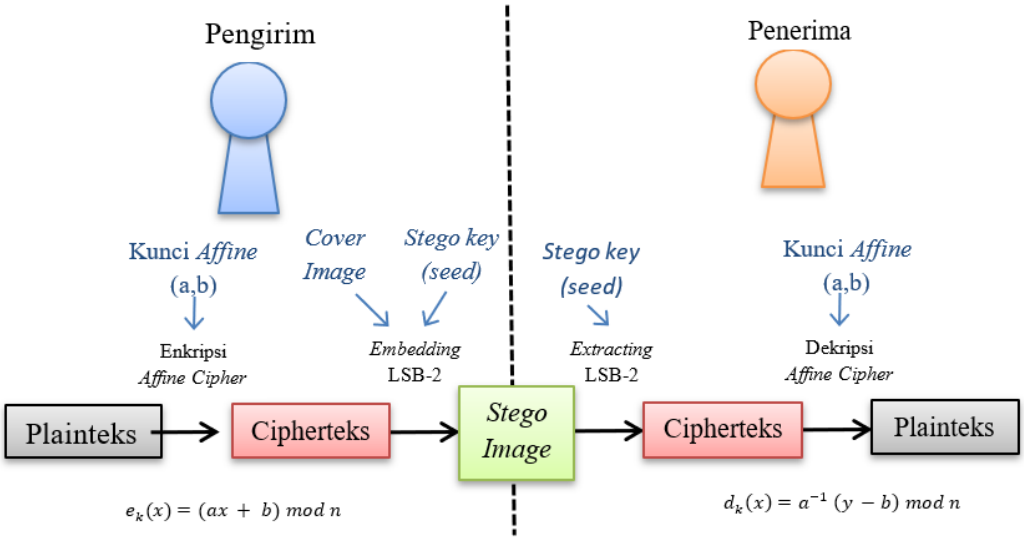
Teknik kriptografi yang digunakan adalah *Affine Cipher* di mana merupakan pengembangan dari *Caesar Cipher*. Pada *Affine Cipher*, plainteks akan dikalikan dengan sebuah nilai dan ditambahkan dengan sebuah pergeseran. Algoritma ini akan digunakan untuk proses enkripsi dan dekripsi (Fadlan & Hadriansa, 2017).

Salah satu teknik steganografi yang terjaga keamanannya adalah *Least Significant Bit-2*, hal itu karena teknik merupakan modifikasi dari *Least Significant Bit* dengan konsep menyembunyikan biner pesan ke bit ke-6 media *cover*, sehingga peretas akan kesulitan mencari di mana keberadaan pesan karena media *cover* terembed informasi tidak akan jauh berbeda dengan media *cover* asli dan tidak bisa langsung dibedakan jika hanya dengan indra penglihatan manusia. Pada penelitian ini, *cover* yang digunakan adalah gambar (Zebua, 2015).

2.1 Model dasar

Sebelum melakukan tahap enkripsi, terlebih dulu dilakukan tahap persiapan pesan/plainteks (*P*) yang akan diinputkan, memilih kunci (*a, b*) di mana *a* merupakan bilangan bulat yang relatif prima dengan *n*, *b* adalah jumlah pergeseran dan *n* adalah ukuran karakter di mana pada penelitian ini terdiri dari 95 karakter serta pilih *seed* dengan ketentuan [1 ukuran gambar]. Selanjutnya masukkan pesan (*P*), kunci (*a, b*), serta *seed* untuk menjadi masukan pada tahap enkripsi dan *embedding*. Pada proses enkripsi dinyatakan ke dalam persamaan berikut $C = (aP + b)\ mod\ n$ (Fadlan & Hadriansa, 2017). Setiap cipherteks (*C*) yang dihasilkan diubah ke dalam bentuk biner ASCII. Kemudian cipherteks yang dihasilkan akan disisipkan pada gambar (*embedding*), sebelum itu dilakukan pemilihan lokasi *pixel* secara acak dengan melakukan pembangkitan bilangan acak menggunakan *Pseudo Random Number Generator* (PRNG) dengan algoritma *Linear Congruential Generator* (LCG) menggunakan *seed* yang sudah ditentukan yang didefinisikan dalam bentuk $X_n = (aX_{n-1} + b)\ mod\ m$ (Djuwitaningrum & Apriyani, 2017). Setelah diperoleh lokasi *pixel*, dilakukan penukaran bit ke-6 dari setiap lokasi *pixel* yang terdiri dari 3 susunan warna (RGB) dengan bit pesan rahasia.

