

# IMPLEMENTASI *STEGANOGRAFI* PENYEMBUNYIAN PESAN PADA GAMBAR MENGGUNAKAN METODE EOF DENGAN ENKRIPSI RSA BERBASIS ANDROID

MOHAMMAD HADI NUR AINI

Teknik Informatika, Fakultas Teknik  
Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia  
e-mail : m-hadi@student.umaha.ac.id

## ABSTRAK

Pada era serba digital ini keamanan data atau pesan sudah menjadi hal yang sangat dibutuhkan. Pengamanan data bisa dilakukan menggunakan cara enkripsi ataupun dengan steganografi. Kriptografi bertujuan untuk mengamankan pesan dengan mengubahnya menjadi tidak bisa dimengerti tanpa teknik tertentu. Steganografi bertujuan untuk menyembunyikan pesan sehingga pesan tersebut tidak diketahui keberadaannya oleh orang lain. Steganografi merupakan teknik yang menarik dan cukup efektif untuk tujuan merahasiakan dan mengamankan pesan. Pada penelitian ini algoritma yang digunakan untuk enkripsi adalah algoritma RSA dan pengamanan pesan akan dikombinasikan dengan steganografi menggunakan metode *End Of File*. Pesan rahasia tersebut akan terlebih dahulu dienkripsi kemudian dilakukan proses penyisipan ke dalam suatu gambar. Penelitian ini menghasilkan suatu aplikasi yang berfungsi untuk keamanan pesan teks menjadi teks yang terenkripsi dan tersembunyi di dalam suatu gambar.

**Kata kunci:** dekripsi, enkripsi, eof, kriptografi, rsa, steganografi

## PENDAHULUAN

Manusia merupakan makhluk sosial yang diciptakan saling berkomunikasi satu sama lain, komunikasi merupakan bagian yang sangat penting di dalam kehidupan manusia, cara berkomunikasi dapat dilakukan dengan pertukaran informasi dengan berbicara secara langsung. Informasi atau pesan yang bersifat sangat rahasia menjadi masalah yang cukup serius apabila dilakukan melalui suatu media karena ancaman akan kebocoran pesan rahasia tersebut terhadap orang yang tidak berwenang, oleh karena itu suatu teknik sangat diperlukan untuk menjaga keamanan dan kerahasiaan pesan tersebut. Salah satu cara yang bisa digunakan untuk mengamatkannya adalah kriptografi. Kriptografi merupakan bidang ilmu yang mempelajari keamanan data. Namun terkadang pihak luar yang mengetahuinya bisa dengan sengaja merusak pesan yang telah diacak tersebut dengan tujuan agar pesan tersebut tidak sampai kepada tujuan utamanya. Untuk mengatasi hal ini maka dapat digunakan ilmu steganografi. Steganografi merupakan ilmu untuk menyembunyikan atau menyisipkan data pada suatu media tertentu sehingga pesan tersebut secara kasat mata tidak diketahui keberadaannya. Dengan adanya penyisipan ini maka pihak lain tidak akan mengetahui adanya suatu pesan rahasia yang sudah disisipkan sehingga tidak menimbulkan suatu kecurigaan.

Salah satu bentuk media yang dapat digunakan untuk penyisipan pesan adalah berupa gambar/citra digital. Teknik yang bisa diterapkan untuk menyisipkan pesan ke dalam citra digital salah satunya adalah dengan metode EOF.

Dengan metode EOF ini pesan rahasia akan disisipkan di akhir *file*/akhir citra. Untuk mendukung keamanan daripada pesan rahasia tersebut maka digunakan suatu metode enkripsi dengan algoritma RSA. Metode enkripsi ini dipilih karena keamanannya yang kuat dikarenakan enkripsi ini termasuk salah satu jenis enkripsi asimetris dimana kunci enkripsi akan berbeda dengan kunci untuk dekripsinya.

