RANCANG BANGUN APLIKASI PENGAMANAN DATA DENGAN STEGANOGRAFI PADA GAMBAR DENGAN METODE END OF FILE (EOF)

Painem

Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur
painem@budiluhur.ac.id

ABSTRAK

Perkembangan berbagai layanan berbasis internet seperti surat elektronik, toko online, berita online dan sebagainya mengakibatkan peningkatan lalu lintas data melalui jaringan internet. Maraknya pencurian data dan informasi yang terjadi di internet menuntut adanya suatu teknik yang dapat mengamankan proses bertukar data dan informasi melalui internet. Teknik steganografi merupakan salah satu teknik yang digunakan untuk menyembunyikan informasi pada suatu media seperti gambar, suara atau video. Teknik tersebut banyak digunakan untuk melindungi informasi dari pihak-pihak tertentu yang tidak berhak untuk mengetahui informasi. Pada penelitian ini dirancang suatu aplikasi steganografi yang bertujuan untuk menyembunyikan informasi berupa teks maupun file ke dalam media gambar. Penyembunyian informasi menggunakan teknik EOF(End Of File) dan algoritma enkripsi DES(Data Encryption Strandart). Teknik EOF (End Of File) merupakan teknik steganografi untuk menyisipkan pesan diakhir file. Melalui aplikasi yang dihasilkan, diharapkan data dan informasi yang akan dipertukarkan melalui media internet dapat terlindungi dengan aman.

Kata kunci: *Steganografi, EOF(End Of File),* penyembunyian informasi

1. PENDAHULUAN

Seiring kemajuan zaman, manusia memasuki era *internet*, dimana perkembangan dan pertukaran informasi pun berkembang dengan pesat. Kini informasi berupa pesan dokumen tersebut bersifat sangat rahasia atau pribadi, sehingga hanya boleh diakses oleh pihak tertentu. Seringkali seseorang yang hendak mengirimkan pesan ataupun dokumen kepada seseorang, tidak ingin jika isi pesan atau dokumen tersebut diketahui oleh orang lain.

Dengan alasan tersebut lahirlah kriptografi, yaitu metode pengolahan informasi algoritma tertentu sehingga menjadi sulit dimengerti maknanya. Namun metode ini sering menimbulkan kecurigaan pihak ketiga, sebabnya pesan yang sulit dimengerti pasti sudah diolah dan menunjukkan bahwa informasi tersebut bersifat penting dan rahasia. Oleh karena itu, muncul konsep steganografi yang berusaha menyembunyikan pesan di dalam media tertentu sehingga tidak mudah diketahui secara kasat mata. Media yang digunakan dalam steganografi dapat berupa teks, citra (gambar), audio maupun video.

Tujuan dari penelitian ini adalah untuk membuat rancang bangun aplikasi steganografi berbasis desktop yang dapat digunakan untuk menyembunyikan informasi atau pesan baik berupa teks biasa ataupun berupa file dokumen ke dalam citra (gambar). Dokumen yang dapat disisipkan berupa dokumen yang berekstensi *.doc, *.docx, *.xls dan *.pdf. Dengan aplikasi ini diharapkan data atau informasi dapat disembunyikan dan terjaga kerahasiaannya.

2. SEKILAS *STEGANOGRAFI*

Kata steganografi berasal dari bahasa Yunani *Steganos*, yang artinya "tersembunyi atau terselubung" dan graphien, "menulis" sehingga artinya adalah "menulis tulisan yang tersembunyi atau terselubung" (Sellars, 1996). Steganografi didefinisikan sebagai ilmu dan seni untuk menyembunyikan pesan rahasia(*hiding message*) sedemikian rupa sehingga keberadaan pesan tidak terdeteksi oleh manusia. Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi "rahasia" di dalam suatu informasi lainnya **[**1**]**.

Saat ini teknik steganografi banyak digunakan untuk menyembunyikan informasi rahasia dengan berbagai maksud. Salah satu tujuan steganografi adalah mengirim informasi rahasia melalui jaringan tanpa menimbulkan kecurigaan (Muhammad Hakim A., Chris Sanders, 2006).

A. Kriteria Penyembunyian Data Steganografi

Kriteria yang harus diperhatikan dalam penyembunyian data rahasia dengan menggunakan citra digital sebagai berkas penampung adalah :

1. *Imperceptibility*
   Keberadaan pesan rahasia tidak dapat dipersepsi oleh inderawi.Misalnya, jika *covertext* berupa citra, maka penyisispan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio (misalnya berkas audio *mp3, wav, midi* dan sebagainya), maka indra telinga tidak dapat mendeteksi perubahan audio *stegotext*-nya.