Pada tahap dekripsi dan *extracting*, penerima melakukan pencarian lokasi *pixel* dengan menggunakan *seed* yang sama pada proses *embedding*. Nilai RGB dari setiap lokasi *pixel* diubah ke dalam bentuk biner ASCII yang kemudian diambil setiap bit ke-6. Setelah setiap bit ke-6 terkumpul dikelompokkan menjadi 8 digit yang kemudian dikonversi kembali ke dalam karakter ASCII. Karakter yang sudah diperoleh akan dikembalikan menjadi pesan semula dengan melakukan dekripsi menggunakan *Affine Cipher* dengan rumus $P = a^{-1}\ (C - b)(mod\ n)$ (Fadlan & Hadriansa, 2017). Skema penggabungan dapat dilihat pada Gambar 1.



Gambar 1. Skema Alur Penggabungan Kriptografi *Affine Cipper* dan Steganografi *Least Significant Bit-2* (LSB-2)

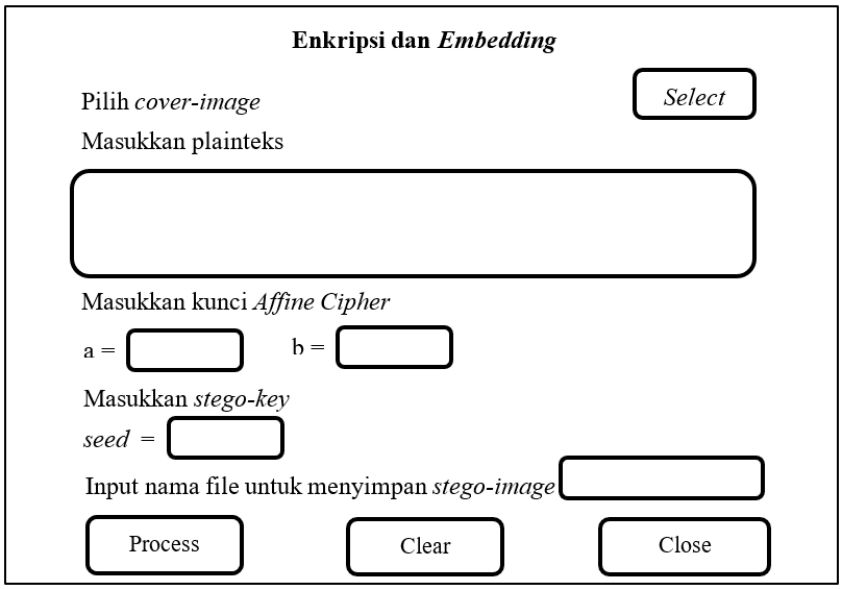
2.2 Konstruksi program aplikasi

Program aplikasi terdiri dari 2 buah program, yang terdiri dari program enkripsi dan *embedding*, dan program dekripsi dan *extracting*. Program enkripsi dan *embedding* berfungsi untuk proses enkripsi pesan menjadi cipherteks dan *embedding* pesan (cipherteks) ke dalam file gambar, program dekripsi dan *extracting* bertujuan untuk mengekstraksi data pada gambar dan mengembalikan kembali pesan (*P*) semula. Rancangan masukan (*input*) serta luaran (*output*) program dapat dilihat pada Tabel 1. *Output* dari program pemilihan kunci adalah list bilangan *a* yang relatif prima terhadap *n*. *Input* dari program enkripsi dan *embedding* adalah pesan (*P*), kunci *Affine Cipher* (*a, b*), *cover image* dan *seed* dengan *output* adalah *stego image*. Input dari program dekripsi dan *extracting* adalah kunci *Affine Cipher* (*a, b*), *stego image* dan *seed*, dengan *output* berupa pesan (*P*) semula.