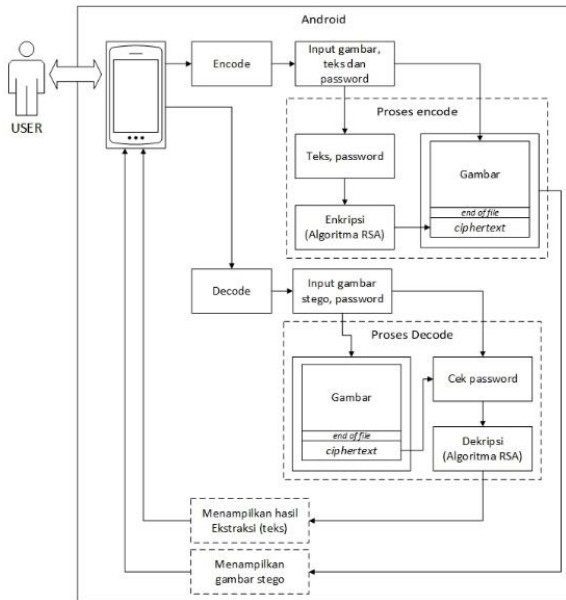
Membangun aplikasi yang dapat mengenkripsikan pesan/teks menjadi teks yang acak kemudian menyisipkan *ciphertext* tersebut ke dalam suatu citra digital yang bisa disebut sebagai *cover image* sehingga menjadi *stego image*. Dengan menggunakan enkripsi dan penyembunyian pesan pada citra maka pesan akan lebih aman dari pihak yang tidak diinginkan. Mendapatkan aplikasi android yang menggabungkan algoritma RSA dengan teknik steganografi EOF.

## PERANCANGAN SISTEM

### Blok Diagram

Perancangan blok diagram berikut bertujuan sebagai acuan untuk pembuatan sistem

sehingga dapat menerangkan cara kerja dan alur sistem secara garis besar. Di dalam blok diagram bagian-bagian atau fungsi utama diwakili oleh blok-blok yang saling terhubung dengan garis dan panah sebagai penunjuk urutan proses. Dengan blok diagram maka sebuah sistem akan dapat lebih mudah dipahami dan dimengerti. Blok diagram dari sistem yang dibuat oleh peneliti ditunjukkan Gambar 1.



Gambar 1 Blok Diagram Sistem

### Citra Digital

Citra digital merupakan matriks yang terdiri atas kolom dan juga baris dimana setiap pasangan indeks dari baris dan indeks dari kolom menyatukan nilai suatu piksel pada citra. Nilai matriks tersebut menentukan tingkat kecerahan dari piksel tersebut. Citra digital pada umumnya dipresentasikan sebagai matriks ( $n \times m$ ), dimana elemen matriks tersebut adalah piksel [1].

### Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *Cryptos* yang artinya Secret (rahasia), dan *Graphain* yang artinya *Writing* (tulisan). Jadi, kriptografi memiliki arti *Secret Writing* atau tulisan rahasia. Definisi yang digunakan di dalam buku sebelum tahun 1980-an yang menyatakan bahwa kriptografi adalah ilmu dan juga seni untuk menjaga keamanan dan kerahasiaan data atau pesan dengan cara mengacaknya menjadi suatu bentuk yang sulit dimengerti lagi maknanya [2].

### Algoritma RSA

Algoritma RSA, ditemukan oleh 3 orang dari MIT (Massachusetts Institute of Technology) pada tahun 1976, yaitu: Ron Rivest, Adi Shamir, dan Leonard Adleman. RSA merupakan salah satu *public key crypto system* yang sangat sering