2. *Fidelity*
   Mutu media penampung tidak berubah banyak akibat penyisipan. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
3. *Recovery*
   Pesan yang disembunyikan harus dapat diungkapkan kembali (*reveal*).Karena tujuan steganografi adalah *data hiding,* maka sewaktu – waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk dapat digunakan lebih lanjut.
4. *Security*
   Pesan atau data rahasia yang disembunyikan ke suatu media haruslah terjamin keamanannya, sehingga pihak-pihak yang tidak berkepentingan tidak dapat mengetahui keberadaan informasi yang telah disisipkan tersebut.

B. Terminologi Steganografi

Steganografi memerlukan setidaknya dua properti. Properti pertama adalah wadah penampung (*cover*) dan yang kedua adalah data atau pesan yang disembunyikan. Berikut adalah beberapa istilah yang sering digunakan dalam steganografi:

1) *Hiddentext* atau *embedded message* : pesan yang disembunyikan.
2) *Cover-object* atau *stego-medium* : media yang digunakan sebagai penampung pesan.
3) *Carrier file* atau *stego-object* : *file* atau media yang sudah berisi pesan rahasia.
4) *Steganalysis* : proses untuk mendeteksi keberadaan pesan rahasia daalam suatu *file*.

C. Media Steganografi

Teknik steganografi yang digunakan dalam dunia modern saat ini sudah beraneka ragam. Mulai dari algoritma yang digunakan sampai pada media yang digunakan. Beberapa contoh media penyisipan pesan rahasia yang digunakan steganografi antara lain adalah [2] [3]:

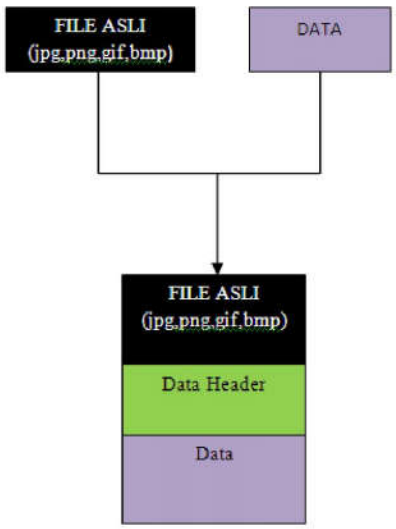
1) Steganografi pada *Text*
   Steganografi pada *text* terbagi menjadi dua penerapannya, yaitu pada *soft-copy text* dan *hard-copy text.* Pada *soft-copy text,* steganografi menyandikan data dengan mengubah jumlah spasi setelah tanda baca. Sedangkan pada *hard-copy text*, ada dua metode yaitu *Line Shift Coding* (menggeser setiap baris ke atas atau ke bawah) dan *Word Shift Coding* (menggeser beberapa kata ke kiri atau ke kanan).
2) Steganografi pada *Image*
   Sebagian besar penelitian dan perancangan aplikasi steganografi adalah pada citra *digital*. Hal ini disebabkan sebuah *image* dengan informasi rahasia di dalamnya lebih mudah disebarluaskan melalui *web* atau forum. Yang perlu diperhatikan adalah ketika informasi disembunyikan ke dalam *file image* kemudian *image* tersebut diubah ke format *image* lain, maka informasi yang disembunyikan akan hilang.
3) Steganografi pada *Audio*
   Steganografi juga dapat diterapkan pada suara *digital*. Namun, untuk steganografi pada *file audio* perlu kehati-hatian pada perancangan algoritma steganografi-nya, karena suara lebih sensitif daripada citra. Hal ini berarti suara *digital* lebih mudah rusak bila ditambahkan steganografi.
4) Steganografi pada *Video*
   Steganografi pada *video* sangat mirip dengan steganografi pada *image*, kecuali bahwa informasi disimpan pada setiap *frame video.* Steganografi pada *video digital* harus dirancang sedemikian rupa sehingga peralihan *image* dari satu *frame* ke *frame* lainnya harus tetap baik dan tidak terlihat dimodifikasi. Karena *video digital* ukurannya relative besar daripada citra *digital*, maka informasi yang disisipkan dapat lebih banyak.

