ISSN : 2087 - 2062

EXPLORE

Jurnal Sistem Informasi & Telematika

(Telekomunikasi, Multimedia & Informatika)

ISSN 2087-2062

EXPLORE

Jurnal Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia, dan Informasi)
Volume 8, Nomor 2, Oktober 2017

Fakultas Ilmu Komputer
Universitas Bandar Lampung

JIST	Volume 8	Nomor 2	Halaman	Lampung Oktober 2017	ISSN 2087 - 2062

Jurnal Manajemen Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia & Informatika)
Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Bandar Lampung

Alamat Penerbit/Redaksi:

Pusat Studi Teknologi Informasi - Fakultas Ilmu Komputer
Universitas Bandar Lampung
Gedung Business Center lt.2
Jl.Zainal Abidin Pagar Alam no.26 Bandar Lampung
Telp.0721-774626
Email: *explore@ubl.ac.id*

PENGANTAR REDAKSI

Jurnal explore adalah jurnal yang diprakrasai oleh program studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Bandar Lampung, yang di kelola dan diterbitkan oleh Fakultas Ilmu Komputer / Pusat Sudi Teknologi Informasi.

Pada Edisi ini, explore menyajikan artikel/naskah dalam bidang teknologi informasi khususnya dalam pengembangan aplikasi, pengembangan machine learning dan pengetahuan lain dalma bidang rekayasa perangkat lunak, redaksi mengucapkan terima kasih dan selamat kepada penulis makalah ilmiah yang makalahnya kami terima dan di terbitkan dalam edisi ini, makalah ilmah yang ada dalam jurnal ini memberikan kontribusi penting pada pengembangan ilmu dan teknologi.

Selain itu, sejumlah pakar yang terlibat dalam jurnal ini telah memberikan kontribusi yang sangat berharga dalam menilai makalah yang dimuat, oleh sebab itu, redaksi menyampaikan banyak terima kasih.

Pada kesempatan ini redaksi kembali mengundang dan memberikan kesempatan kepada para peneliti, di bidang pengembangan perangkat lunak untuk mempublikasikan hasil penelitiannya dalam jurnal ini.

Akhirnya redaksi berharap semoga makalah dalam jurnal ini bermanfat bagi para pembaca khususnya bagi perkembangan ilmu dan teknologi dalam bidang perekaan perangkat lunak dan teknologi pada umumnya.

REDAKSI

TEKNIK STEGANOGRAFI UNTUK PENYEMBUNYIAN PESAN TEKS MENGGUNAKAN ALGORITMA *END OF FILE*

Dedi Darwis[1], Kisworo[2]
Program Studi Manajemen Informatika
AMIK Teknokrat Lampung
Jl. Zainal Abidin Pagar Alam No. 9-11 Kedaton Bandarlampung 35142
Telp.(0721) 702022 web www.teknokrat.ac.id
darwisdedi@teknokrat.ac.id, kisworo@teknokrat.ac.id

ABSTRAK

Pengiriman pesan untuk menyampaikan informasi sering dilakukan oleh banyak manusia dalam kemajuan teknologi saat ini. Sehingga penelitian ini dilakukan untuk menghindari terjadinya pencurian maupun sabotase informasi pesan yang dilakukan antara dua belah pihak dan tidak terbaca oleh orang yang tidak diinginkan. Kemudian dalam penelitian ini menggunakan teknik steganografi End Of File yang dimana pesan rahasia akan disisipkan pada media citra digital berformat JPG. Namun menyisipkan pada akhir suatu file citra tersebut. Hasil dari penelitian ini akan menghasilkan stego image yang tidak berubah secara signifikan serta proses pengambilan pesan yang relative cepat sehingga menjadikan alternatif pengiriman pesan agar terhindar dari pencurian dan sabotase.

Kata Kunci: *Steganografi, JPG, Stego Image, Cover Image, EOF.*

1. PENDAHULUAN

Manusia merupakan makhluk sosial yang dimana tidak lepas dari komunikasi sebagai media penyampaian pesan atau informasi kepada orang lain. Komunikasi dapat kita artikan sebagai media untuk berbagi pikiran, informasi, dan intelijen. Segala bentuk aktifitas yang dilakukan kebanyakan masyarakat dengan tujuan menyampaikan pesannya pada orang lain merupakan tujuan komunikasi. Dilatarbelakangi oleh kebutuhan tersebut, manusia dapat melakukan pengiriman pesan dengan mudah dan cepat dimana saja dengan menggunakan berbagai macam media. Pesan juga bisa disampaikan melalui tatap muka (lisan) maupun media komunikasi (tertulis). Perkembangan dunia digital saat ini menjadikan lalu lintas pengiriman pesan atau data semakin pesat. Pertukaran informasi yang pesat didukung juga oleh perkembangan telepon genggam, khususnya *smartphone* menggunakan sistem operasi *android*. Dimana telah diketahui bahwa pengguna *smartphone* dengan sistem operasi *android* telah dikonsumsi jutaan masyarakat diseluruh dunia. Sehingga menjadikan platform *smartphone* terbesar di dunia. Tidak sedikit pula masyarakat didunia melakukan pengiriman pesan menggunakan media seperti *facebook*, *twitter*, dan lain-lain. Dengan mudahnya kita mengirim pesan sehingga bisa mengirim berita atau informasi dengan cepat walau mengirim kepada orang yang tempatnya jauh.