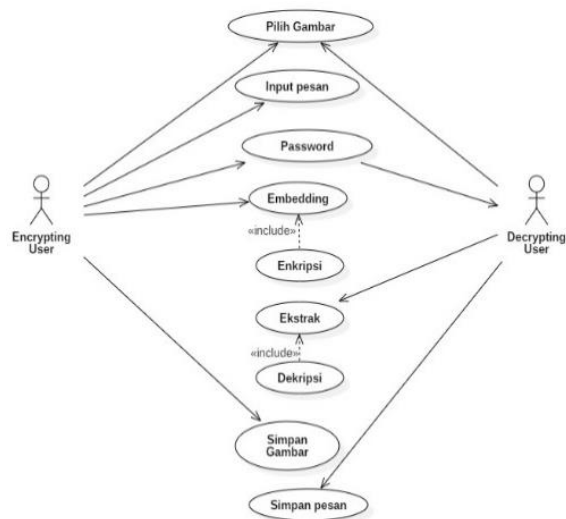
digunakan untuk memberikan kerahasiaan terhadap keaslian suatu data digital. Keamanan enkripsi dan dekripsi data model ini terletak pada kesulitan untuk memfaktorkan modulus  $n$  yang sangat besar [3].

### Metode End Of File (EOF)

Menurut [4] menyatakan bahwa teknik EOF atau *End Of File* adalah termasuk teknik yang digunakan untuk steganografi. Proses dengan teknik ini adalah dengan menyisipkan data atau pesan di akhir suatu file. Teknik ini bisa diterapkan untuk menyisipkan suatu data yang ukurannya dapat disesuaikan. Ukuran file yang sudah disisipi data masih sama dengan file sebelum disisipi data ditambah ukuran data yang disisipkan.

### Use Case Diagram

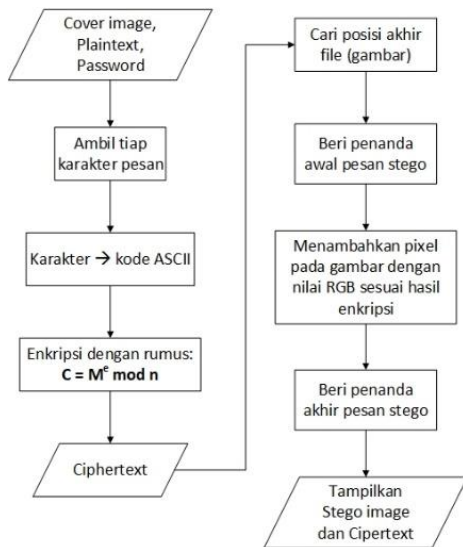
Pada perancangan menggunakan *use case diagram*, terdapat dua jenis *user* yang dapat menggunakan aplikasi ini. Pengguna yang pertama adalah *user* yang ingin menyisipkan atau menyembunyikan pesan ke dalam gambar. *User* ini akan dapat melakukan aksi seperti memilih gambar, memasukkan pesan, menambahkan *password*, menyisipkan pesan, dan menyimpan gambar. Sedangkan pengguna yang kedua adalah *user* yang ingin mengekstraksi pesan dalam gambar yang sudah disisipi pesan oleh jenis *user* pertama. Diagram *use case* dari sistem ini ditunjukkan Gambar 2.



Gambar 2. Use Case Diagram

### Algoritma Enkripsi beserta Encode

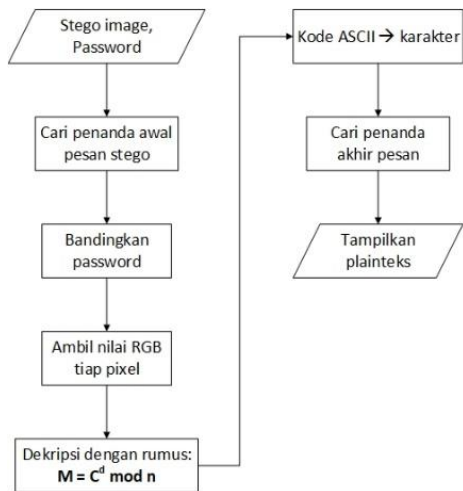
Pada proses *encode*/penyisipan pesan ke dalam gambar, maka sistem akan secara otomatis mengenkripsikan pesan dan *password* terlebih dahulu sebelum menyisipkannya ke dalam gambar. Langkah-langkah untuk menyisipkannya ditunjukkan Gambar 3.



