IMPLEMENTASI ALGORITMA *END OF FILE* (EoF) PADA STEGANOGRAFI CITRA

Minarni[1)], Andre Gesta Fernando[2)]
Jurusan Teknik Informatika
Fakultas Teknik
Institut Teknologi Padang
minarni1706@gmail.com, andregestaf@gmail.com

Abstract

In maintaining the confidentiality of information or data from sabotage, thieves or from unauthorized parties, can take a step by hiding messages on a media that can be said by steganography. One of the media that can be used in hiding messages is GIF image, where GIF image has a good ability to insert messages, because the GIF image itself does not experience changes after the message is inserted because the GIF image is lossless.

The End Of File (EoF) algorithm is one of the algorithms used in steganography by adding data or secret messages at the end of the file. Calculation of file size that has been inserted data is the same as the size of the file before inserting data plus the size of the secret data that has been changed to encoding the file. The EoF algorithm has the advantage of being able to hide an unlimited number of messages.

Based on the results of the tests carried out, the GIF that has been inserted messages from 50 characters to 100000 characters has not changed. Based on the steganography criteria with EoF algorithm recovery testing is good done on GIF images, testing the GIF image robustness is not resistant to manipulation, and for recovery message testing can be returned properly without manipulation on the stego GIF image.

Keyword *- implementation, end of file, steganography, GIF image*

Intisari

Dalam menjaga kerahasiaan informasi atau data dari sabotase, pencuri atau dari pihak yang tidak berwenang, dapat dilakukan dengan menyembunyikan pesan pada sebuah media yang dikenal dengan steganografi. Salah satu media yang dapat digunakan dalam menyembunyikan pesan yaitu citra GIF, dimana citra GIF memiliki kemampuan yang bagus dalam menyisipkan pesan, karena citra GIF sendiri tidak mengalami perubahan setelah disisipkan pesan karena citra GIF yang bersifat lossless.

Algoritma End Of File (EoF) merupakan salah satu algoritma yang digunakan dalam steganografi dengan cara menambahkan data atau pesan rahasia pada akhir file. Perhitungan ukuran file yang sudah disisipkan data sama dengan ukuran file sebelum disisipkan data ditambah ukuran data rahasia yang telah diubah mejadi encoding file. Algoritma EoF mempunyai kelebihan dapat menyembunyikan pesan dalam jumlah yang tidak terbatas.

Berdasarkan hasil pengujian yang dilakukan pada GIF yang telah disisipkan pesan dari 50 karakter sampai 100000 karakter tidak mengalami perubahan. Berdasarkan kriteria steganografi dengan algoritma EoF pengujian recovery bagus dilakukan pada citra GIF, pengujian robustness citra GIF tidak tahan terhadap manipulasi, dan untuk pengujian recovery pesan dapat dikembalikan dengan baik tanpa dilakukan manipulasi pada citra stego GIF.

Kata kunci *- implementasi, end of file, steganografi, citra GIF*

1. PENDAHULUAN

Steganografi adalah seni menyembunyikan pesan teks sedemikian rupa sehingga tak seorangpun selain pegirim dan penerima yang dituju mengetahui keberadaan informasi tersebut [1]. Secara umum teknik steganografi menggunakan *redundant bits* sebagai tempat menyembunyikan pesan pada saat dilakukan kompresi data, dan kemudian menggunakan kelemahan indera manusia yang tidak *sensitive* sehingga pesan tersebut tidak ada perbedaan yang terlihat [2]. Steganografi dapat digunakan sebagai *tag-notes* untuk citra *online*,

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

steganografi juga dapat digunakan untuk melakukan penyimpanan atas kerahasiaan informasi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotasi, pencuri, atau dari pihak yang tidak berwenang. Steganografi juga dapat digunakan untuk alasan yang ilegal misalnya seseorang telah mencuri data, mereka dapat menyembunyikan arsip curian tersebut ke dalam arsip lain dan mengirimkannya keluar tanpa menimbulkan kecurigaan siapapun karena tampak seperti *email* atau arsip normal.

