PERBANDINGAN HASIL IMPELEMENTASI STEGANOGRAFI DAN KRIPTOGRAFI MENGGUNAKAN LSB (*LEAST SIGNIFICANT BIT*)DENGAN EOF (*END OF FILE*)

Bayu Widia Santoso[1], Fikri Reza AlHadi[2]
Program Studi Pascasarjana, Magister Ilmu Komputer, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail :bayuwidiasantoso@gmail.com[1], fikrirezaa@gmail.com[2]

ABSTRAK

*Seiring dengan semakin maju dan terus berkembangnya ilmu tentang teknologi informasi, Internet menjadi salah satu media komunikasi data yang banyak digunakan. Internet mampu menghubungkan hampir semua komputer yang ada diseluruh dunia sehingga dapat digunakan untuk media komunikasi. Bentuk informasi yang dapat ditukar bisa berupa data teks, citra digital, audio, video. Steganografi sebagai salah satu seni penyembunyian pesan ke dalam pesan lain yang telah ada sejak dahulu dan kini seiring dengan kemajuan teknologi informasi serta perkembangan dari teknologi digital, steganografi banyak dimanfaatkan untuk mengirim pesan melalui internet tanpa diketahui oleh orang lain dengan media citra digital. Untuk menjaga kerahasiaan dan keamanan informasi tersebut dilakukan dengan teknik kriptografi. Dalam tulisan ini akan dibahas tentang perbandingan beberapa teknik algoritma steganografi dan kriptografi, diantaranya: LSB **(Least Significant Bit)**, EOF **(End Of File)**, DES **(Data Encryption Standard)** dan Subtitusi **(Shift Chiper)**. Dimana tujuan penulisan perbandingan ini adalah untuk mengetahui kelemahan dan keunggulan dari teknik steganografi dan kriptografi tersebut.*

*Along with the more advanced and continued development of the science of information technology, the Internet becomes a medium of communication data is widely used. Internet is able to connect almost any computer that exist throughout the world so that it can be used for communication media. The information that can be exchanged such as text data, digital images, audio and video. Steganography is the art of hiding messages in other messages that have existed long ago and are now in line with advances in information technology and the development of digital technology, steganography being used to send messages via the Internet without being noticed by others with digital image media. To maintain the confidentiality and security of information is done by a cryptographic technique. In this paper will be discussed on a comparison of some steganography algorithms and cryptographic techniques, including: LSB **(Least Significant Bit)**, EOF **(End Of File)**, DES **(Data Encryption Standard)** and Shift Chiper. Where the purpose of the writing of this comparison is to*

identify the weaknesses and advantages of the technique of steganography and cryptography.

Kata Kunci : *steganografi, kriptografi, EOF, LSB, DES, Shift Chiper.*

1. PENDAHULUAN

1.1.Latar Belakang

Keamanan data dan informasi merupakan hal yang sangat penting di jaman yang semakin pesat ini. Hal itu sedang menjadi masalah bagi dunia telekomunikasi terutama dalam pengiriman informasi penting yang memerlukan kerahasiaan yang tinggi seperti keuangan bank, informasi rahasia negara. Oleh karena itu, diperlukannya metode untuk pengamanannya, salah satunya dengan menggunakan metode kriptografi Saat ini, kriptografi menjadi dasar bagi keamanan informasi atau dokumen-dokumen yang diamankan sedemikian rupa oleh pengirim sehingga orang lain tidak dapat mengetahui bahkan mengenali data tersebut. Biasanya perkembangan teknologi akan selalu diikuti dengan perkembangan kejahatan yang menggunakan teteknologi sebagai media. Salah satunya dengan cara untuk mengindari kejahatan tesebut dengan cara menyembunyikan atau menyisipkan pesan tersebut kedalam media lain. Seni menyembunyikan pesan kedalam media lain, sehingga keberadaan pesan tersebut tidak dapat diketahui hal ini dikenal dengan nama*steganografi*.

1.2. Masalah

Permasalahan yang akan terjadi pada penulisan yang penulis kerjakan diantaranya:

a. Rentannya keamanan pada informasi atau dokumen-dokumen.
b. Adanya pihak yang tidak berhak untuk mengetahui privasi atau kerahasian informasi atau dokumen-dokumen.
c. Sistem keamanan data yang mudah dipecahkan oleh pihak lain.

1.3. Tujuan Penulisan

Dengan adanya permasalahan tersebut, maka dikembangkan perangkat lunak yang dapat membantu menyelesaikan permasalahan tersebut. Adapun tujuannya adalah:

a. Memanfaatkan teknik Steganografi untuk menyembunyikan pesan kedalam citra digital
b. Memanfaatkan tehnik kriptografi untuk menyandikan pesan sebelum disisipkan.
c. Memunculkan kepedulian bagi para perancang sistem informasi terhadap keamanan infromasi atau dokumen-dokumen bahwa keamanan infromasi atau dokumen-dokumen merupakan bagian utama sistem yang patut diperhitungkan

1.4. Batasan Masalah

Membandingan Algoritma EOF (*End Of File*) dan DES *(Data Encryption Standard*) dengan Algoritma LSB (*Least Significant Bit*) dan Subsitusi (*Shift Chiper*) serta Mengimplemantasikan dan menguji sistem keamanan infromasi atau dokumen-dokumen guna mengetahui keunggulan system yang dibuat.