Gambar 3. Langkah-langkah Enkripsi beserta Encode

**Algoritma Decode beserta Dekripsi**

Pada proses *decode*/pengekstrakan pesan dari gambar, sistem akan terlebih dahulu mengecek *password* apakah benar. Apabila *password* yang dimasukkan benar maka sistem akan mengekstrak pesan dari gambar kemudian mendekripsikannya kembali. Langkah-langkah untuk mengekstraknya ditunjukkan Gambar 4.



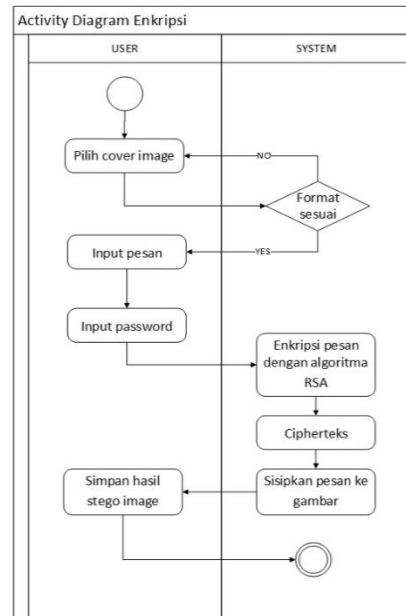
Gambar 4. Langkah-langkah Decode beserta Dekripsi

**Activity diagram enkripsi**

*Activity diagram* enkripsi menggambarkan alir aktivitas enkripsi yang dilakukan oleh *user* yang ingin menyisipkan pesannya ke dalam gambar. Alir tersebut ditunjukkan Gambar 5.

Gambar 5 menunjukkan aktivitas *user* yang ingin menyisipkan pesannya ke dalam gambar. Ketika *user* memilih menu *Decode* maka sistem akan menampilkan halaman untuk penyisipan pesan. Pada aktivitas ini hal yang pertama kali yang harus dilakukan oleh *user* adalah memilih gambar sebagai *cover image*, selanjutnya *user* harus memasukkan pesannya, kemudian memberi *password*, untuk pemberian *password* ini dapat

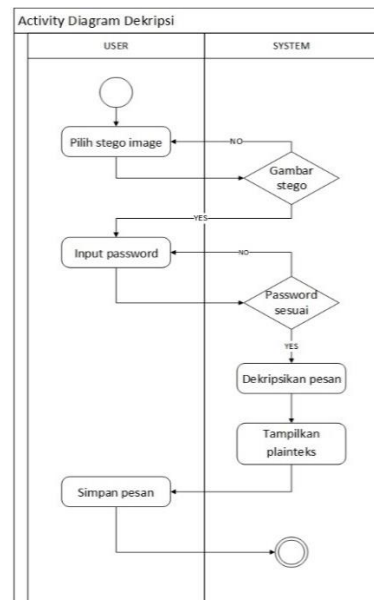
diabaikan karena tidak diwajibkan. Setelah semua *input* terisi langkah selanjutnya *user* harus menekan tombol proses dan menunggu sistem selesai menyisipkan pesan ke dalam gambar.



Gambar 5. Activity Diagram Enkripsi

**Activity diagram dekripsi**

*Activity diagram* dekripsi menggambarkan alir aktivitas dekripsi yang dilakukan oleh *user* yang ingin mengekstrak pesan yang terdapat pada *stego image*. Alir tersebut ditunjukkan Gambar 6.



Gambar 6. Activity Diagram Dekripsi

Gambar 6 menunjukkan aktivitas *user* yang ingin mengekstrak pesan, pertama *user* harus memilih gambar yang mengandung pesan rahasia, apabila pesan rahasia tersebut mengandung *password* maka *user* harus memasukkan *password* yang sesuai kemudian menekan tombol proses dan

menunggu sampai sistem selesai mengekstrak pesan.