Semua kemudahan tersebut, seringkali seseorang yang hendak mengirim pesan kepada orang lain tidak ingin isi pesan tersebut diketahui oleh orang lain atau bisa disebut dengan istilah *digital attacker* seperti penyadapan. Data yang dipertukarkan pun bervariasi baik dari jenisnya maupun tingkat kerahasiaannya. Mulai dari data pribadi, data organisasi sampai data Negara yang sangat rahasia. Hal inilah yang menuntut adanya pengamanan pesan tersebut sehingga tidak sampai tersadap oleh pihak ketiga. Oleh karena itu, salah satu hal yang dapat dilakukan untuk mengatasi situasi keamanan terhadap pengiriman pesan adalah mengembangkan suatu aplikasi yang mampu menyembunyikan pesan dimana pihak lain tidak dapat mengetahui pesan rahasia dan hanya diakses oleh orang yang diinginkan atau penerima yang tepat.

Steganografi merupakan seni maupun teknik yang akan digunakan untuk menyisipkan pesan kedalam sebuah media. Teknik steganografi ini sangat berbeda dengan kriptografi yang dimana hanya mengacak pesan sehingga tidak dapat dimengerti namun pihak ketiga dapat mendeteksi adanya data (*chipertext*), karena hasil dari kriptografi

sendiri berupa data yang berbeda dari bentuk aslinya dan seolah-olah data tersebut berantakan, tetapi dapat dikembalikan ke bentuk semula. Sedangkan steganografi membahas bagaimana sebuah pesan dapat disisipkan ke dalam suatu media baik gambar, audio, video, dll. Sehingga pihak ketiga tidak dapat menyadari hal tersebut karena teknik steganografi memanfaatkan keterbatasan sistem indera manusia seperti mata dan telinga. Sehingga teknik steganografi ini mudah diimplementasikan dengan menggunakan beberapa metode dan salah satunya metode *End Of File* (EOF), dimana metode ini berfokus pada ukuran suatu citra digital untuk menyisipkan pesan pada *file* yang terakhir. Dengan keterbatasan inilah, teknik steganografi dapat diterapkan pada berbagai media digital. Hasil yang dikeluarkan dari steganografi ini memiliki persepsi yang sama seperti bentuk aslinya, dimana dapat diproses oleh komputer tetapi tidak berlaku untuk kemampuan indera manusia.

2. TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka pada Penelitian ini direferensi dari beberapa penelitian sebelumnya yaitu :

a) Krisnawati (2008) meneliti tentang Metode *Least Significant Bit* (LSB) dan *End Of File* (EOF) untuk Menyisipkan Teks ke dalam Citra *Grayscale*. Penggunaan dua metode yaitu metode LSB (*Least Significant Bit*) dan EOF (*End Of File*) dalam penyisipan pesan teks. Hal ini dperlukan karena sering terjadi bahwa pesan teks yang dikirim merupakan suatu pesan rahasia yang tidak boleh diketahui sembarang orang. Penggunaan dua metode ini dilakukan pada media penampung citra *grayscale*. Metode LSB bekerja dengan mengganti bit terakhir kode biner citra dengan kode biner pesan, sedangkan metode EOF bekerja dengan menambahkan teks sebagai nilai derajat keabuan citra pada akhir citra. Kelebihan metode LSB adalah ukuran citra yang mengandung pesan tidak berubah, namun mempunyai kelemahan berupa kapasitas pesan yang terbatas. Sebaliknya metode EOF memiliki kelebihan dapat menyisipkan pesan dalam jumlah yang tidak terbatas, dengan kelemahan kapasitas *file* citra akan bertambah.

b) Wasino, Rahayu (2012) meneliti tentang Implementasi Steganografi Teknik *End Of File* dengan enkripsi *Rijndael*. Keamanan suatu data yang rahasia merupakan suatu tindakan yang bertujuan untuk mengamankan data tersebut dari gangguan pihak lain yang tidak bertanggung jawab terhadap kerahasiaan data. Sehingga diperlukannya suatu teknik dalam proses pengamanan pesan rahasia saat proses pengiriman pesan tersebut sesuai tujuan. Teknik steganografi sangat sesuai dalam melakukan pengamanan pesan dengan menggunakan algoritma *End Of File* yang merupakan suatu teknik dengan cara menambahkan data atau pesan rahasia pada akhir *file* dan pesan yang akan disisipkan tidak terbatas sesuai keinginan namun disesuaikan dengan media penampung sehingga tidak mengalami perubahan ukuran citra penampung yang sangat signifikan dan dapat memberikan kecurigaan bagi pihak ketiga. Kemudian pesan yang akan disisipkan nanti akan dilakukan proses enkripsi yaitu teknik kriptografi, hal ini dilakukan untuk menambah pengaman data dengan menggunakan algoritma *Rijndael*.