2. LANDASAN TEORI

2.1. *Steganografi*

a. Sejarah dan Pengertian *Steganografi*

Steganografi berasal dari bahsa yunani "*steganos*" yang artinya "tersembunyi" atau "terselubung" dan "*graphein*" yang

artinya "menulis". *Steganografi* dapat diartikan "tulisan tersembunyi" (*cover writing*). *Steganografi* adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui [1].*Steganografi* membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. *Steganografi* digital menggunakan media digital sebagai wadah penampung, misalnya: voice, video, image dan teks. Data rahasia yang disembunyikan juga dapat berupa voice, video, image dan teks [2].

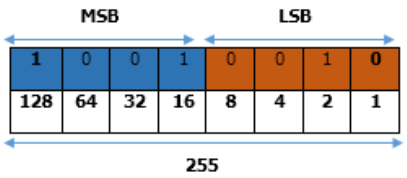
b. Metode *Steganografi*

1) Metode LSB (*Least Significant Bit*)

a) Pengertian Metode LSB (*Least Significant Bit*)

LSB (*Least Significant Bit*) merupakan metode stegnografi yang paling sederhana dan mudah untuk diimplementasikan ke sebuah aplikasi. Metode ini menggunakan citra digital sebagai *convertext*. Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti (*most significant bit* atau MSB) dan bit yang plaing kurang berarti (*least significant bit* atau LSB)[1].

b) Cara Kerja Metode LSB (*Least Significant Bit*)



Gambar 1: Struktur LSB

Berikut cara kerja dari metode LSB (*Least Significant Bit*): *byte* **1**0010010 angka **1** (pertama dari kiri) adalah bit dari MSB (*Most Significant Bit*) dn angka **0** (pertama dari kanan) adalah LSB (*Least Significant Bit*). Bit yang cocok untuk digantikan adalah bit LSB, Karena perubahan dari LSB tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya, misalnya *byte* tersebut menyatakan warna orange, maka perubahan satu bit LSB (*Least Significant Bit*) tidak mengubah warna orange tersebut secara berarti. Mata manusia tidak dapat membedakan perubahan kecil.

2) Metode EOF (*End Of File*)

a) Pengertian Metode EOF (*End Of File*)

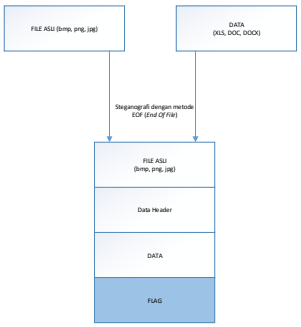
Metode EOF (*End Of File*) merupakan salah satu teknik yang menyembunyikan pesan pada akhir file. Teknik ini dapat digunakan untuk menyembunyikan pesan yang ukurannya sama dengan ukuran file sebelum disisipkan ditambah dengan ukuran pesan yang disisipkan ke dalam sebuah file. Dalam teknik EOF (*End Of File*) pesan yang disisipkan pada akhir diberi tanda khusus sebagai pengenal *start* dari pesan tersebut dan pengenal akhir dari pesan tersebut.

b) Cara Kerja Metode EOF (*End Of File*)

Teknik EOF (*End Of File*) tidak akan mengubah isi awal file yang disisipi. Berikut contoh menyisipkan sebuah pesan ke dalam sebuah file, isi dari

file tersebut tidak akan berubah. Ini salah satu kelebihan dati metode EOF (*End Of File*) dibandingkan dengan metode *steganografi* yang lain. Karena disisipkan pada akhir file, maka pesan yang disisipkan tidak bersinggungan dengan isi file, hal ini menjadi integritas data dari file yang disisipi tetap terjaga (Saputro, 2013). Namun metode EOF (*End Of File*) akan mengubah size file sesuai dengan ukuran pesan yang disembunyikan, namun tidak akan merubah image dari media yang dipakai sebagai tempat penyisipan pesan tersebut, berikut struktur medote EOF (*End Of File*):

Gambar 2: Struktur File Awal



Gambar 3: Struktur File Akhir

2.2. *Kriptografi*

a. Sejarah dan Pengertian *Kriptografi*

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut Bruce Scheiner dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga pesan tetap aman [5]. Suatu data yang tidak disandikan disebut *plaintext* atau *cleartext*. Sedangkan data yang telah tersandikan disebut *ciphertext*. Proses yang dilakukan untuk mengubah *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *encipherment*. Sedangkan proses untuk mengubah *ciphertext* kembali ke *plaintext* disebut dekripsi (*decryption*) atau *decipherment*. Dalam kriptografi diperlukan parameter yang digunakan untuk proses konversi data yaitu suatu set kunci. Enkripsi dan dekripsi data dikontrol oleh sebuah kunci atau beberapa kunci [8]. Berikut gambar *enkripsi* dan *dekripsi*:

Gambar 4: Proses Enkripsi/Dekripsi

b. Metode *Kriptografi*

1) Metode DES (*Data Encryption Standard*)

a) Pengertian Metode DES (*Data Encryption Standard)*

DES termasuk ke dalam system kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit. DESmengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan

menggunakan 56 bit kunci internal (*internal key*) atau up-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit[6]. Skema global dari algoritma DES adalah sebagai berikut:

(1) Blok *plaintext* dipermutasi dengan matriks permutasi awal

(2) Hasil permutasi awal kemudian di*enciphering*-sebanyak 16 kali (16putaran). Setiap putaran menggunakan kunci internal yang berbeda.

(3) Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan menjadi blok*cipherteks*.

Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masingmasing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok*R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal *Ki*. Keluaran dai fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok*L* yang baru langsung diambil dari blok *R* sebelumnya.