D. Metode End Of File

Metode EOF (*End Of File*) [4] merupakan salah satu teknik yang menyisipkan data pada akhir file dan pengembangan daripada metode LSB (*Least Significant Byte*) [5] [6]. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sama dengan ukuran file sebelum disisipkan ditambah dengan ukuran data yang disisipkan ke dalam file tersebut. Dalam teknik EOF, data yang disisipkan pada akhir diberi tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut. Dalam teknik ini, data disisipkan pada akhir file dengan diberi tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut.

Teknik EOF tidak akan mengubah isi awal file yang disisipi. Sebagai contoh, jika akan menyisipkan sebuah pesan ke dalam sebuah file dokumen, isi dari dokumen tersebut tidak akan berubah. Ini yang menjadi salah satu keunggulan metode EOF di banding metode steganografi yang lain. Karena disisipkan pada akhir file, pesan yang disisipkan tidak akan bersinggungan dengan isi file, hal ini menjadikan integritas data dari file yang disisipi tetap terjaga. Namun metode EOF akan mengubah besar ukuran file sesuai dengan ukuran pesan yang disisipkan kedalam file master namun tidak mengubah citra dari media yang dipakai sebagai tempat penyisipan pesan tersebut.

Struktur file steganografi dengan metode *End Of File*, secara umum steganografi (file yang disisipi data) memiliki struktur di bawah ini:



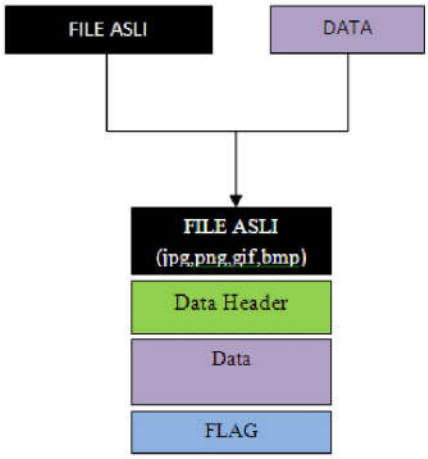
Gambar 1 : Struktur Awal File Steganografi dengan Metode EOF

Setiap blok pada sebuah *file* dapat kita baca dengan menggunakan dua parameter, posisi awal dan panjang blok tersebut. Dengan struktur diatas, kita dapat membaca posisi awal [*Data Header*] yang isinya meliputi :

- Posisi awal [DATA] pada file.
- Panjang [DATA] pada file.

Bagaimana menentukan posisi dan panjang data [*Data Header*] itu sendiri ? Kita dapat memakai *looping* yang mencari penanda (FLAG) sebagai penentu posisi awal [*Data Header*] pada *file* media mulai dari awal *file*. Namun cara ini akan menjadi tidak efisien dan menjadi lambat apabila *file* media [File Asli] berukuran sangat besar (misalnya 100MB).

Karenanya, penanda [Data Header] atau FLAG akan kita letakan di awal atau di akhir file dimana tidak ada looping yang digunakan untuk mencarinya.



Gambar 2: Struktur Akhir File Steganografi dengan Metode EOF

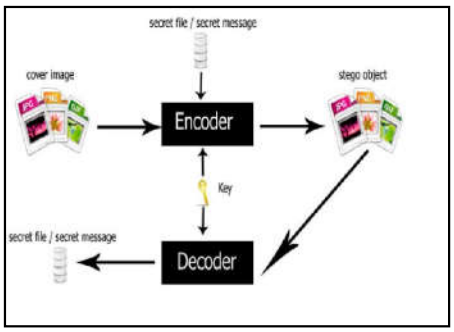
3. RANCANGAN APLIKASI

Aplikasi steganografi yang akan dibuat terdiri dari 6 buah *form* yang terdiri dari *Form* Menu Utama, *Form Embed Message*, *Form Retrieve Message*, *Form Embed File*, *Form RetrieveFile*, dan *Form Help*.

Untuk melakukan penyisipan pesan ke dalam gambar *user* dapat memilih menu *Embed Message*.Pada menu ini, *user* diharuskan memilih *file master image* terlebih dahulu, kemudian pilih juga *output image*nya yang nantinya *output* inilah yang akan berisi pesan setelah dilakukan proses *embed message*, baru kemudian memilih pesan berupa teks yang akan disisipi sebelum melakukan proses *Embed Message.* Setelah menentukan *file master image,file output image*dan pesan yang akan disispi, barulah proses *Embed Message* bisa berjalan.