c) Sukrisno, Utami (2007) meneliti tentang Implementasi Steganografi Teknik EOF dengan Gabungan Enkripsi Rijndael, Shift Cipher dan Fungsi Hash MD5. Pertukaran informasi melalui jaringan internet sering dilakukan banyak masyarakat sekarang ini dengan perkembangan teknologi yang ada. Perkembangan tersebut, menjadikan kejahatan teknologi komunikasi dan informasi juga turut berkembang seperti yang sering kita dengan bisa disebut dengan *hacker, cracker*, *carder*, *phreaker*, dan sebagainya. Ancaman yang datang menjadi sorotan bagi para pengguna internet seperti interupsi, penyadapan, modifikasi maupun fabrikasi. perkembangan teknik steganografi ini menjadi salah satu alternatif pengamanan dalam komunikasi data di jaringan internet. Berbeda dengan kriptografi, kecurigaan terhadap pesan yang disamarkan mudah dikenali karena pesan disamarkan dengan cara mengubah pesan yang asli menjadi seolah-olah tidak

terbaca. Sedangkan steganografi lebih mengurangi kecurigaan saat bertukar pesan dalam komunikasi di jaringan internet. Penggunaan kriptografi kemudian dilakukan untuk enkripsi pesan dimana plaintext diubah menjadi chipertext yang dimana proses enkripsi menggunakan algoritma *rijndael*, MD5, dan *shift cipher*.

d) Iswahyudi, Setyaningsih(2012) meneliti tentang Pengamanan Kunci Enkripsi Citra pada Algoritma Super Enkripsi Menggunakan Metode *End Of File.* Keamanan informasi menjadi isu yang sangat penting dalam penyimpanan dan transmisi data. Salah satu pengamanan tersebut menggunakan teknik kriptografi yang dimana melakukan penyamaran pesan asli yang akan dikirimkan menjadi pesan yang tidak beraturan atau tidak dapat dimengerti maupun dibaca. Algoritma yang dikembangkan pada penelitian ini menggunakan konsep *symmetric cryptosystem* yang dimana sangat menekankan pada kerahasiaan kunci yang digunakan untuk proses enkripsi dan dekripsi. Sehingga sistem ini sering disebut sebagai *secret-key cryptography* dimana merupakan bentuk kriptografi yang lebih tradisional, sebuah kunci tunggal digunakan untuk proses enkripsi dan dekripsi. Pengirim maupun penerima saling memiliki kunci yang sama, namun masalah utama yang dihadapi adalah bagaimana pengirim dan penerima menyetujui kunci rahasia tanpa ada orang lain yang mengetahuinya. Sehingga diusulkan sebuah cara untuk mengoptimalkan keamanan pada kunci yang digunakan. Teknik yang diusulkan adalah mengadopsi dari konsep steganografi menggunakan metode *end of file*.

2.1 Landasan Teori

2.1.1 Definisi Steganografi

Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi rahasia di dalam suatu informasi lainya (Ariyus, 2008).

Menurut Sembiring (2013) mengemukakan ada beberapa hal yang diperlukan untuk menyembunyikan pesan yaitu:

1. Algoritma Penyisipan (*Embeding Algorithm*)
   Algoritma ini digunakan untuk menyisipkan suatu pesan yang disembunyikan ke dalam suatu data yang akan dikirim. Proses penyisipan ini diproteksi oleh sebuah *key-word* sehingga hanya orang-orang yang mengetahui *key-word* ini yang dapat membaca pesan yang disembunyikan tersebut.
2. Fungsi Detektor (*Detector Function*)
   Fungsi Detektor ini adalah untuk mengembalikan pesan-pesan yang disembunyikan tersebut.
3. *Carrier Document*
   Merupakan dokumen yang berfungsi sebagai media yang digunakan untuk menyisipkan informasi. Dokumen ini dapat berupa file-file seperti file audio,video atau citra(gambar).
4. *Key*
   Merupakan kata kunci yang ikut disisipkan kedalam dokumen berguna dan dipakai sebagai proses verifikasi sewaktu informasi akan ditampilkan atau diuraikan.
5. *Secret Message/ Plaintext*
   Merupakan pesan rahasia yang akan disisipkan kedalam carrier document. Pesan inilah yang tidak terlihat dan terbaca orang yang tidak berkepentingan.

2.1.2 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Secara harafiah, citra (*Image*) adalah gambar pada bidang dwimatra (2 dimensi). Ditinjau dari sudut pandang sistematis, citra merupakan fungsi *continue* dari intensitas cahaya pada bidang dwimatra (2D). (Sembiring, 2013).