Dalam steganografi terdapat beberapa algoritma yang digunakan yaitu *Least Significant Bit* (LSB), *GifShuffle*, *End Of File* (EoF). LSB merupakan algoritma terapan dari algoritma substitusi. Teknik ini menggantikan bit yang paling akhir dari data asli dengan bitpesan. Dasar dari algoritma ini adalah bilangan berbasiskan biner atau dengan kata lain angka 0 dan angka 1. *GifShuffle* adalah sebuah algoritma *steganography* yang digunakan untuk menyembunyikan pesan dalam berkas citra dengan format GIF. Sesuai dengan namanya *GifShuffle* akan melakukan "*Shuffle*" terhadap palet warna dari sebuah berkas gif. "*Shuffle*" jika diterjemahkan ke dalam bahasa Indonesia berarti memutar. Sehingga dapat diartikan bahwa *GifShuffle* adalah algoritma yang memanfaatkan penukaran posisi ke 256 palet warna dalam berkas citra berformat GIF. Hal tersebut aman dilakukan karena dua buah berkas GIF dengan palet warna yang berbeda akan ditampilkan secara sama persis.

Algoritma EoF merupakan pengembangan dari algoritma LSB, dimana algoritma LSB merupakan algoritma terapan dari algoritma substitusi. Teknik ini menggantikan bit yang paling akhir dari data asli dengan bit pesan. Sehingga pesan yang disisipkan terbatas sedangkan algoritma EoF menambahkan data atau pesan pada akhir *file*. Pada algoritma EoF perhitungan ukuran *file* yang telah disisipkan data sama dengan ukuran *file* sebelum disisipkan data ditambah ukuran data rahasia yang telah diubah menjadi *encoding file.* Dari ketiga algoritma diatas algoritma EoF memiliki kelebihan dari kedua algoritma LSB dan *GifShuffle* yaitu dapat menyisipkan pesan yang lebih banyak. Pada algoritma LSB pesan yang disisipkan terbatas sedangkan pada *GifShuffle* hanya dapat disisipkan pesan dengan ukuran yang relatif lebih kecil yaitu 209 *byte* [3].

Beberapa penelitian tentang steganografi telah dilakukan oleh beberapa peneliti, diantaranya penelitian tentang Implementasi Steganografi menggunakan Algoritma EoF, membahas tentang penyisipan pesan pada gambar dengan *input* dan ekstraksi pesan menggunakan algoritma EoF. Penelitian tentang Pengamanan *File* Multimedia dengan Algoritma Steganografi EoF Untuk Menjaga Kerahasiaan Pesan, penelitian ini membahas tentang penyisipan pesan pada *file* multimedia [4], [5], [6], [8]. Selain EoF terdapat algoritma *Least Significant Bit* (LSB) yang dapat digunakan sebagai metode steganografi [9]. Penggunaan algoritma *giftshuffle* agar dapat menyisipkan pesan lebih banyak dengan menggunakan *cover image* pada citra berformat GIF [3], [10].

Penelitian ini bertujuan merancang dan menganalisa algoritma steganografi EoF untuk mengamankan pesan dalam citra GIF.

2. METODOLOGI

2.1 Landasan Teori

a. Steganografi

Steganografi (*steganography*) adalah teknik menyembunyikan data rahasia di dalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang. Steganografi membutuhkan dua properti yaitu, wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara (audio), teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video. Penggunaan steganografi antara lain bertujuan untuk menyamarkan eksistensi (keberadaan) data rahasia sehingga sulit dideteksi, dan melindungi hak cipta suatu produk. Steganografi dapat dipandang sebagai kelanjutan kriptografi. Jika pada kriptografi, data yang telah disandikan (*chiphertext*) tetap tersedia, maka dengan steganografi *cipherteks* dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Data rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti keadaan aslinya [1].