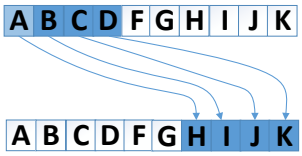
2) Metode Subsitusi (*Shift Chiper*)

a) Pengertian Metode Subsitusi (*Shift Chiper*)

Teknik subsitusi adalah sebuah teknik enkripsi yang menggunakan metode pertukaran huruf dengan huruf lainnya atau angka atau simbol tertentu. Ada beberapa algoritma dalam teknik ini, salah satunya metode *shift chiper* metode kriprografi ini mula-mula digunakan kaisar Romawi yaitu Julius Caesar untuk penyandian pesan yang dikirim kepada para bawahannya, sehingga metode ini disebut metode Caesar chipper.

b) Cara Kerja Metode Subsitusi (*Shift Chiper*)

Shift chiper merupakan teknik enkripsi yang paling sederhana dan banyak digunakan. Dimana setiap huruf pada *plaintext* digantikan dengan huruf yang lain. Misalnya diketahui pergeseran setiap huruf = **7**, maka huruf **A**akan digantikan oelh huruf **H**, huruf H akan menjadi **O** dan seterusnya, berikut gambar penjelesan pergeseran huruf:



Gambar 5: Subsitusi shift chipper

3. ANALISA DAN PERANCANGAN PROGRAM

3.1. Analisa Masalah

Seiring dengan perkembangan jaman, kebutuhan akan pengamanan terhadap document atau pesan semakin meningkat. Untuk menjaga kerahasiaan pada document atau pesan tersebut, maka dibutuhkan aplikasi yang dapat menjaga privacy dari pesan atau document tersebut. Yang menjadi masalah dalam

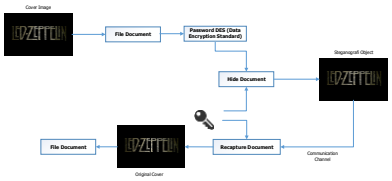
pengaman document atau pesan tersebut adalah pembajak (Hacker) untuk memperoleh data-data tersebut, selain itu juga keterbatasan sumber daya manusia yang belum mahir dalam pengaman document atau pesan.

3.2. penyelesaian Masalah

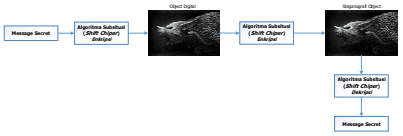
Dari permasalahan yang telah diuraikan tersebut, maka diperlukan adanya sebuah aplikasi yang dapat menjaga kerahasiaan informasi atau dokumen-dokumen yang sangat penting. Sehingga keberadaannya tidak terditeksi oleh pihak yang tidak berhak atas informasi tersebut. Aplikasi tersebut nantinya akan dapat menyembunyikan atau menyisipkan sebuah informasi atau dokumen-dokumen penting ke dalam citra digital yang berupa sebuah *image*. Pengguna atau sipengirim pesan dapat mengirimkan sebuah *image* yang telah disisipkan informasi atau dokumen-dokumen rahasia tersebut melalui jalur komunikasi publik, sehingga dapat diterima oleh pengguna lainnya atau pengguna yang dituju. Kemudian penerima informasi tersebut dapat mengekstrasi informasi rahasia yang ada didalamnya.

3.3. Perancangan Aplikasi

Aplikasi steganografi yang diusulkan yakni terdiri dari 2 tipe yang terdiri dari *Form Stegnografi* dengan EOF (*End Of File*) yang terdiri dari *Form Hide Document* untuk menyembunyikan file pada menu ini *user* diharuskan memilih *file image* terlebih dahulu baru kemudian menyisipkan file yang akan disembunyikan, *Form Recapture Document* untuk menampilkan kembali file informasi atau dokumen-dokumen dari sebuah gambar yang telah disisipkan informasi tersebut. Dan *form Steganografi* dengan LSB (*Least Significant Bit*) pada *form* ini pengguna memilih gambar yang akan disembunyikan pesan dan memasukkan pesan yang akan disandikan atau di *enkrip* kemudian disembunyikan kedalam gambar, kemudian gambar yang sudah disembunyikan pesan akan disimpan dan ditampilkan pada form ini.



Gambar 6 Steganografi EOF



Gambar 7 Steganografi LSB

3.4. Rancangan Layar

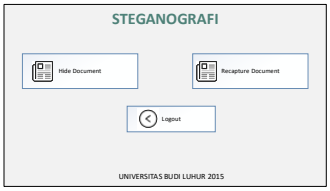
a. *Form Menu Utama*

Pada gambar 8 merupakan rancangan layar *form* menu home terdapat menu *Steganografi* EOF(*End Of File*), menu *Steganografi* LSB *(Least Significant Bit)* dan menu *Logout*.

Gambar 8 Rancangan Layar Form Menu Utama

b. *Form*Home EOF (*End Of File*)

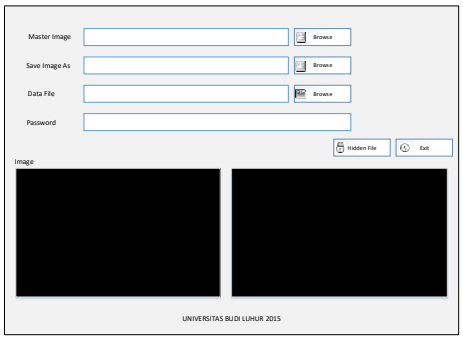
Pada gambar 9 merupakan rancangan layar *form* menu home EOF (*End Of File*), terdapat menu *Hide Document*, menu *Recapture Document* dan menu *Exit*.