Untuk melakukan penyisipan *file* dokumen ke dalam gambar *user* dapat memilih menu *Embed File*.Pada menu ini, *user* diharuskan memilih *file master image* terlebih dahulu, kemudian pilih juga *output image*nya yang nantinya *output* inilah yang akan berisi *file* dokumen setelah dilakukan proses *embed file*, baru kemudian memilih *file* dokumen yang akan disisipi sebelum melakukan proses *Embed File.* Setelah menentukan *file master image, file output image*dan pesan yang akan disispi, barulah proses *Embed File* bisa berjalan.

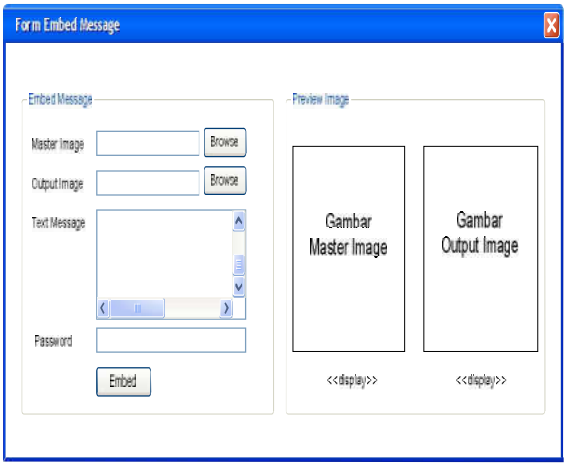
Sedangkan untuk membaca kembali pesan atau file dokmen dari sebuah gambar, *user* dapat memilih menu *Retrieve Message atau Retrieve File.* Pada aplikasi ini juga disediakan menu *Help* untuk membantu *user* dalam menggunakan aplikasi ini. Gambar 3 merupakan alur atau proses dari aplikasi steganografi



Gambar 3: Alur proses aplikasi steganografi

3.1 Rancangan *Embed Message*

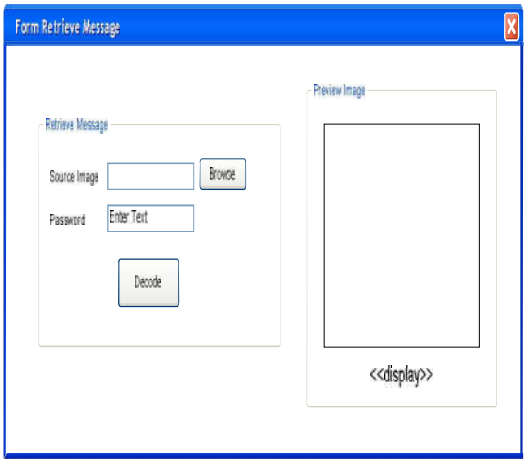
Rancangan layar *form embed message* digunakan untuk menyisipkan pesan kedalam gambar. Untuk menyisipkan pesan ke dalam gambar *user* terlebih dahulu memilih gambar yang akan dijadikan master *image* kemudian pilih *image* atau buat file *image* baru yang nantinya akan disisipkan pesan, kemudian tulis pesan pada *text area*, dan isikan *password* . Setelah data dirasa valid tekan tombol *Embed*.



Gambar 4: Rancangan layar *form embed message*

3.2 Rancangan Layar *Form Retrieve Message*

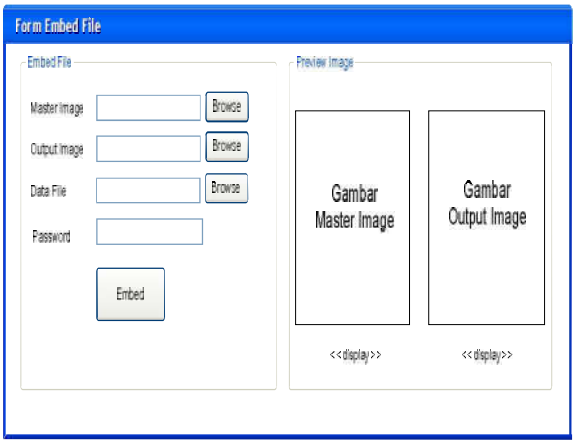
Rancangan Layar pada gambar 3.4 yaitu *form retrieve message* digunakan untuk mendapatkan kembali pesan yang telah disisipkan sebelumnya. Untuk mendapatkan kembali pesan yang telah disisipkan yaitu dengan cara pilih file image yang telah disisipkan pesan dan isikan *password* yang sesuai setelah itu klik tombol *Decode*.