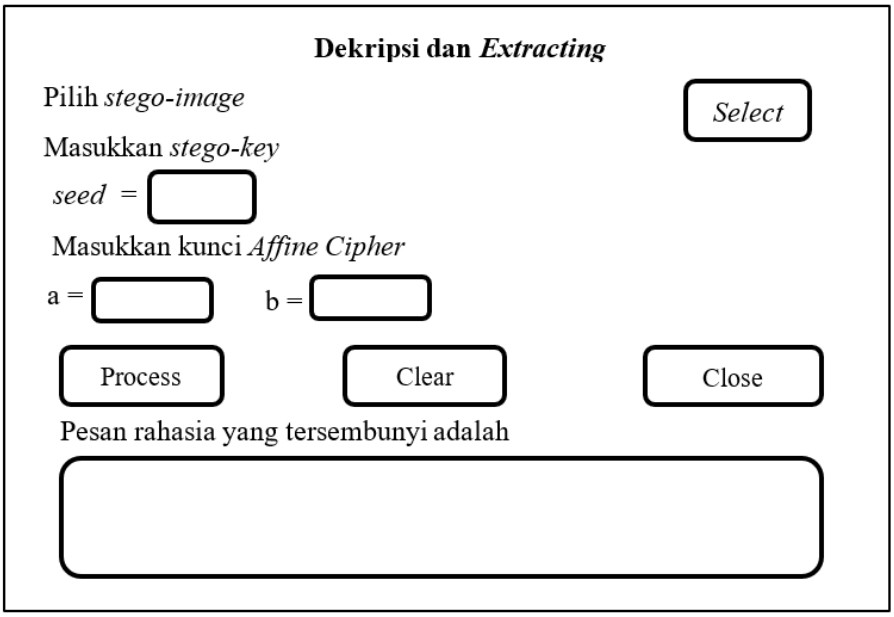
Tabel 1. Rancangan *Input* dan *Output* Program

Keterangan	Enkripsi dan *Embedding*	Dekripsi dan *Extracting*
Input	– Kunci *Affine Cipher* – *Plaintext* – *Cover Image* – *Stego-key (seed)*	– Kunci *Affine Cipher* – *Stego Image* – *Stego-key (seed)*
Output	– *Stego Image*	– *Plaintext*

Rancangan tampilan program terdapat pada Gambar 2 dan Gambar 3.



Gambar 2. Rancangan Tampilan Program Aplikasi Enkripsi dan *Embedding*



Gambar 3. Rancangan Tampilan Program Aplikasi Dekripsi dan *Extracting*

2.3 Tahap Validasi

Pada tahap ini dilakukan validasi terhadap program aplikasi yang dirancang. Program aplikasi tervalidasi jika plainteks dapat diperoleh kembali pada proses *extracting* dan dekripsi serta melakukan pengujian *stego image* yang diperoleh menggunakan *Peak Signal to Noise Ratio* (PSNR) untuk melihat kualitas *stego image*. Untuk menghitung PSNR, perlu dilakukan perhitungan *Mean Squared Error* (MSE) terlebih dahulu. MSE merupakan rata-rata dari selisih kuadrat eror antara *pixel cover image* dan *stego image*.

Perhitungan MSE menggunakan persamaan berikut (Djuwitaningrum & Apriyani, 2017):

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|I(i,j) - K(i,j)|^2$$

dengan:
MSE = Nilai *Mean Square Error* antara *cover image* dengan *stego image*
m = panjang citra tersebut (dalam *pixel*)
n = lebar citra tersebut (dalam $pixel$)
(i,j) = koordinat *pixel*
I = *cover image*
K = *stego image*

Selanjutnya nilai PSNR dihitung dari kuadrat maksimum nilai pixel citra dibagi dengan MSE, yang diberikan dalam bentuk persamaan berikut (Djuwitaningrum & Apriyani, 2017):

$$PSNR = 10.\log\left(\frac{MAX_I^2}{MSE}\right)$$

dengan:
$PSNR$ = nilai PSNR *stego image* (dalam dB)
MAX_I = maksimum nilai *pixel*
MSE = nilai MSE

Kriteria kualitas citra dapat dilihat dari nilai PSNR, seperti pada Tabel 2 (Hidayat & Hastuti, 2013).