Kriteria yang harus diperhatikan dalam penyembunyian data adalah:

1. *Fidelity*.

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.

2. *Robustness*.
   Data yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada citra penampung, seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, pemotongan (*cropping*), enkripsi, dan sebagainya. Bila pada citra penampung dilakukan operasi-operasi pengolahan citra tersebut, maka data yang disembunyikan seharusnya tidak rusak (tetap valid jika diekstraksi kembali)
3. *Recovery*.
   Data yang disembunyikan harus dapat diungkapkan kembali (*reveal*). Karena tujuan steganografi adalah data hiding, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut [11].

b. Algoritma End of File (EoF)

Implementasi algoritma EoF dilakukan dalam dua tahapan yaitu *encoding* dan *decoding*. Adapun langkah-langkah *encoding* menggunakan algoritma EoF adalah sebagai berikut [8] :

1. Proses *encoding* dimulai dengan menghitung nilai *pixel* yang menjadi objek steganografi dan akan menghasilkan bilangan *decimal*.
2. Mengubah pesan yang akan disisipkan menjadi bilangan bentuk decimal.
3. Memberi penanda pesan di awal pesan dan di akhir pesan.
4. Menyisipkan pesan di akhir *file*.
5. Kemudian *pixel* yang sudah disisipi akan dikompresi ulang dengan palet warna yang baru, untuk menghasilkan berkas yang baru/*stegoimage*.

Langkah- langkah proses *decoding* atau mengekstrak pesan dari citra yang telah disisipi pesan dengan algoritma EoF adalah sebagai berikut:

1. Menghitung *pixel* citra yang sudah disisipi pesan kemudian di ubah menjadi bilangan *decimal*.
2. Melakukan pencarian pesan penanda yang telah disisipkan dalam *stegoimage*
3. Pesan *decimal* di ambil dan dipisahkan dengan penanda
4. Pesan *decimal* di konversikan ke bentuk karakter.

2.2 Alat dan Bahan Penelitian

a. Alat

- Perangkat Keras (*Hardware*) : *Processor Intel Core i5 2.5 Ghz, Harddisk 500 GB, Memory 8 GB*
- Perangkat Lunak (*software*) : Sistem Operasi *Windows 10, Microsoft Office Word 2010, Visual basic 6, Adobe Photoshop CS3, Askapache*(https://www.askapache.com/online-tools/base64-image-converter/), *Cryptii*(https://v2.cryptii.com/base64/decimal)

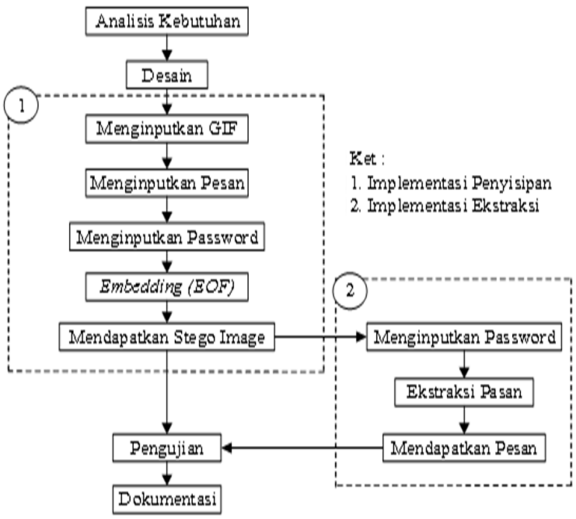
b. Bahan

Bahan-bahan yang digunakan adalah Citra Gif dengan ukuran 200 KB sampai dengan 2 MB

Tabel 1. Citra GIF Yang Digunakan

Nama	Size (bytes)	Dimensi (pixel)
1.GIF	230.308	*245 x 200*
2.GIF	1.219.635	*300 x 300*
3.GIF	1.863.430	*272 x 261*
4.GIF	553.587	*400 x 250*
5.GIF	1.686.634	*486 x 276*