Terdapat tiga aspek yang perlu diperhatikan dalam menyembunyikan pesan:kapasitas, keamanan, dan ketahanan. Kapasitas merujuk kepada besarnya informasi yang dapat disembunyikan oleh media, keamanan merujuk kepada ketidakmampuan pihak lain untuk mendeteksi keberadaan informasi yang disembunyikan, dan ketahanan merujuk kepada sejauh mana medium *steganography* dapat bertahan sebelum pihak lain menghancurkan informasi yang disembunyikan. (Darwis, 2015)

2.1.3 Algoritma *End Of File (EOF)*

Konsep EOF yang dikemukakan pada penelitian Iswahyudi dan Setyaningsih (2012) yaitu pesan disisipkan diakhir file citra, dengan

metode ini pesan yang disisipkan jumlahnya tak terbatas. Akan tetapi efek sampingnya adalah ukuran file menjadi lebih besar dari ukuran semula. Ukuran file yang terlalu besar dari yang seharusnya, tentu akan menimbulkan kecurigaan bagi yang mengetahuinya. Oleh karena itu dianjurkan agar ukuran pesan dan ukuran citra yang digunakan proporsional.
Proses penyisipan pesan dengan metode EOF dapat dituliskan dalam algoritma sebagai berikut:

1. Inputkan pesan yang akan disisipkan.
2. Ubah pesan menjadi kode desimal.
3. Inputkan citra grayscale yang akan disisipi pesan.
4. Dapatkan nilai derajat keabuan masing-masing piksel.
5. Tambahkan kode desimal pesan sebagai nilai derajat keabuan diakhir citra.
6. Petakan menjadi citra baru.

Sedangkan ekstraksi pesan yang sudah disisipkan dengan metode EOF dapat dilakukan dengan algoritma berikut:

1. Inputkan image yang sudah mengandung pesan.
2. Dapatkan nilai derajat keabuan citra tersebut.
3. Ubah nilai tersebut menjadi karakter pesan.

2.1.4 PSNR (*Peak Signal to Noise Ration*) dan MSE (*Mean Square Error*)

Menurut penelitian yang dilakukan oleh Darwis (2015) PNSR merupakan parameter yang digunakan mengukur kualitas citra yang dihasilkan. Metode PNSR adalah ukuran perbandingan antara nilai piksel *cover image* dengan nilai piksel pada citra *stego* yang dihasilkan. Sebelum menentukan PNSR terlebih dahulu ditentukan nilai rata-rata kuadrat *absolute error* antara *cover image* dengan citra *stego* yaitu nilai MSE (*Mean Square Error*). Berikut ini rumus MSE untuk *cover image* berwarna:

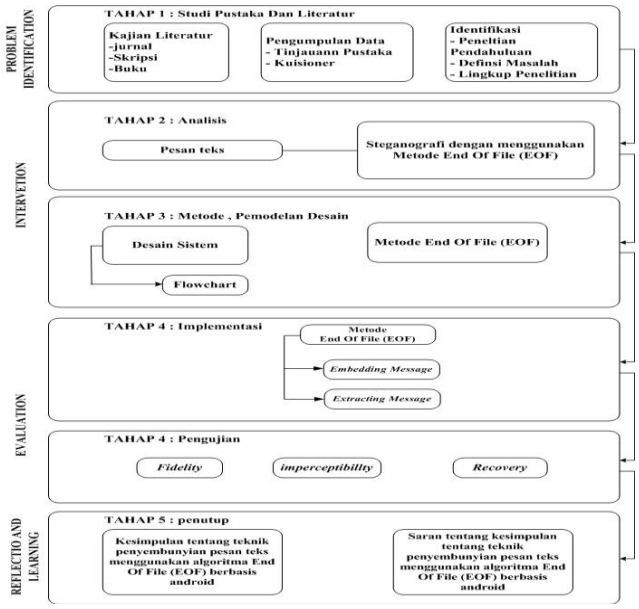
$$MSE_{AVG} = \frac{MSE_R + MSE_G + MSE_B}{X.Y}$$

Keterangan:
PNSR = Nilai PNSR citra digital.
MSE = Nilai *Mean Square Error* dari citra.
Citra *stego* dapat dikatakan berkualitas baik jika nilai PNSR dari citra *stego* tersebut bernilai tinggi. Terdapat sedikit perbedaan antara *cover image* dan citra *stego* setelah penambahan pesan rahasia. Tingkatan kualitas nilai PNSR berbanding terbalik dengan nilai MSE, semakin tinggi nilai PNSR semakin rendah nilai MSE. Semakin tinggi kualitas yang dihasilkan dari citra *stego* maka semakin rendah nilai dari MSE.