Gambar 5 : Rancangan layar *form retrieve message*

3.3 Rancangan Layar Form *Embed File*

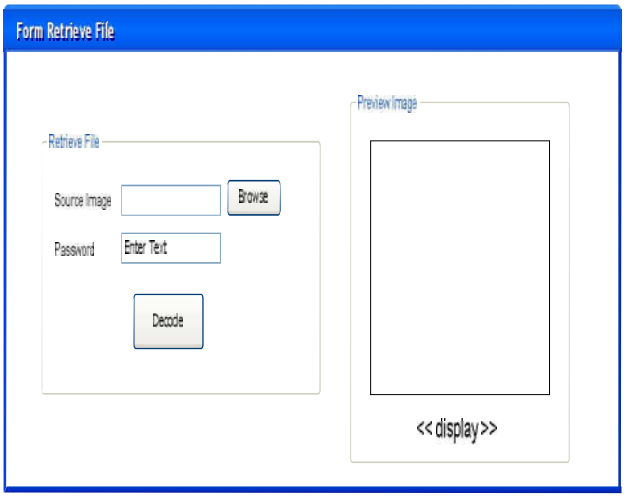
Rancangan layar *form embedfile* digunakan untuk menyisipkan *file* dokumen seperti .doc, .docx, .xls, dan .pdf ke dalam gambar. Untuk menyisipkan pesan ke dalam gambar *user* terlebih dahulu memilih gambar yang akan dijadikan master image kemudian pilih image atau buat *file image* baru yang nantinya akan disisipkan *file* dokumen, kemudian pilih *file* dokumen, dan isikan *password* . Setelah data dirasa valid tekan tombol Embed.



Gambar 6 : Rancangan layar *form embed file*

3.4 Rancangan Layar Form Retrieve File

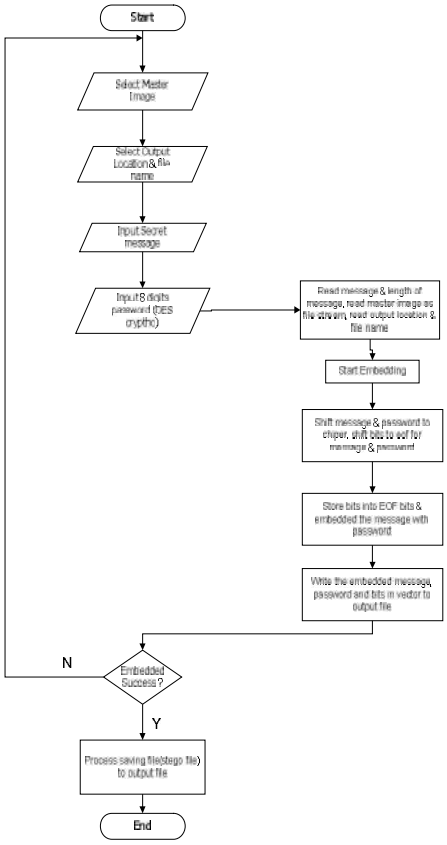
Rancangan Layar pada gambar 3.7 yaitu *form retrieve file* digunakan untuk mendapatkan kembali *file* dokumen yang telah disisipkan sebelumnya. Untuk mendapatkan kembali *file* dokumen yang telah disisipkan yaitu dengan cara pilih *file image* yang telah disisipkan file dokumen dan isikan *password* yang sesuai setelah itu klik tombol *Decode*.