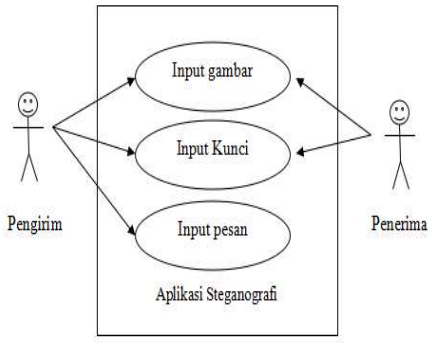
2.3 Tahapan Penelitian



Gambar 1. Tahapan Penelitian dan Pengembangan Sistem

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

Berikut desain Unified Modeling Language Aplikasi Steganografi EoF

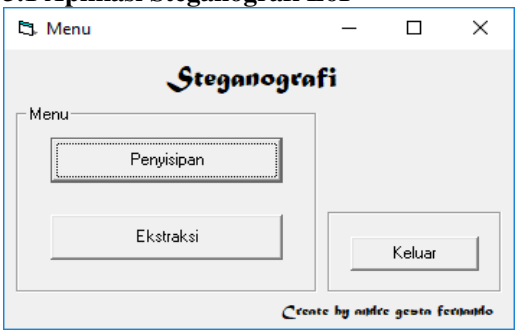


Gambar 2. UML Aplikasi Steganografi EoF

Pada gambar di atas dapat dijelaskan pengirim menginputkan sebuah gambar berformat GIF, kunci steganografi dan pesan kemudian diolah sistem dan dijadikan GIF dengan berisi informasi, pada sisi penerima menginputkan gambar berformat GIF dan kunci steganografi yang dikirim oleh pengirim kedalam sistem dan sistem mengeluarkan sebuah pesan.

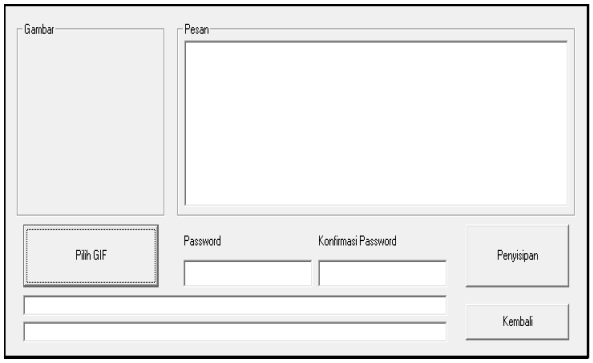
3. HASIL DAN PEMBAHASAN

3.1 Aplikasi Steganografi EoF



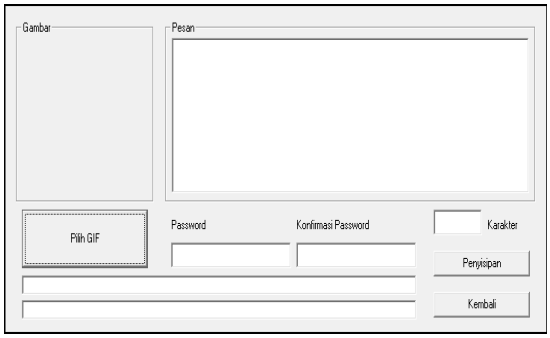
Gambar 3 Halaman Menu Aplikasi Steganografi

Pada halaman menu terdapat menu penyisipan yaitu halaman yang digunakan untuk menyisipkan pesan pada citra GIF. Menu ekstraksi merupakan halaman yang digunakan untuk menguraikan kembali pesan yang telah disisipkan. Proses steganografi dimulai dengan menekan tombol penyisipan, dan akan muncul halaman penyisipan.