Tabel 2. Nilai PSNR Kriteria Kualitas Citra

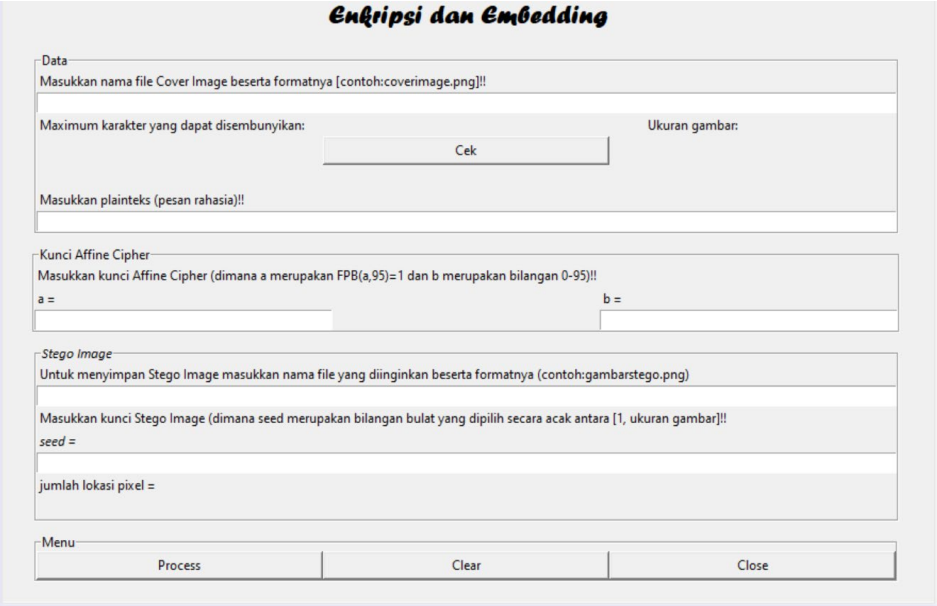
Nilai PSNR	Kualitas Citra
60 dB	*Excellent*, tanpa derau
50 dB	*Good*, terdapat sejumlah derau tapi kualitas citra masih bagus
40 dB	*Reasonable*, terdapat butiran halus atau seperti salju di dalam citra
30 dB	*Poor picture*, terdapat banyak derau
20 dB	*Unusable*

3. HASIL DAN PEMBAHASAN

Program aplikasi dibuat untuk memudahkan *user* dalam melakukan enkripsi dan *embedding* juga untuk mengembalikan pesan dengan *extracting* dan dekripsi dengan kriptografi *Affine Cipher* dan steganografi *Least Significant Bit-2*. Program dibuat menggunakan bahasa pemrograman Python 3.11.

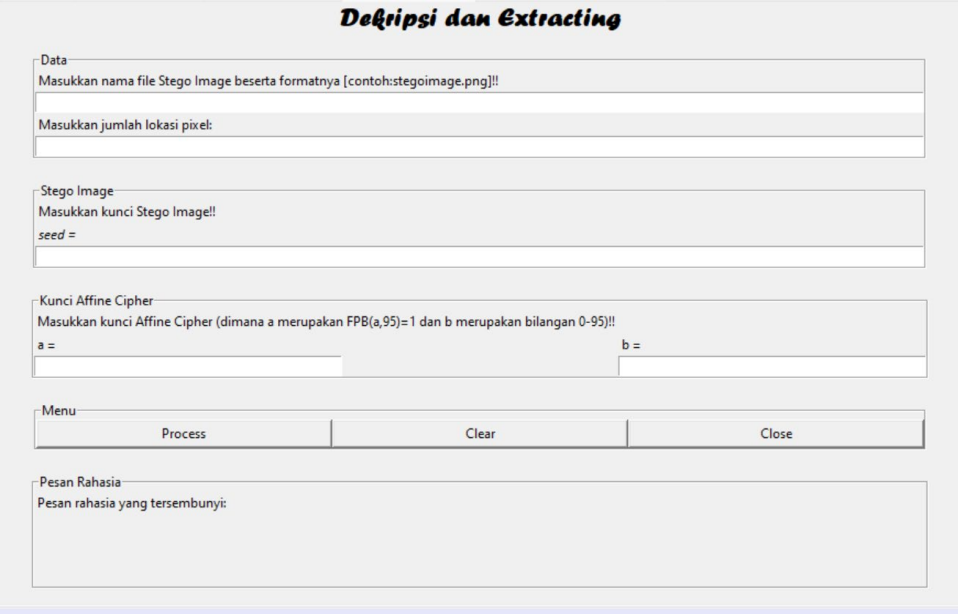
3.1 Tampilan program aplikasi dan penggunaannya