3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

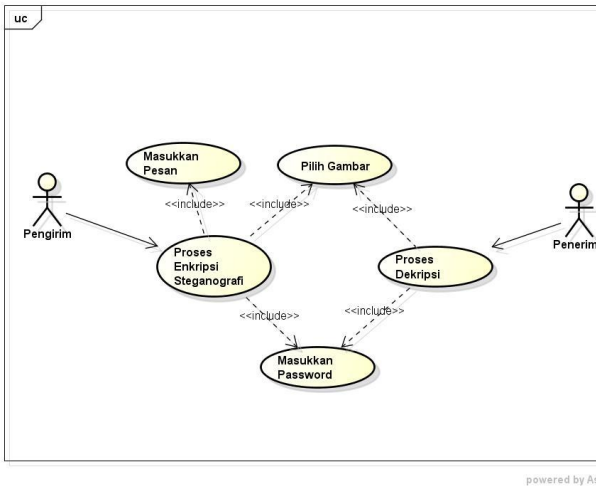
Tahapan penelitian terbagi menjadi beberapa sub menu bagian. Tahapan penelitian yang peneliti lakukan dapat dilihat pada gambar 1



Gambar 1. Tahapan Penelitian

3.2 *Use Case* Diagram

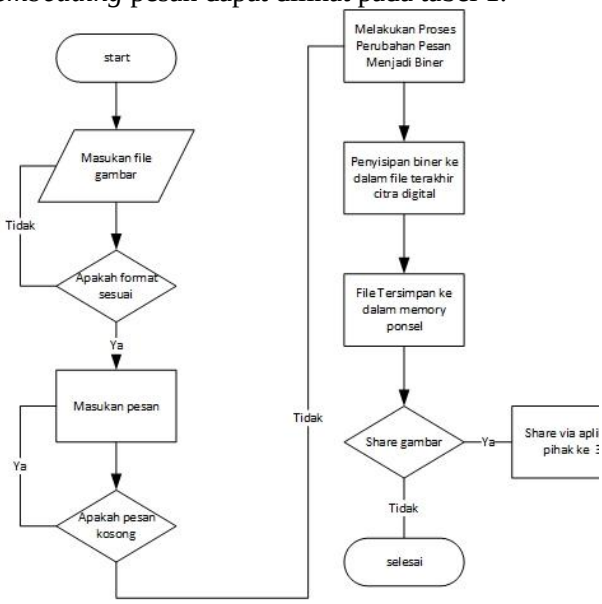
Use case diagram merupakan salah satu diagram dalam bahasa pemodelan UML yang dapat menggambarkan kegiatan yang dilakukan oleh *actor* secara garis besar, dan hubungan antara *actor* dengan setiap kegiatan (*actor – use case*) atau hubungan antara kegiatan (*use case – use case*). Gambaran atau model dari pembuatan penelitian ini dapat dilihat pada gambar 2.



Gambar 2. *Use Case* Diagram

3.3 *Flowchart Embedding Pesan*

Pada proses *Embedding* pesan dimulai dari menginputkan gambar yang dapat diambil dari kamera secara langsung maupun gambar yang sudah ada pada *gallery*, kemudian proses *embedding* dimulai dengan membaca nilai file suatu gambar setelah itu pesan yang akan disisipkan dibaca nilai nya dalam bentuk decimal. Kemudian nilai tersebut dimasukkan diakhir suatu file image tersebut. Hasil *output file image* disimpan ke lokasi yang telah ditentukan. *Flowchart embedding* pesan dapat dilihat pada gambar 3 berikut dan proses *embedding* pesan dapat dilihat pada tabel 1.



Gambar 3. *Flowchart Embedding* Pesan

Tabel 1. Proses *Embedding*

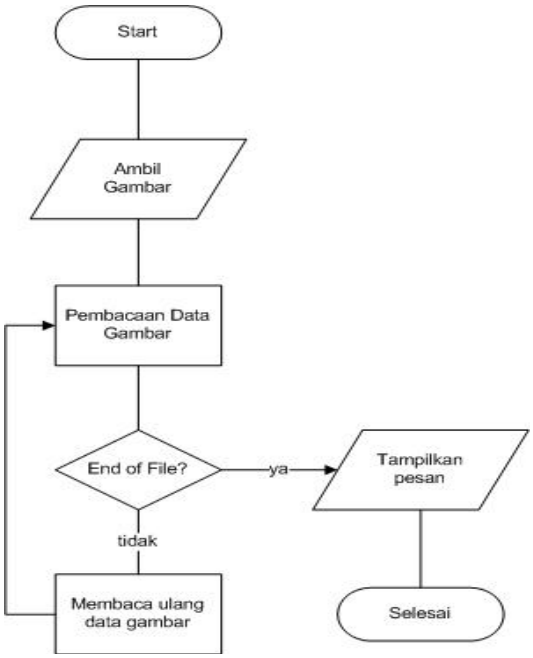
COVER IMAGE				
201	76	88	35	17
138	200	221	140	87
78	76	54	14	211
89	80	78	65	90
88	76	34	111	133

Desimal Pesan		
65	107	117

STEGO IMAGE				
201	76	88	35	17
138	200	221	140	87
78	76	54	14	211
89	80	78	65	90
88	76	34	111	133
65	107	117		

3.4 *Flowchart Extracting Pesan*

Pada proses *extracting* pesan dimulai dari menginputkan gambar hasil steganografi yang dimana diambil nilai derajatnya berupa bilangan decimal, dan mencari nilai akhir sebuah file dimana pesan rahasia dapat dibaca dalam bentuk bilangan decimal setelah itu konversi dalam standar kode ASCII yang menghasilkan pesan yang dapat dibaca oleh penerima. Flowchart ektraksi pesan dapat dilihat pada gambar 4.