Gambar 9 Rancangan Layar Form Home EOF (End Of File)

c. *Form Hide Document*

Pada gambar 10 merupakan rancangan layar *form* menu *Hide Document*, terdapat *image original*, *duplicate image*, *document*, *password*, tombol *browse* untuk melakukan mengunggah gambar dan mengunggah dokumen, tombol *hide document* untuk melakukan proses penyisipan document pada sebuah gambar dan menu *Exit* untuk melakukan keluar dari *form Hide Document*.



Gambar 10 Rancangan Layar Form Hide Document

d. *Form Recapture Document*

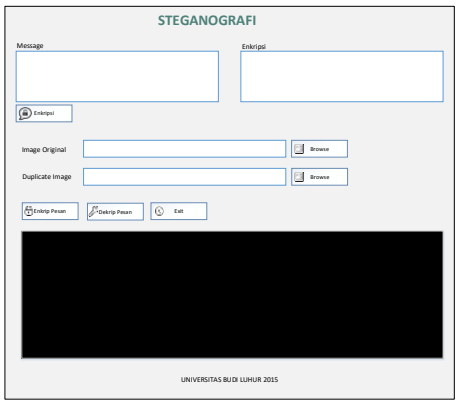
Pada gambar 11 merupakan rancangan layar *form* menu *Recapture Document*, terdapat *duplicate image*, *password*, tombol *browse* untuk melakukan mengunggah gambar, tombol *recapture document* untuk melakukan proses pengambilan document pada sebuah gambar dan menu *Exit* untuk melakukan keluar dari *form Recapture Document*.

Gambar 11 Rancangan Layar FormRercapture Document

e. *Form Home LSB*

Pada gambar 12 merupakan rancangan layar *form* menu *Steganografi* LSB (*Least Significant Bit*), terdapat *message*, *enkripsi*,

image original, *duplicate image*, tombol *enkripsi* untuk melakukan proses pengamanan pada *message*, tombol *browse* melakukan mengunggah gambar, tombol *enkrip message* untuk melakukan proses penyisiapan pesan pada sebuah gambar, tombol dekrip message untuk melakukan proses pengambilan pesan pada sebuah gambar dan menu *Exit* untuk melakukan keluar dari *form Home*.

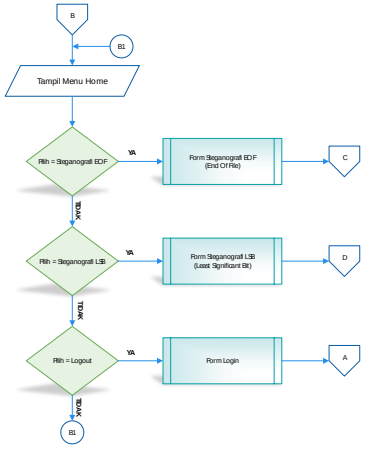


Gambar 12 Rancangan Layar Form Home LSB (Least Significant Bit)

3.5. *lowchart*

a. *Flowchart Form Menu Utama*

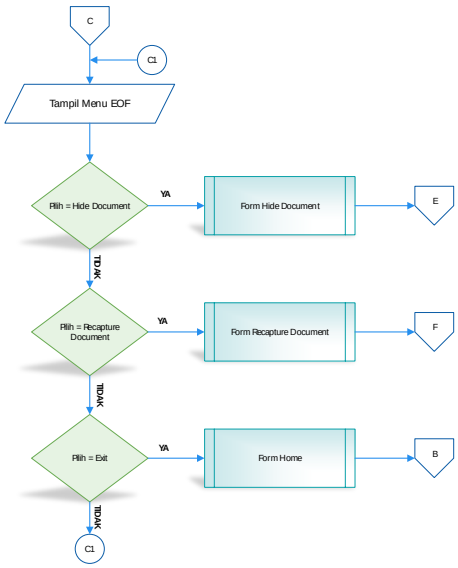
Flowchart ini menggambarkan proses untuk menampilkan *form menu* utama. Berikut merupakan *flowchart* untuk halaman *formmenu* utama, dapat dilihat pada gambar 13:



Gambar 13 : Flowchart Form Menu Utama

b. *Flowchart FormHome* EOF (*End Of File*)

Flowchart ini menggambarkan proses untuk menampilkan *form Home* EOF (*End OF File*). Berikut merupakan *flowchart* untuk halaman *formHome* EOF (*End OF File*), dapat dilihat pada gambar 14:

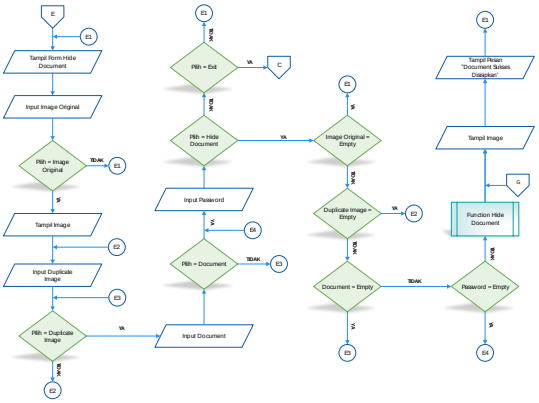


Gambar 14 : Flowchart Form Home EOF

c. *Flowchart Form Hide Document*

Flowchart ini menggambarkan proses untuk menampilkan *form Hide Document*. Berikut merupakan *flowchart* untuk halaman *formHide*