Gambar 4 Halaman Penyisipan Pesan

Pada halaman penyisipan terdapat beberapa propertis yaitu, *button* "Pilih GIF", *button* "Penyisipan", *button* "Kembali", *RichTextBox* untuk pesan yang akan disisipkan, dan *text* untuk *password* yang akan digunakan. Selanjutnya untuk *button* "Penyisipan", disinilah inti bagaiman pesan disisipkan ke dalam citra GIF dengan algoritma *End OF File* (EOF). Langkah kerja pada halaman ini sebagai berikut pertama *file* GIF di ubah ke bentuk *binary* dan disisipkan ke media penyisipan, kemudian *password* dan pesan di ubah ke bentuk *binary*. *Password* pertama di masukkan ke media setelah itu baru pesan dan di masukkan lagi *password* kedua. Pada halaman pengekstrakan tidak jauh berbeda dengan halaman penyisipan, yang berbeda yaitu, pada halaman penyisipan menggunakan *button* penyisipan sedangkan pada halaman pengekstrakan menggunakan *button* pengekstrakan. Selain itu perbedaan terletak pada posisi penginputan *password* dan pesan, di halaman pengekstrakan tidak memiliki pengulangan *password*.



Gambar 5. Halaman Pengekstrakan Pesan

Pada halaman di atas *file stego* di ubah ke bentuk biner, setelah di ubah pesan

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

dipisahkan dari *file stego* dan *password*nya. Pesan yang sudah di dapat ditampilkan di *RichTextBox*. Dan didapatkan pesan yang tersembunyi di dalam *file* stego tersebut.

3.2 PENGUJIAN

Dalam melakukan pengujian diambil beberapa sampel citra GIF dengan berbagai ukuran dan dimensi. Dalam pengujian yang akan di uji di antaranya perubahan ukuran citra, pengujian dengan aplikasi, histogram, dan mengubah *stego* menjadi *grayscale*. Dalam aplikasi ini menggunakan satu *password* yaitu "informatika".

a. Pengujian Perubahan Ukuran Citra

Tabel 2. Pengujian Perubahan Ukuran Citra

No	Σchar	Ukuran Citra (*byte*)				
		1.gif (245 x 200)	**2.gif (300 x 300)**	**3.gif (272 x 261)**	**4.gif (400 x 250)**	**5.gif (486 x 276)**
1	50	230.384 (245 x 200)	1.219.711 (300 x 300)	1.863.506 (272 x 261)	553.663 (400 x 250)	1.686.710 (486 x 276)
2	100	230.434 (245 x 200)	1.219.761 (300 x 300)	1.863.556 (272 x 261)	553.713 (400 x 250)	1.686.760 (486 x 276)
3	500	230.834 (245 x 200)	1.220.161 (300 x 300)	1.863.956 (272 x 261)	554.113 (400 x 250)	1.687.160 (486 x 276)
4	1000	231.334 (245 x 200)	1.220.661 (300 x 300)	1.864.456 (272 x 261)	554.613 (400 x 250)	1.687.660 (486 x 276)
5	5000	235.334 (245 x 200)	1.224.661 (300 x 300)	1.868.456 (272 x 261)	558.613 (400 x 250)	1.691.660 (486 x 276)

Dari tabel 2 dapat dijelaskan bahwa perubahan ukuran dari citra GIF menjadi *stego image*, sangat berpengaruh pada jumlah karakter yang disisipkan. Sehingga semakin banyak karakter pesan yang disisipkan maka semakin besar ukuran *file stego*. Untuk perubahan dimensi pada *file stego*, *file* tidak mengalami perubahan dimensi walaupun pesan yag dimasukkan mencapai 5000 karakter.

b. Pengujian Dengan Aplikasi

Pengujian dengan aplikasi bertujuan untuk melihat apakah *file* GIF masih bisa dibuka dengan aplikasi pembuka *file* GIF setelah disisipkan pesan. Hal ini dapat dilihat pada tabel .