Gambar 6 : Rancangan layar *form retrieve file*

3.5 Flowchart dan algoritma

1) Flowchart proses embed

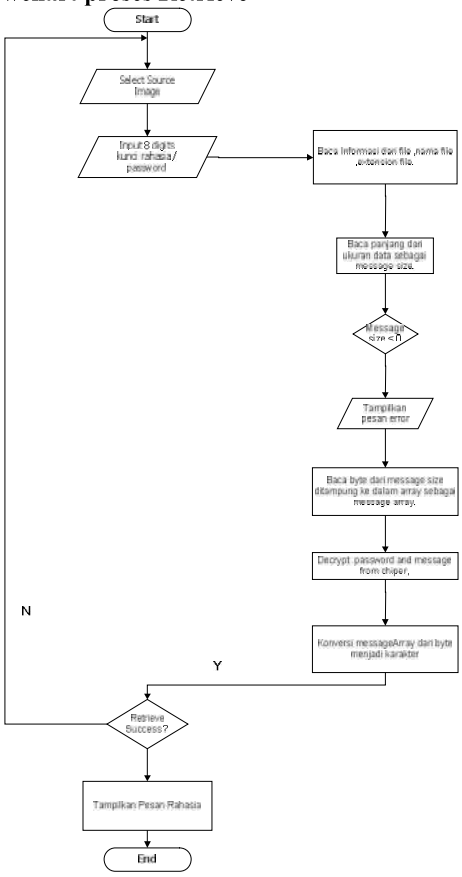


Gambar 7: Flowchart Proses *Embed*

2) Algoritma Proses Embed

1. *Start*
2. *Select Master Image*
3. *Select Output Location & file name*
4. *Input Secret Message*
5. *Input 8 digits password (DES cryptho)*
6. *Read message & length of message, read master image as file stream, read output location & file name.*
7. *Start Embedding*
8. *Shift message & password to chiper, shift bits to eof for message & password*
9. *Store bits into EOF bits & embedded the message with password*
10. *Write the embedded message,password and bits in vector to output files*
11. *If "Embedded Success ?"*
12. *Process saving file stego file to output file.*
13. *Else*
14. *Kembali ke baris 1*
15. *End If*
16. *End*

3) Flowchart proses Retrieve



Gambar 8: Flowchart Proses *Retrieve*

4) Algoritma proses Retrieve

1. *Start*
2. *Select Source Image*
3. *Input 8 digits kunci rahasia / password*
4. *Baca Informasi dari file, nama file,extension file.*
5. *Baca panjang dan ukuran data sebagai message size*
6. *If "Message Size < 0"*
7. *Tampilkan pesan error*
8. *End If*
9. *Baca byte dan message size ditampung kedalam array sebagai message array*
10. *Decrypt password dan pesan dari chipper texr ke plain text*
11. *Konversi messageArray dari byte menjadi karakter array*
12. *If "RetrieveSuccess ?"*
13. *Tampilkan pesan rahasia.*
14. *Else*
15. *Kembali ke baris 1*
16. *End If*
17. *End*

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

a. Rancang bangun aplikasi ini memberikan sesuatu hal yang baru dan menarik untuk diterapkan bagi pihak-pihak yang berkepentingan
b. Dengan metode *End Of File* ini, gambar yang disisipi pesan secara kasat mata tidak terlihat perbedaan dengan gambar aslinya, sehingga gambar tersebut tidak mudah dicurigai oleh pihak yang tidak berkepentingan.
c. Rancang bangun aplikasi ini dilengkapi enkripsi pesan sebelum pesan tersebut disisipi kedalam gambar, agar lebih terjaga kerahasiaan dari suatu data

4.2. Saran

Metode *End Of File* ini dapat dicurigai apabila file dokumen atau pesan yang disisipkan sangat besar karena mengubah ukuran file gambar yang dijadikan *stego-image* sehingga diharapkan kedepannya dapat ditambahkan fungsi kompres pada file yang disisipkan agar tidak terlalu terlihat perbedaannya.

DAFTAR PUSTAKA

[1] Cummins, Jonathan., Patrick,Diskin., Samuel,lau., Robert,Parlet. (2004), *Steganography and Digital Watermaking*.
[2] Cahyana, T. Basarudin dan Danang Jaya, 2007, Teknik Watermarking Citra berbasis SVD. National Conference on Computer Science & Information Technology 2007.
[3] Solichin, Achmad. 2010. *Digital Watermarking untuk Melindungi Informasi Multimedia,* Jurnal BIT FTI Vol 7 No 1 2010 hal 1-8.
[4] Utami, ema dan Sukrisno 2007 , *Implementasi Steganografi EoF dengan Gabungan Ekripsi Rijndael, Shift Chiper dan Fungsi Hash*, Yogyakarta
[5] Iswahyudi, C. (2008), *Penyisipan Pesan Rahasia pada Citra Digital dengan Teknik Steganografi.*
[6] Rakhmat, Basuki dan Fairuzabadi, Muhammad, (2010*), Steganografi Menggunakan Metode Least Significant Bit dengan Kombinasi Algoritma Kriptografi Vigenere dan RC4*, Jurnal Dinamika Informatika, Vol 5 No 2