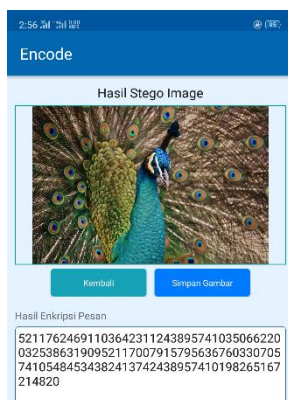
## IMPLEMENTASI DAN PEMBAHASAN

### Tampilan Menu Utama

Pada saat aplikasi dibuka yang akan pertama kali tampil adalah tampilan menu utama. Bagian atas tampilan merupakan nama dari aplikasi ini, kemudian di bagian bawah terdapat empat pilihan menu yang dapat dipilih oleh *user*. Pada tampilan ini *user* dapat memilih menu-menu yang terdapat di dalam aplikasi ini, sehingga fungsional dari aplikasi ini bisa dilakukan melalui pilihan menu yang tersedia. Tampilan menu utama ditunjukkan Gambar 7.



Gambar 7. Tampilan menu utama



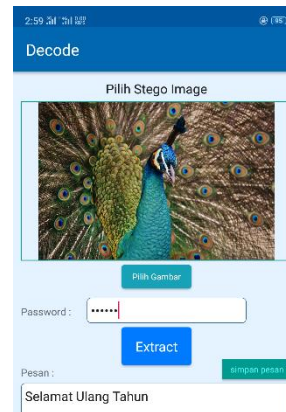
Gambar 8. Hasil Proses *Encoding*

### Proses *Decoding*

Proses *decoding* merupakan proses yang bertujuan untuk mengekstraksi kembali pesan yang terdapat di dalam *stego image*. Proses ini dapat berjalan pada menu *decode*. Untuk dapat melakukan proses *decoding* ini, terlebih dahulu

*user* harus memilih *stego image* dengan cara menekan tombol “Pilih gambar” yang terdapat pada tampilan menu *decode* ini. Setelah *user* memilih gambar yang dikehendaki maka sistem akan terlebih dahulu mengecek gambar tersebut apakah gambar yang telah terpilih terdapat pesan tersembunyi.

Setelah dekripsi pesan berhasil, maka proses *decoding* ini akan diakhiri dengan menampilkan pesan tersebut pada bagian bawah tampilan menu *decode* seperti yang terlihat pada Gambar 9.



Gambar 9. Tampilan menu *decode*

### Akurasi

Untuk menghitung nilai akurasi sistem ini maka penulis melakukan uji coba sebanyak 10 kali terhadap sistem yang telah dibangun, setiap uji coba akan dilakukan dengan memberi pesan dengan jumlah karakter yang berbeda-beda, hasil dari uji coba ditunjukkan Tabel 1.

Tabel 1. Pengujian Akurasi

Pesan asli	Pesan hasil dekripsi & ekstraksi	Akurasi
UMAHA	UMAHA	100%
Selamat Pagi	Selamat Pagi	100%
Pesan rahasia	Pesan rahasia	100%
Kriptografi	Kriptografi	100%
Citra digital	Citra digital	100%
Indonesia	Indonesia	100%
Semoga berhasil	Semoga berhasil	100%
Steganografi	Steganografi	100%
Teknik Informatika	Teknik Informatika	100%
0123456789	0123456789	100%

Berdasarkan hasil uji coba pada Tabel 1 terlihat bahwa seluruh hasil uji coba menunjukkan nilai akurasi 100%, seluruh pesan yang telah terenkripsi dan tersisipkan ke dalam gambar/*cover image* dapat didekripsi dan diekstrak kembali secara sempurna menjadi pesan semula, bentuk pesan maupun panjang pesan sebelum disisipkan masih sama dengan pesan hasil ekstraksi. Jumlah karakter atau panjang pesan yang disisipkan tidak berpengaruh terhadap nilai akurasi. Hal ini menunjukkan bahwa dalam keadaan tanpa adanya gangguan, sistem steganografi yang disertai







dengan enkripsi ini mempunyai tingkat akurasi sebesar 100%.