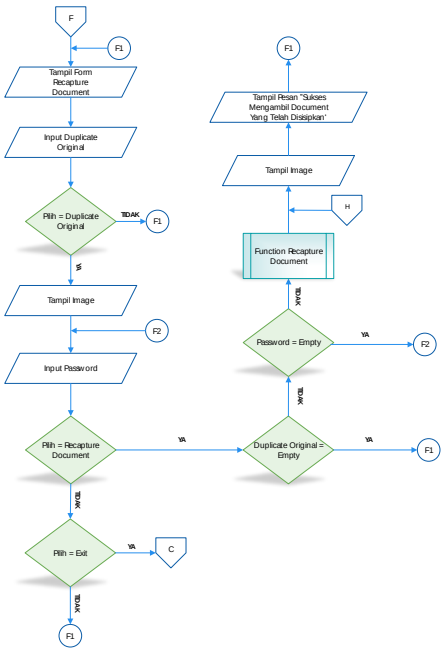
Document, dapat dilihat pada gambar 15:



Gambar 15 : Flowchart Form Hide Document

d. *Flowchart FormRecapture Document*

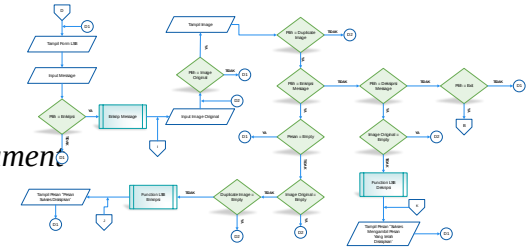
Flowchart ini menggambarkan proses untuk menampilkan *form Recapture Document*. Berikut merupakan *Flowchart* untuk halaman *formRecapture Document*, dapat dilihat pada gambar 16:



Gambar 16 : Flowchart Form Recaoture Document

e. *Home* LSB (*Least Significant Bit*)

Flowchart ini menggambarkan proses untuk menampilkan *form Home* LSB(*Least Significant Bit*). Berikut merupakan *flowchart* untuk halaman *formHome* LSB(*Least Significant Bit*), dapat dilihat pada gambar 17:



Gambar 17 : Flowchart Form Home LSB

4. IMPLEMENTASI DAN ANALISA PROGRAM

4.1. Pendahuluan Implementasi

Pada kasus ini akan dilakukan pengujian serta analisa dari program *steganografi* yang telah dibuat. Tujuannya adalah untuk mengukur sejauh mana *steganografi* ini dapat menyelesaikan suatu masalah. Dengan adanya pengimplementasian serta uji coba tersebut dapat mempermudah untuk melihat adanya kekurangan pada program yang telah dibuat.Dengan demikian pada masa yang akan datang dapat dilakukan pengembangan aplikasi kearah yang lebih baik lagi. Analisa program dilakukan untuk mengukur sejauh mana program ini dapat berjalan dengan baik dan membantu pengguna dalam pencarian. Pada program *steganografi* ini dibuat beberapa tampilan untuk mempermudah pemakai

dalam menggunakan program *steganografi* ini.

4.2. Implementasi Program

Pada bagian ini, diuraikan mengenai tampilan layar aplikasi *steganografi* mulai dari *login* kali dijalankan sampai program selesai dijalankan. Berikut ini akan diberikan penjelasan dan gambar mengenai tampilan-tampilan yang ada pada program aplikasi *steganografi* ini.

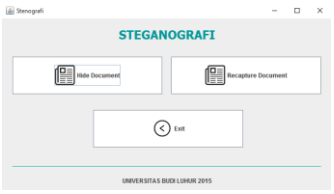
a. Tampilan Layar *Form Home*

Pada gambar 18 adalah tampilan layar *formhome* yang akan muncul apabila *login* diisi dengan benar. Menu utama ini berfungsi untuk menampilkan menu yang ada diaplikasi *Steganografi home*.

Gambar 18 : Tampilan Layar Form Home

b. Tampilan Layar *Form Home* EOF (*End Of File*)

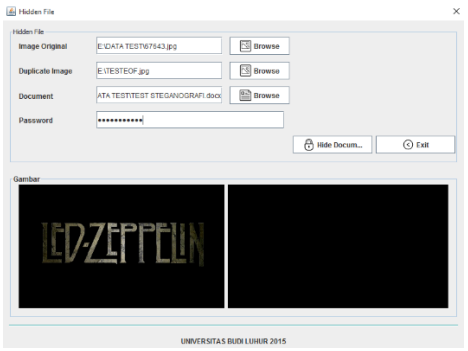
Pada gambar 19 adalah tampilan layar *formhome* EOF (*End Of File*) yang akan muncul apabila *menekan tombol Steganografi* EOF (*End Of File*). Menu utama ini berfungsi untuk menampilkan menu yang ada diaplikasi *Home* EOF (*End Of File*).