Gambar 4. *Flowchart* Ektraksi Pesan

Proses yang dilakukan pada extracting pesan dapat dilihat pada tabel 2.

Tabel 2. Proses Ekstraksi Pesan

STEGO IMAGE				
201	76	88	35	17
138	200	221	140	87
78	76	54	14	211
89	80	78	65	90
88	76	34	111	133
65	107	117		

Desimal Pesan		
65	107	117

Kemudian dikonversikan menjadi sebuah pesan yaitu :

65 = A
107 = k
117 = u
} ASCII CODE

Sehingga menjadi sebuah pesan dengan bunyi **Aku.**

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Steganografi *End Of File* (EOF)

Pengujian steganografi digunakan untuk melihat keberhasilan aplikasi dalam melakukan penyisipan pesan pada *cover image*. Penyisipan pesan teks dilakukan pada *cover image* dengan memperhatikan ukuran file suatu gambar, dimana pesan yang akan disisipkan diubah menjadi bilangan decimal dan kemudian ditambahkan dengan file gambar dibagian akhir.

4.2 Skenario Pengujian

Pengujian dilakukan dengan 2 skenario yakni pengujian tanpa serangan dan pengujian dengan serangan. Pada pengujian tanpa serangan, dilakukan dengan menguji kwalitas dari masing-masing citra setelah mengalami proses *steganography* seperti pengujian *imperectibility*, *fidelity*, dan *recovery*. Kemudian skenario pengujian dengan serangan, yaitu memberikan serangan terhadap citra yang telah disteganografi dengan dilakukan pemotongan maupun perputaran pada *stego image* (*robustness*).

4.3 Pengujian Tanpa Serangan

Pengujian ini dilakukan tanpa serangan yang dimana hanya mengukur dari segi kualitas dan mutu yang tertuang dalam nilai-nilai, pengujiannya sebagai berikut :

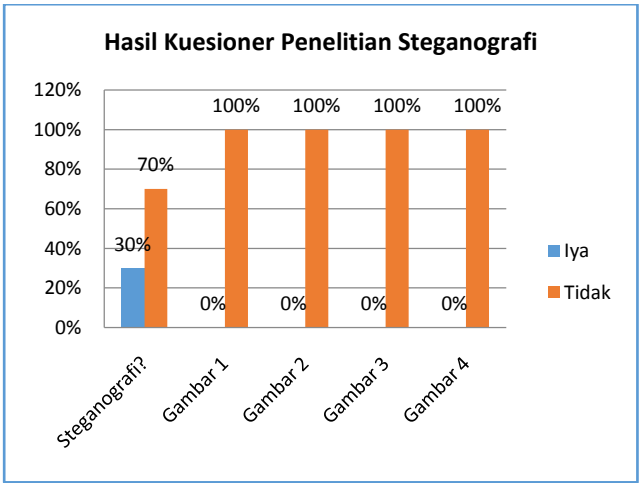
4.3.1 Imperectibility

Pengujian *imperectibility* bertujuan untuk mengetahui seberapa mudah sebuah *stegano-image* dapat terdeteksi oleh indra manusia. Pengujian ini telah dilakukan secara manual dengan melibatkan 30 responden mahasiswa dibidang komputer yang diminta untuk menjawab beberapa pertanyaan dan melihat gambar hasil steganografi. Hasil kuisioner untuk pengujian ini dapat dilihat pada tabel 3.

Tabel 3. Hasil Kuisioner Pengujian *Imperectibility*

No	Soal	Iya	Tidak
1	Pertanyaan Steganografi?	9	21
2		0	30
3		0	30
4		0	30
5		0	30

Kemudian dari hasil kuesioner yang dilakukan dan diperoleh menjadi nilai pembuktian dengan menunjukan melalui hasil grafik pada gambar 5.