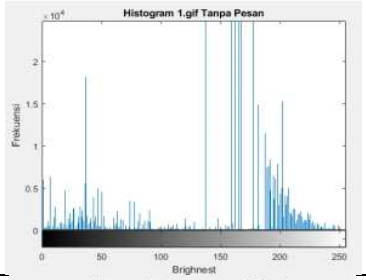
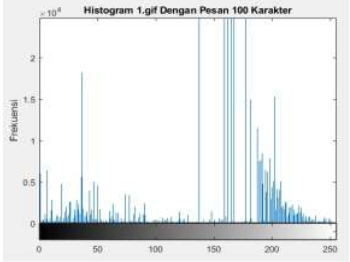
Tabel 3**.** Pengujian Dengan Aplikasi

No	Nama Aplikasi	Pengujian Stego *Image* Pada Pesan 50				
		1.gif	**2.gif**	**3.gif**	**4.gif**	**5.gif**
1	*Photos*	Terbuka	Terbuka	Terbuka	Terbuka	Terbuka
2	*Microsoft Office Picture Manager*	Terbuka	Terbuka	Terbuka	Terbuka	Terbuka
3	*Paint*	Terbuka	Terbuka	Terbuka	Terbuka	Terbuka
4	*Windows Media Player*	Terbuka	Terbuka	Terbuka	Terbuka	Terbuka
5	*Media Player Classic*	Terbuka	Terbuka	Terbuka	Terbuka	Terbuka

c.Pengujian *Fidelity* Citra *Stego*

Pada pengujian *fidelity* citra *stego* yang diuji yaitu perubahan yang terjadi pada citra GIF. Untuk melihat perubahan yang terjadi, dilihat pada *histogram* dari *file stego*. Pada pengujian *fidelity* digunakan citra asli 1.gif dengan citra yang sudah disisipkan pesan 10 karakter.

Tabel 4.Pengujian *Fidelity* Citra *Stego*

Karakter Pesan	Histogram
Tanpa Pesan	
100	

Berdasarkan tabel di atas dapat dilihat pada *histogram file stego* tidak adanya perubahan intensitas warna yang terjadi dengan histogram GIF aslinya, hal ini dilakukan dengan melihat perubahan intensitas warna

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

dilakukan dengan menghitung frekuensi dari nilai 0 sampai 255. Tidak terjadinya prubahan intensitas warna disebabkan karena algoritma EOF melakukan penyisipan di akhir *file* dan juga tidak merubah dimensi dari citra GIF, sedangkan nilai *histogram* dihitung berdasarkan dimensi citra, jadi dalam melakukan pencarian nilai *histogram* pesan tidak terhitung. Sehingga penerapan steganografi EoF pada citra GIF tidak merubahan citra GIF, sehingga pengujian *fidelity*-nya dapat dikatakan baik.

Pada pengujian robustness berupa manipulasi grayscale disimpulkan pesan tidak dapat di-*recovery* setelah dilakukan manipulasi. Hal ini disebabkan karena terjadinya perubahan terhadap *pixel* gambar, yang menyebabkan terjadinya perubahan tehadap pesan. Sehingga pesan tidak dapat dikenali. Dengan demikian steganografi dengan algoritma EoF tidak tahan terhadap *robustness.*

Dari hasil pengujian berdasarkan kriteria steganografi yaitu pengujian *fidelity,* citra *stego* tidak mengalami perubahan setelah disisipkan pesan. Ini menandakan bahwa pengujian *fidelity*-nya sangat baik. Kedua pengujian terhadap *robustness*, pada pengujian yang telah dilakukan citra *stego* tidak tahan terhadap manipulasi yang menyebabkan pesan tidak bisa dikembalikan lagi. Ketiga pengujian terhadap *recovery*, citra *stego* masih bisa mengembalikan pesan jika tidak dilakukan manipulasi terhadap citra *stego.* Sehingga dapat disimpulkan bahwa, steganografi dengan algoritma EoF pada media GIF memiliki kelebihan yaitu dapat menyisipkan pesan dengan baik tanpa merusak citra GIF dan dapat menyisipkan pesan hingga 100000 karakter. Algoritma EoF juga memiliki kekurangan yaitu sangat rentan terhadap manipulasi yang dilakukan terhadap citra penampung atau *file stego.*