Gambar 19 Tampilan Layar Form Home EOF (End Of File)

c. Tampilan Layar *Form Hide Document*

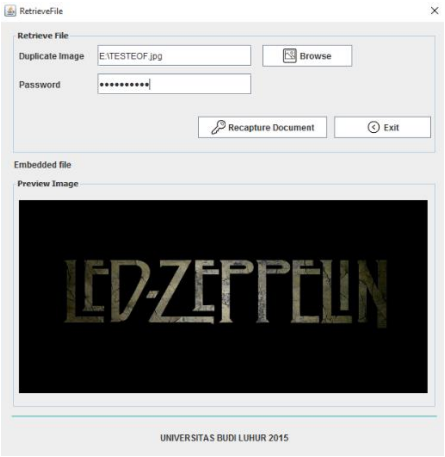
Pada gambar 20 adalah tampilan layar *formHide Document*. Pada *formHide Document* ini berfungsi untuk menambahkan gambar yang ingiin disisipkan dengan sebuah file atau document seperti DOC, DOCX dan XLS dengan mengisi *field-field* yang tersedia tombol *browse* untuk mengambil gambar atau mengambil document yang ingin disisipkan, kemudian pilih tombol "*Hide Document*" untuk menyimpan data, pilih tombol "exit" untuk kembali ke *home* EOF (*End Of File*).



*Gambar 20 Tampilan Layar Form **Hide** Document*

d. Tampilan Layar *Form Recapture Document*

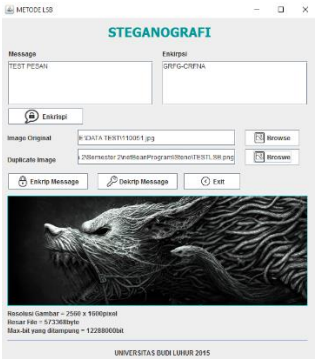
Pada gambar 21 adalah tampilan layar *formRecapture Document*. Pada *formRecapture Document* ini berfungsi untuk mengambil document atau file yang telah disisipkan pada gambar dengan mengisi *field-field* yang tersedia tombol *browse* untuk mengambil gambar atau mengambil document yang ingin disisipkan, kemudian pilih tombol "*RecaptureDocument*" untuk mengambil data, pilih tombol "exit" untuk kembali ke *home* EOF (*End Of File*).



Gambar 21 Tampilan Layar Form Recapture Document

e. Tampilan Layar *Form* LSB(*Least Significant Bit*)

Pada gambar 22 adalah tampilan layar *form* LSB (*Least Significant Bit*). Pada *form* LSB (*Least Significant Bit*) ini berfungsi untuk mengambil atau menambahkan pesan yang telah disisipkan pada gambar dengan mengisi *field-field* yang tersedia tombol *browse* untuk mengambil gambar atau mengambil document yang ingin disisipkan, kemudian pilih tombol "*Enkrip Message"* untuk menyisipkan pesan kedalam sebuah image,pilih tombol "*Dekrip Message"* untuk megambil pesan didalam sebuah image, pilih tombol "exit" untuk kembali ke *home* .



Gambar 22 Tampilan Layar Form LSB(Least Significant Bit)

4.3. Uji Kasus

Agar aplikasi steganografi dapat berjalan dengan baik maka perangkat yang dipakai untuk implemetasi aplikasi ini juga harus mendukung. Spesifikasi berikut bisa mendukung untuk aplikasi ini, berikut spesifikasinya:

1) *Mainboard* : Acer
2) *Processor* :Intel(R) Core(TM) i5
3) *Memory* : 4.00 GB
4) *DVD* : Super Multi

Software yang digunakan dalam uji coba pada *hardware* di atas memiliki spesifikasi, yaitu:

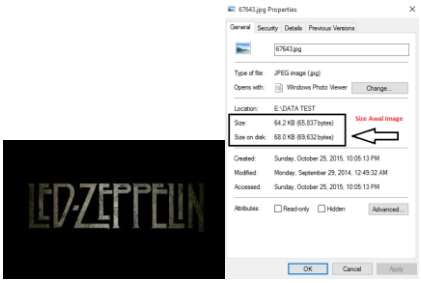
1) *Operating System* : Microsoft Windows 10 64-bit
2) Bahasa Pemrograman : *Java*

4.4. Hasil Pengujian Aplikasi

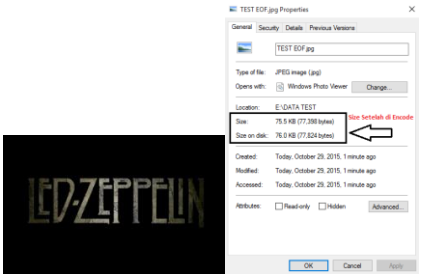
a. Metode EOF (*End Of File*) dan

***Algoritma* DES (*Data Encryption Standard*)**

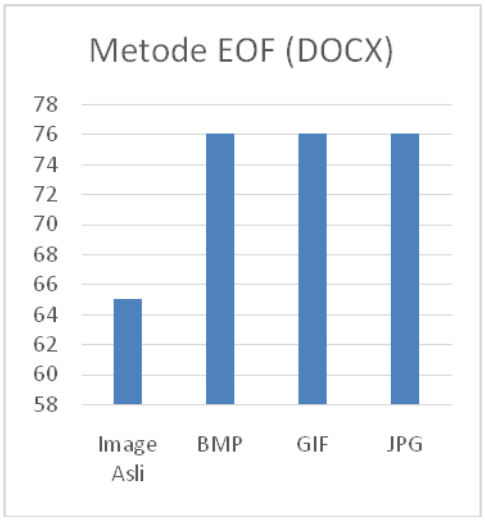
Berikut merupakan perbandingan gambar antara image asli dengan image yang sudah di steganografi dengan metode EOF, terdapat perbedaan size antara image asli dengan image yang sudah disteganografi tetapi secara tampilan tidak berubah, kedua gambar dapat dilihat dibawah ini:



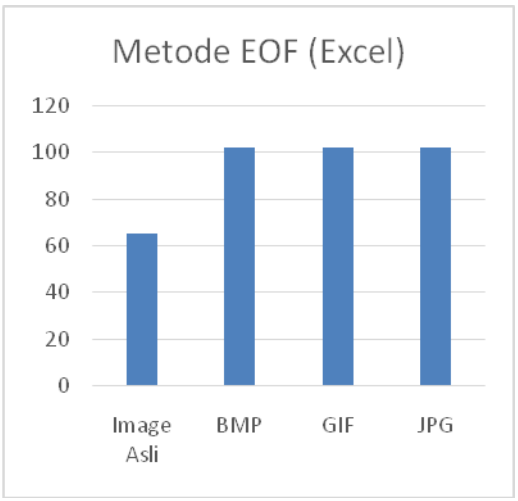
Gambar 23 Tampilan Image Asli EOF (End Of File)



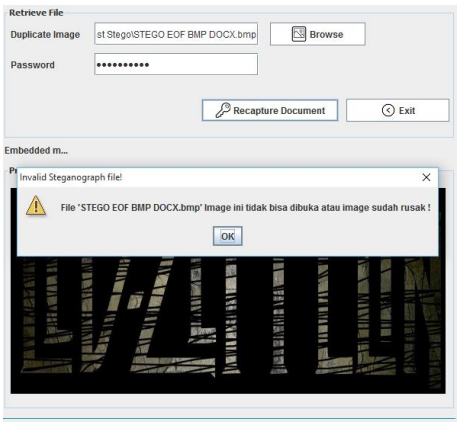
Gambar 24 Tampilan Image Steganografi EOF (End Of File)



Gambar 25 Tampilan Perbandingan Image EOF (End Of File) DOCX

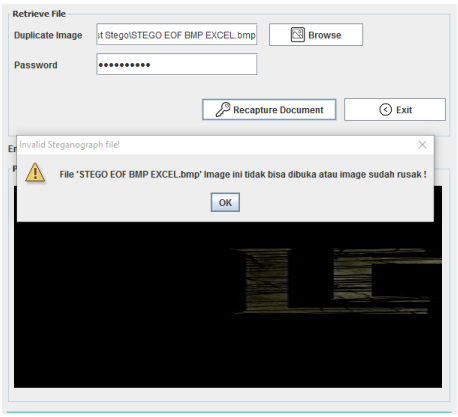


Gambar 26 Tampilan Perbandingan Image EOF (End Of File) EXCEL



Gambar 27 Tampilan Layar

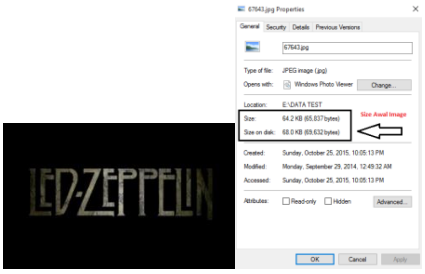
Setelah *Image Steganografi* rusak EOF (*End Of File*)



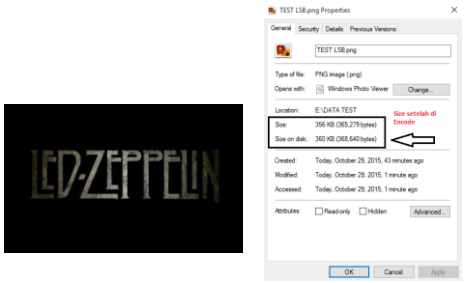
Gambar28 Tampilan Layar Setelah Image Steganografi Crop EOF (End Of File)

b. Metode LSB (*Least Significant Bit*) *Algoritma* Subsitusi (*Shift Chiper*)

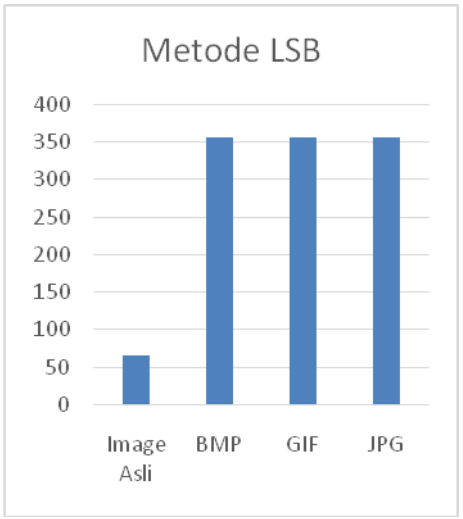
Berikut merupakan perbandingan gambar antara image asli dengan image yang sudah di steganografi dengan metode LSB, terdapat perbedaan size antara image asli dengan image yang sudah disteganografi tetapi secara tampilan tidak berubah, kedua gambar dapat dilihat dibawah ini:



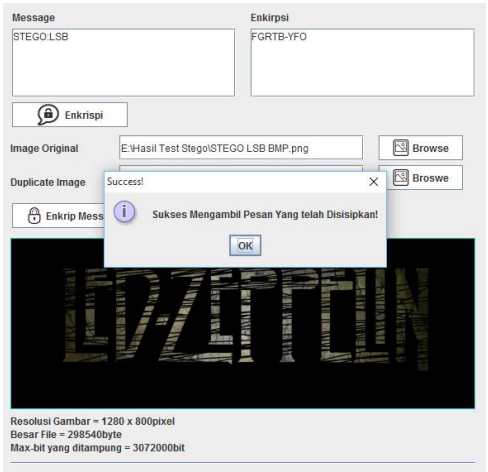
Gambar 29 Tampilan Image Asli LSB (Least Significant Bit)