Gambar 5. Grafik Hasil Pengujian *Imperectibility*

Analisis hasil pengujian :

Sesuai dengan skenario pengujian, hasil dari pengujian *imperectibility* membuktikan bahwa *stego-image* yang dihasilkan dengan metode *end of file* tidak memberikan perubahan signifikan dikarenakan gambar hasil steganografi ini hanya mengubah ukuran gambar bukan merubah piksel maupun intensitas warna sehingga dapat disimpulkan secara kasat mata, indera manusia tidak dapat mendeteksi perubahan gambar tersebut.
Dari semua responden yang dimana merupakan mahasiswa yang mengambil bidang ilmu computer, dilihat dari hasil grafik menunjukan bahwa dari 30 responden menghasilkan 30% menjawab "Iya" dan 70% menjawab "Tidak". Sehingga dapat disimpulkan banyak mahasiswa terutama dibidang komputer, kurang mengetahui apa itu "*Steganography*".

4.3.2 Fidelity

Pengujian ini dilakukan untuk melihat mutu media penampung tidak mengalami banyak perubahan akibat penyisipan. Pada pengujian *fidelity* ini akan dilakukan pengujian :

1. Pengujian dengan menggunakan 10 gambar media penampung dengan ukuran yang berbeda antara 10kb – 100kb dengan pesan yang disisipkan 500 karakter
2. Pengujian kedua dilakukan dengan media penampung yang sama namun dengan jumlah pesan yang berbeda.
3. Pengujian PSNR dan MSE untuk melihat hasil mutu media penampung sebelum dan sesudah disisipkan pesan.

Tabel 4. Hasil Pengujian *Fidelity* Pertama

Gambar	Ukuran (kb, bytes)	Karakter	Ukuran karakter (kb, bytes)	Ukuran *Stego* (kb, bytes)
	50,0 51,284	500	0,4882 500	50,5 51,784
	10,4 10,699	500	0,4882 500	10,9 11,199
	39,9 40,945	500	0,4882 500	40,4 41,445
	20,0 20,48	500	0,4882	20,4 20,98
	80,4 82,414	500	0,4882 500	80,9 82,914
	60,3 61,797	500	0,4882 500	60,8 62,297
	99,6 102,052	500	0,4882 500	100 102,552
	70,4 72,154	500	0,4882 500	70,9 72,654
	30,4 31,196	500	0,4882 500	30,9 31,696
	89,7 91,931	500	0,4882 500	90,2 92,431

Analisis hasil pengujian :
Pada tabel 4 menunjukkan pengujian terhadap 10 gambar yang berbeda dengan pesan yang disisipkan menggunakan jumlah karakter yang sama yaitu 500 karakter dengan perhitungan 1 karakter bernilai 1 byte jadi pesan yang ditambahkan berukuran 500 byte, jika dijadikan kb(*kilobyte*) dibagi dengan nilai 1024 (nilai baca komputer) menjadi 0,4882..kb, sehingga memperoleh hasil yang dapat disimpulkan bahwa pesan yang disisipkan hanya menambahkan ukuran dari file gambar tanpa merubah intensitas warna pesan dalam tiap piksel dan dengan pesan 500 karakter dapat terlihat perubahan yang signifikan terhadap ukuran file gambar.

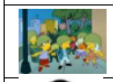
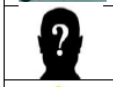
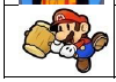
Tabel 5. Hasil Pengujian *Fidelity* Kedua

Gambar	Ukuran (kb, bytes)	Karakter	Ukuran karakter (kb, bytes)	Ukuran *Stego* (kb, bytes)
	70,4 72,154	10	0,0097 10	70,4 72,164
	70,4 72,154	20	0,0195 20	70,4 72,174
	70,4 72,154	30	0,0292 30	70,4 72,184
	70,4 72,154	40	0,0390 40	70,4 72,194
	70,4 72,154	50	0,0488 50	70,5 72,204
	70,4 72,154	60	0,0585 60	70,5 72,214
	70,4 72,154	70	0,0683 70	70,5 72,224
	70,4 72,154	80	0,0781 80	70,5 72,234
	70,4 72,154	90	0,0878 90	70,5 72,244
	70,4 72,154	100	0,0976 100	70,5 72,254

Analisis hasil pengujian :
Pengujian terhadap 10 gambar yang sama dengan pesan yang disisipkan berbeda 10 – 100 karakter dengan perhitungan 1 karakter bernilai 1 byte jadi pesan yang ditambahkan berukuran 10 - 100 bytes, jika dijadikan kb(*kilobyte*) dibagi dengan nilai 1024 (nilai baca komputer) menjadi 0,0097kb – 0,0976kb, sehingga memperoleh hasil yang dapat disimpulkan bahwa pesan yang disisipkan dengan jumlah sedikit antara 10 – 100 karakter terlihat ukuran gambar tidak mengalami perubahan besar pada ukuran file gambar.

Pengujian MSE (*Mean Squre Error*) dan PSNR (*Peak Signal to Noise Ratio*) digunakan untuk mengukur kualitas citra yang dihasilkan. Metode MSE adalah nilai error kuadrat rata-rata antara citra asli dengan citra manipulasi (dalam kasus steganografi). Sedangkan PSNR merupakan ukuran perbandingan antara nilai piksel *cover image* dengan nilai piksel pada citra *stego* yang dihasilkan. Pada tabel 6 menunjukkan hasil pengujian MSE.