**Pengujian Kriteria Steganografi**

**Fidelity**

Yang dimaksud dengan kriteria *fidelity* ini adalah mutu atau kualitas dari citra penampung yang digunakan tidak jauh berubah setelah proses penyisipan pesan dilakukan. Citra hasil dari proses *encoding* masih terlihat dengan baik, dan apabila dilihat oleh orang lain mereka tidak menyadari bahwa pada citra tersebut telah disisipi pesan rahasia. Tabel 2 merupakan hasil dari pengujian kriteria *fidelity*.

Tabel 2. Hasil pengujian kriteria *fidelity*

Gambar asli	Gambar <i>stego image</i>	Responden
 Format: PNG Dimensi: 960x640 <i>pixel</i>	 Format: PNG Dimensi: 960x641 <i>pixel</i>	5 orang tidak melihat
 Format: PNG Dimensi: 800x600 <i>pixel</i>	 Format: PNG Dimensi: 800x601 <i>pixel</i>	5 orang tidak melihat
 Format: BMP Dimensi: 600x375 <i>pixel</i>	 Format: PNG Dimensi: 600x376 <i>pixel</i>	5 orang tidak melihat
 Format: PNG Dimensi: 500x313 <i>pixel</i>	 Format: PNG Dimensi: 500x314 <i>pixel</i>	5 orang tidak melihat
 Format: JPG Dimensi: 400x225 <i>pixel</i>	 Format: PNG Dimensi: 400x226 <i>pixel</i>	5 orang tidak melihat

Berdasarkan hasil pengujian pada tabel 2 diperoleh bahwa gambar asli dengan gambar *stego image* tidak tampak perbedaannya bahkan seluruh gambar tersebut terlihat sama di mata responden. Sehingga berdasarkan hasil pengujian tersebut bisa didapat suatu kesimpulan bahwa kriteria *fidelity* ini telah terpenuhi.



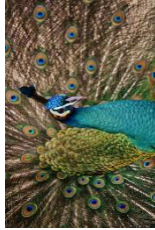



**Robustness**

*Robustness* merupakan kriteria dimana pesan yang terkandung di dalam *stego image* tetap utuh meski telah dilakukan manipulasi terhadap

gambar *stego image*. Untuk dapat mengetahui apakah pada sistem ini hasil dari gambar *stego image* memenuhi kriteria *robustness* maka dilakukan pengujian dengan menyisipkan pesan terlebih dahulu ke dalam gambar, kemudian hasil dari *stego image* tersebut dilakukan beberapa manipulasi dan setelah itu dilakukan proses ekstraksi kembali.

Pengujian dilakukan pada sampel gambar "Burung Merak.png" dengan format gambar png dan disisipkan pesan "Selamat Ulang Tahun" serta *password* "123456". Setelah berhasil disisipkan hasil *stego image* akan dimanipulasi kemudian dilakukan ekstraksi kembali. Hasil pengujian kriteria *robustness* dapat ditunjukkan pada Tabel 3.

Tabel 3. Hasil pengujian kriteria *robustness*

Gambar hasil manipulasi <i>stego image</i>	Jenis manipulasi	Hasil
	<i>Brightness</i>	NULL
	<i>Contrast</i>	NULL
	<i>Rotate 90° CCW</i>	NULL
	<i>Resize</i>	NULL
	<i>Crop</i>	NULL
	<i>Grayscale</i>	NULL

Berdasarkan hasil pengujian pada Tabel 3 sistem tidak berhasil mengekstraksi kembali pesan yang terdapat pada gambar *stego image* yang telah dilakukan manipulasi, sehingga dapat diambil kesimpulan bahwa aplikasi ini tidak mendukung kriteria *robustness*, hal ini dikarenakan pesan yang tersimpan pada gambar dengan format ASCII sangat rentan terhadap perubahan nilai *pixel* pada

*stego image* yang menjadi tempat penyisipan pesan.