Pada program aplikasi enkripsi dan *embedding user* meng-*input*-kan nama file *cover image*, pesan (*P*), kunci *Affine Cipher* (*a, b*), *cover image*, *seed* dan nama file *stego image*. Terdapat tombol cek untuk melihat berapa karakter maksimum yang dapat disembunyikan serta melihat ukuran *cover image* dan tombol *process* yang berfungsi untuk memulai proses enkripsi dan *embedding* serta informasi mengenai jumlah lokasi *pixel*. Selanjutnya akan dihasilkan *stego image* yang akan tersimpan di folder yang sama dengan program. Tampilan program enkripsi dan *embedding* dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Program Enkripsi dan *Embedding*

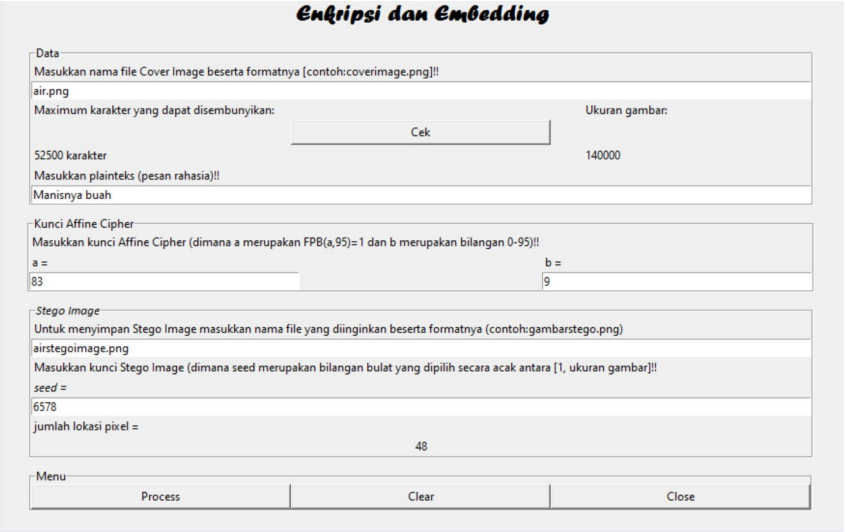
Ketika penerima membutuhkan kembali informasi pesan, penerima dapat menggunakan aplikasi dekripsi dan *extracting*. Masukan pada program ini adalah kunci *Affine Cipher* (*a, b*), nama file *stego image*, *seed* dan lokasi *pixel*. Jika *user* menekan tombol *process* maka proses *extracting* dan dekripsi akan dimulai kemudian hasil pesan yang tersembunyi akan muncul. Tampilan dari program dekripsi dan *extracting* terdapat pada Gambar 5.