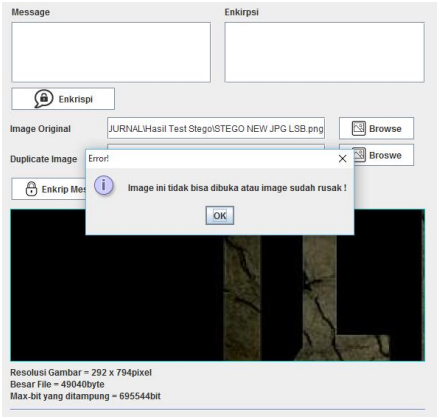
Gambar 30 Tampilan Image Steganografi LSB (Least Significant Bit)



Gambar 29 Tampilan Perbandingan Image LSB (Least Significant Bit)



Gambar 30 Tampilan Layar Setelah Image Steganografi rusak LSB (Least Significant Bit)



Gambar 31 Tampilan Layar Setelah Image Steganografi Crop LSB (Least Significant Bit)

4.5. Kelebihan dan Kekurangan Program

a. Kekurangan Program

1) Metode LSB (*Least Significant Bit*) image yang sudah dipotong atau di *cropping* mengalami kehancuran pesan yang disisipkan tidak bisa ditampilkan.
2) Metode EOF (*End Of File*) image yang sudah dipotong atau di *cropping* mengalami kehancuran pesan yang disisipkan tidak bisa ditampilkan.
3) Metode LSB (*Least Significant Bit*) jika disisipkan pesan maka size jauh lebih besar dibandingkan dengan image asli.
4) Metode EOF (*End Of File*) tidak bisa mengunggah PNG.
5) Output Image dengan metode LSB (*Least Significant Bit*) hanya bisa jpg.

b. Kelebihan Program

1) Metode EOF (*End Of File*) dokumen yang disisipkan ke dalam sebuah gambar dapat lebih besar ukurannya dari image nya.
2) Metode LSB (*Least Significant Bit*) image yang sudah dirubah bentuknya (misal coret) akan tetap menampilkan pesan.
3) Gambar asli dan gambar yang telah disisipkan tidak mengalami perubahan dalam segi tampilan.
4) Menjalankan aplikasi tidak memakan waktu lama walaupun image yang dipakai memiliki ukuran size yang cukup besar.

5. PENUTUP

5.1. Kesimpulan

Adapun kesimpulan yang diperoleh penulis dari perancangan, pembuatan serta uji coba dan analisa program aplikasi *steganografi* ini, maka penulis dapat membuat kesimpulan berikut kesimpulan antara lain:

a. Aplikasi *steganografi* memberikan suatu hal yang menarik untuk diterapkan bagi institusi-institusi yang berkepentingan.

b. Aplikasi *steganografi* memberikan keamanan bagi dokumen-dokumen, file dan pesan yang bersifat privasi atau rahasia.

c. Dengan metode LSB (*Least Significant Bit*) dan EOF (*End Of File*)ini, image yang disisipkan pesan, file dan dokumen dari segi tampilan tidak terlihat perbedaan dengan image aslinya.

5.2. Saran

Selain menarik beberapa kesimpulan, penulis juga mengajukan beberapa saran yang mungkin bisa dijadikan pertimbangan dalam pengembangan aplikasi *steganografi* sebagai berikut:

a. User interface lebih disempurnakan lagi agar tampak lebih menarik dan mempermudah dalam pemakaiannya.

b. Penambahan metode compressi agar ukuran image atau size tidak terlalu besar setelah disisipkan pesan ataupun dokumen.

c. Merubah bentuk (missal cropping) image bisa menampilkan pesan atau dokumen yang telah di *enkripsi.*

d. Bukan hanya pesan atau document saja yang bisa disisipkan ke image (misal voice, video, mp3).

DAFTAR PUSTAKA

[1] Rakmat, Basuki. 2010. *SteganografiMenggunakan Metode Least Significant Bit Dengan Kombinasi Algoritma Kriptografi Vignere dan RC4..*

[2] Nyura, Yusni. 2010. *Pembuatan Aplikasi Bahasa Inggris pada Handphone J2ME.*

[3] Saputro, Fahar Septian Dwi. 2013. *Implementasi Sistem Kemanan Dokumen Teknik Steganografi Demgam Metode EOF (End Of FIle).*

[4] Saputra, Hasbian. 2012 *Implementasi Algoritma Steganografi Embedding Dengan Metode Least Significant Bit (LSB) Insertion Dan Huffman Coding Pada PEngiriman Pesan Menggunakan Media MMS Berbasis J2ME.*

[5] FAIRUZABADI, MUHAMMAD. 2010. *Implementasi Kriptografi Klasik Menggunakan Borland Delphi.*

[6] Ibrahim, Rohmat Nur. 2012.*Kriptografi Algoritma DES, AES/RISNDAEL, Blowfish Untuk Keamanan Citra Digital Dengan Menggunakan Metode Discrete Wavelettra Transformation (DWT).*

[7] **Fadli, Anri. 2015.***Aplikasi Kriptografi Dan Steganografi Menggunakan Algoritma Caesar Chiper Dan Least Significant Bit (LSB)*

[8] Dafid. 2006. *Kriptografi Kunci Simetris dengan Menggunakan Algoritma Crypton.*

[9] *Pengertian, Kelebihan Dan Kekurangan Java*, dilihat pada tanggal01 November 2015<*httprepository.usu.ac.idbitstream123456789272264Chapter%20II.pdf*>.