4. KESIMPULAN DAN SARAN

Kesimpulan dari perancangan dan menganalisa steganografi dengan metode *End OF File* (EoF) pada citra GIF yaitu, perancangan steganografi dengan metode EoF dilakukan dengan *Microsoft Visual Basic* 6, yang dapat menyisipkan dan mengekstrak pesan dengan baik. Pengujian dilakukan dengan menyisipkan pesan mulai dari 50 karakter sampai 100000 karakter dengan ukuran citra GIF 200 kb hingga 2 MB. Pengujian dibatasi sampai 100000 karakter pesan karena dalam pengekstrakan pesan membutuhkan waktu yang lama. Pengujian berdasarkan kriteria steganografi. pengujian *fidelity,* citra *stego* tidak mengalami perubahan setelah disisipkan pesan. Ini menandakan bahwa pengujian *fidelity*-nya sangat baik. Kedua pengujian terhadap *robustness*, pada pengujian yang telah dilakukan citra *stego* tidak tahan terhadap manipulasi yang menyebabkan pesan tidak bisa dikembalikan lagi. Ketiga pengujian terhadap *recovery*, citra *stego* masih bisa mengembalikan pesan jika tidak dilakukan manipulasi terhadap citra *stego*. Sehingga dapat disimpulkan bahwa, steganografi dengan algoritma EoF pada media GIF memiliki kelebihan yaitu dapat menyisipkan pesan dengan baik tanpa merusak citra GIF dan dapat menyisipkan pesan hingga 100000 karakter.

DAFTAR PUSTAKA

[1] Munir. Rinaldi, 2004, "*Pengolahan Citra Digital Dengan Pendekatan Algoritmik*", Informatika, Bandung

[2] Morkel, T., Eloff, J. H. P., dan Olivier, M.S., 2005, "*An Overview of Image Steganography*", ICSA Reaserch Group, Department Computer Sceince Pretoria, South Africa.

[3] Darwis. Dedi, "*Teknik Steganografi Untuk Penyembunyian Pesan Teks Menggunakan Algoritma Ghifshuffle*", Jurnal TEKNOINFO, Vol.11, No.1, 2017, ISSN 1693-0010.

[4] Wamiliana, Hijriani. Astria, Darusman. Azharico, "*Implementasi Untuk Menyisipkan Pesan Teks pada Media Gambar dengan Metode End OF File*", Jurnal Komputasi, vol.3, No.2, 2015.

[5] Muslih, Rachmawanto. Eko Hari, "*Pengamanan File Multimedia dengan Metode Steganografi End Of File untuk Menjaga Kerahasian Pesan*", Techno.COM, Vol.1, No.1, Februari 2016.

[6] Sitorus. Michael, "*Aplikasi Keamanan Data Dengan Teknik Steganografi Menggunakan Metode End Of File (EOF)*", Univesitas Satya Negara Indonesia, volume 1, No 1. 2015, ISSN 2477-5894.

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31

[7] Sembiring. Sandro, "*Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End Of File*", 2013, Pelita Informatika Budi Darma, ISSN : 2301-9425.

[8] Martono, Irawan, "*Penggunaan Steganografi Dengan Metode End Of File (EOF) Pada Digital Watermarking*", Jurnal TICOM, Vol.2 No.1, September, 2013.

[9] Putra. Adhar Pratama, "*Rekayasa Perangkat Lunak Steganografi Dengan Metode Least Significant Bit Pada Web*", Institut Teknologi Padang, 2012.

[10] Penalosa, "*Steganografi Pada Citra Dengan Format GIF Menggunakan Algoritma Gifshuffle*", 2005, Teknik Informatika Institut Teknologi Bandung.

[11] Ariyus. Dony, 2008. "*Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*", Andi Offset, Yogyakarta.

DOI : https://doi.org/10.21063/JTIF.2020.V8.1. 25-31