Gambar 5. Tampilan Program Dekripsi dan *Extracting*

3.2 Tahap validasi

Validasi program dilakukan dengan membandingkan pesan yang menjadi masukan pada program enkripsi dan *embedding* sama dengan pesan hasil konstruksi pada program dekripsi dan *extracting*. Validasi selanjutnya adalah membandingkan *cover image* dengan *stego image* dengan perhitungan *Peak Signal to Noise Ratio* (PSNR). Validasi pertama dilakukan dengan meng-*input*-kan pesan yaitu “Manisnya buah”, kunci *Affine Cipher* $(a, b) = (83, 9)$, $seed = 6578$, file *cover image* dengan nama file adalah “air.png” dan *stego image* dengan nama file yaitu “airstegoimage.png”. Hasil yang didapatkan adalah pesan hasil konstruksi program dekripsi dan *extracting* sama dengan pesan yang menjadi masukan pada program enkripsi dan *embedding*. Validasi pertama dapat dilihat pada Gambar 6 dan Gambar 7. *Cover image* dan *stego image* dapat dilihat pada Gambar 8 dan Gambar 9.



Gambar 6. Tampilan Masukan pada Program Enkripsi dan *Embedding*

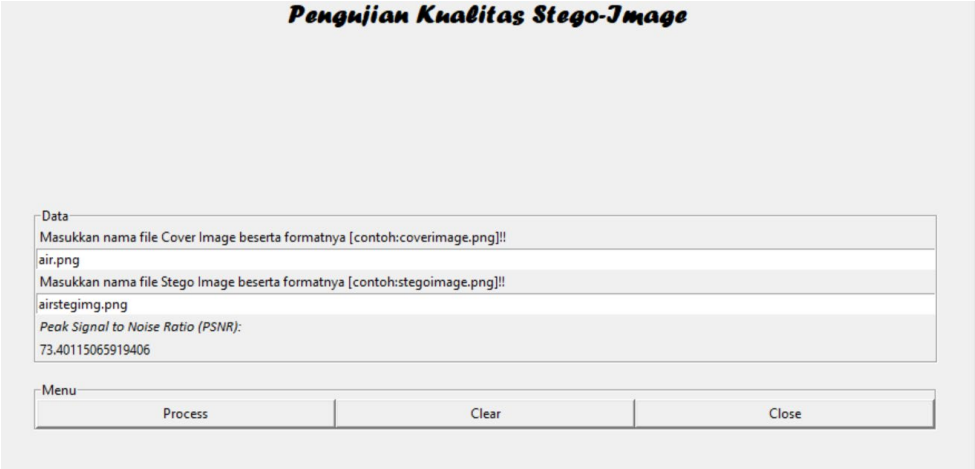
Gambar 7. Tampilan Pesan Hasil Konstruksi pada Program Dekripsi dan *Extracting*

Gambar 8. *Cover image* (product.labodiaprima.co.id)

Gambar 9. *Stego image*

Validasi kedua adalah dengan melihat perbandingan *cover image* dengan *stego image* dengan melakukan perhitungan *Peak Signal to Noise Ratio* (PSNR). Nilai PSNR dihitung dengan

rumus $PSNR = 10.\log\left(\frac{MAX_I^2}{MSE}\right)$ yang mana MAX_I^2 ini merupakan maksimum nilai *pixel* (Djuwitaningrum & Apriyani, 2017). Pada penelitian ini gambar yang digunakan merupakan RGB maka untuk nilai $MAX_I^2 = 255$ dan untuk nilai MSE merupakan nilai rata-rata kuadrat eror antara citra asli (*cover image*) dengan citra hasil penyisipan (*stego-image*). Validasi dilakukan dengan meng-*input*-kan file *cover image* dengan nama file yaitu "air.png" dan *stego image* dengan nama file adalah "airstegoimage.png". Hasil yang didapatkan adalah nilai PSNR sebesar 73,4 dB. Mengacu pada Tabel 2, $nilai\ PSNR > 60\ dB$ di mana termasuk ke dalam kategori *excellent* yang berarti gambar tanpa derau. Validasi kedua dapat dilihat pada Gambar 10.



Gambar 10. Nilai PSNR Kualitas *Stego Image*

Validasi Program dilakukan dengan membandingkan pesan hasil konstruksi pada program dekripsi dan *embedding* dengan pesan yang menjadi masukan pada program enkripsi dan *embedding.* Kedua input tersebut menghasilkan pesan yang sama. Selain itu dilakukan validasi lain, di mana hasil nilai PNSR sesuai dengan ketentuan kriteria kualitas *stego image* yang terdapat pada Tabel 2. Oleh karena itu, program dinyatakan berhasil.