### Recovery

Kriteria *recovery* akan dikatakan berhasil atau terpenuhi apabila hasil pesan ekstraksi masih sama persis jika dibandingkan dengan pesan yang belum disisipkan sebelumnya, mulai dari karakter setiap kata sampai panjang dari pesan tersebut. Pengujian ini dilakukan dengan cara setiap gambar akan disisipi pesan dengan kata yang sama. Adapun pesan yang akan disisipkan penulis telah menyediakan sebanyak lima pesan dengan panjang yang berbeda-beda yakni pesan dengan panjang 50 karakter, 100 karakter, 200 karakter, 500 karakter, dan 1000 karakter.

Tabel 4. Hasil pengujian kriteria *recovery*

Panjang pesan sebelum penyisipan	Panjang pesan setelah ekstraksi	status
50 karakter	50 karakter	Berhasil
100 karakter	100 karakter	Berhasil
200 karakter	200 karakter	Berhasil
500 karakter	500 karakter	Berhasil
1000 karakter	1000 karakter	Berhasil

Melihat hasil percobaan pada Tabel 4 didapatkan hasil bahwa dari lima gambar berbeda yang masing-masing telah disisipkan lima jenis pesan yang berbeda secara bergantian, hasil dari ekstraksi kembali pesan menunjukkan bahwa pesan sebelum disisipkan mempunyai panjang yang sama dengan pesan hasil ekstraksi, sehingga dapat disimpulkan bahwa pada sistem ini hasil *stego image* telah memenuhi kriteria *recovery*.

### PENUTUP

Kunci enkripsi dari algoritma RSA berbeda dengan kunci untuk dekripsinya. Pada sistem ini pesan teks rahasia akan dienkripsi kemudian hasil dari enkripsi tersebut berupa *ciphertext* dalam bentuk bilangan bulat (*integer*) dan kemudian disembunyikan ke dalam gambar dengan format

PNG sehingga kerahasiaan pesan terjaga karena orang lain tidak menyadarinya.

Hasil dari proses steganografi menggunakan metode EOF ini tidak mengubah kualitas dari citra aslinya, bahkan citra *stego image* yang dihasilkan masih terlihat sama dengan citra *cover image*. Metode EOF ini masih memiliki sifat mudah rusak oleh serangan (*robustness*) oleh manipulasi citra seperti *rotate*, *crop*, *brightness* dan *contrast*.

Perancangan yang telah dibuat oleh penulis dapat digunakan sebagai pembelajaran terutama untuk hal-hal yang berkaitan dengan keamanan pesan. Media yang digunakan untuk penyembunyian ini adalah citra digital, sehingga untuk pengembangannya bisa digunakan media lain seperti audio atau video.

### DAFTAR PUSTAKA

- [1] H. Manurung, "Teknik Penyembunyian Pesan Teks Pada Media Citra Gif Dengan Metode Least Significant Bit (LSB)," no. 0911765, pp. 62-68, 2014.
- [2] M. Husein, "Implementasi Caesar Cipher untuk Penyembunyian Pesan Teks Rahasia pada Citra dengan Menggunakan Metode Least Significant Bit," *Pelita Inform. Budi Darma*, vol. VII, no. 2, pp. 116-122, 2014.
- [3] M. Ihwani, "Model Keamanan Informasi Berbasis Digital Signature Dengan Algoritma Rsa," *CESSJournal Comput. Eng. Syst. Sci.*, vol. 1, no. 1, pp. 15-20, 2016.
- [4] Y. Aditya, A. Pratama, and A. Nurlifa, "Studi pustaka untuk steganografi dengan beberapa metode," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2010.
- [5] B. C. Putra and Y. N. Afifah, "Gaussian Mixture Model untuk Penghitungan Tingkat Kebersihan Sungai Berbasis Pengolahan Citra," *Tek. Eng. Sains J.*, vol. 2, no. 1, pp. 53-58, 2018.