4. KESIMPULAN

Penggabungan teknik kriptografi *Affine Cipher* dengan steganografi *Least Significant Bit-2* (LSB-2) mampu meminimalisir terjadinya kriptanalisis. Pesan akan dienkripsi menggunakan *Affine Cipher* dan disembunyikan secara acak keberadaannya di dalam gambar menggunakan LSB-2 kemudian dikirimkan kepada yang berhak menerimanya. Hal ini menciptakan tingkat kesulitan bagi peretas dalam mencari dan mengakses pesan rahasia yang tersembunyi. Dengan penggunaan teknik ini, privasi komunikasi digital dapat terjaga dengan lebih baik, terutama dalam situasi di mana perlindungan informasi sensitif sangat penting.

Dalam implementasinya, program yang dihasilkan dari penggabungan kriptografi *Affine Cipher* dengan steganografi *Least Significant Bit-2* (LSB-2) menggunakan bahasa pemrograman Python 3.11 dapat mengenkripsi dan *embedding* pesan ke dalam gambar, sehingga *stego image* yang diperoleh tidak terlihat berbeda dengan *cover image*. Hasil dari program dekripsi dan *extracting* adalah dapat merekonstruksi data menjadi pesan semula.

mSelain itu, penelitian selanjutnya dapat dikembangkan untuk mengimplementasikan teknik yang serupa pada media audio dan video. Hal ini akan membantu dalam

menyempurnakan metode steganografi untuk berbagai jenis media digital, sehingga informasi sensitif dapat disembunyikan dengan efektif dalam berbagai format yang berbeda.

5. DAFTAR PUSTAKA

Diana, M. (2020). Implementasi metode GOST (Government Standard) dan LSB-1 (Least Significant Bit) untuk mengamankan teks. *Terapan Informatika Nusantara*, *1*(7), 363-382.

Djuwitaningrum, E. R., & Apriyani, M. (2017). Teknik steganografi pesan teks menggunakan metode least significant bit dan algoritma linear congruential generator. *Jurnal Informatika*, *4*(2), 79-85.

Fadlan, M., & Hadriansa, H. (2017). Rekayasa aplikasi kriptografi dengan penerapan kombinasi algoritma Knapsack Merkle Hellman dan Affine Cipher. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *4*(4), 268-274.

Hidayat, E. Y., & Hastuti, K. (2013). Analisis steganografi metode Least Significant Bit (LSB) dengan penyisipan sekuensial dan acak secara kuantitatif dan visual. *Jurnal Teknologi Informasi*, *12*(3), 157-167.

Humaira, A. F., Marwati, R., & Yulianti, K. (2023). Implementasi kriptografi secret sharing scheme dan steganografi audio Least Significant Bit (LSB). *Jurnal Matematika dan Terapan*, *5*(1), 1-11.

Irvando, I., Purnama, B., & Wijaya, I. S. (2014). Perancangan aplikasi steganografi teknik LSB (Least Significant Bit) dalam keamanan komputer. *Jurnal Processor*, *9*(1), 77-88.

Juliadi, B. P., & Kusumastuti, N. (2013). Kriptografi klasik dengan metode modifikasi Affine Cipher yang diperkuat dengan Vigenere Cipher. *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, *2*(2), 87-92.

Ricky, M., Setyaningsih, F. A., & Dipenogoro, M. (2018). Analisis kompresi steganography pada citra digital dengan menggunakan metode least significant bit berbasis mobile android. *Coding Jurnal Komputer dan Aplikasi*, *6*(3), 75-86.

Syawal, M. F., Fikriansyah, D. C., & Agani, N. (2016). Implementasi teknik steganografi menggunakan algoritma Vigenere Cipher dan metode LSB. *Jurnal TICom*, *4*(3), 93707.

Zebua, T. (2015). Penerapan metode LSB-2 untuk menyembunyikan ciphertext pada citra digital. *Pelita Informatika: Informasi dan Informatika*, *10*(3), 135-137.