Tabel 6. Hasil Pengujian MSE(*Mean Squre Error*)

Nama Gambar	Panjang String (Huruf)	Ukuran Gambar Asli(kb)	Ukuran *Stego Image*(kb)	MSE
	500	50,0 51,284	50,5 51,784	0
	500	10,4 10,699	10,9 11,199	0
	500	39,9 40,945	40,4 41,445	0
	500	20,0 20,486	20,4 20,986	0
	500	80,4 82,414	80,9 82,914	0
	500	60,3 61,797	60,8 62,297	0
	500	99,6 102,052	100 102,552	0
	500	70,4 72,154	70,9 72,654	0
	500	30,4 31,196	30,9 31,696	0
	500	89,7 91,931	90,2 92,431	0

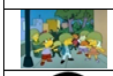
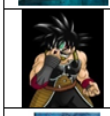
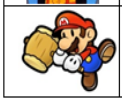
Analisis hasil pengujian :
Setelah dilakukan perhitungan dengan rumus matematka MSE (*Mean Squre Error*) dimana untuk mengetahui seberapa banyak *error* yang dialami saat disisipkan pesan pada *cover image*, akan tetapi dari hasil yang diperoleh bahwa nilai MSE dari semua gambar dengan pesan yang disisipkan berjumlah 500 karakter bernilai 0. Dikarenakan metode ini hanya menyisipkan pesan pada file gambar di akhirnya, bukan pada intensitas warna RBG suatu piksel sehingga tidak merubah maupun merusak nilai piksel gambar.

Berikut ini adalah rumus yang digunakan untuk menghitung PSNR.

$$PSNR = 10_{log\,10}\left(\frac{255^2}{MSE}\right)$$

Keterangan :
PSNR = Nilai PSNR citra Digital
MSE = Nilai *Mean Squre Error* dri citra

Tabel 7. Pengujian PSNR (*Peak Signal to Noise Ratio*)

Nama Gambar	Panjang String (Huruf)	Ukuran Gambar Asli(kb)	Ukuran *Stego Image*(kb)	PSNR
	500	50,0 51,284	50,5 51,784	∞
	500	10,4 10,699	10,9 11,199	∞
	500	39,9 40,945	40,4 41,445	∞
	500	20,0 20,486	20,4 20,986	∞
	500	80,4 82,414	80,9 82,914	∞
	500	60,3 61,797	60,8 62,297	∞
	500	99,6 102,052	100 102,552	∞
	500	70,4 72,154	70,9 72,654	∞
	500	30,4 31,196	30,9 31,696	∞
	500	89,7 91,931	90,2 92,431	∞

Analisis hasil pengujian :
Setelah dilakukan perhitungan dengan rumus matematka MSE (*Mean Squre Error*) setelah itu melakukan perhitungan PSNR (*Peak Signal to Noise Ratio*) dan telah diketahui hasil PSNR di tabel atas menunjukan nilai tak terhingga dikarenakan nilai MSE sebelumnya semua gambar mendapatkan nilai 0. Sehingga pada rumus PSNR menunjukan pembagian terhadap MSE, jika nilai dibagi oleh nilai 0 maka akan menghasilkan nilai tak terhingga gambar. Disimpulkan bahwa hasil PSNR tidak menunjukan nilai yang membuktikan bahwa hasil penyisipan pesan kedalam gambar menggunakan metode *end of file* tidak membuat *noise ratio* maupun kerusakan pada intensitas citra tersebut.

4.3.3 Pengujian *Recovery*

Pengujian *recovery* dilakukan untuk menguji apakah pesan rahasia yang disisipi pada sebuah citra digital harus dapat dipisahkan kembali dari stego-image-nya. Pengujian dapat dilakukan dengan melihat keutuhan pesan yang diekstraksi dari sejumlah citra uji. Berikut *recovery* yang dilakukan.

Pada tabel 8 menunjukkan hasil pengujian recovery.

Tabel 8. Hasil Pengujian *Recovery*

No	Nama Gambar	Normal	*Recovery*
1		✓	Berhasil
2		✓	Berhasil
3		✓	Berhasil
4		✓	Berhasil
5		✓	Berhasil
6		✓	Berhasil
7		✓	Berhasil
8		✓	Berhasil
9		✓	Berhasil
10		✓	Berhasil

Analisis hasil pengujian :
Dari hasil pengujian *recovery* yang telah tertuang dalam laporan pada tabel di atas dapat dilihat bahwa pesan rahasia yang disisipkan pada sebuah citra dapat dipisahkan kembali dari stego-imagenya jika dilakukan secara normal dan tanpa serangan.

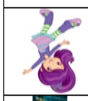
4.4 Pengujian Dengan Serangan

Pengujian ini dilakukan dengan serangan yang berarti melakukan sesuatu terhadap citra yang telah disisipkan pesan, dan untuk pengujiannya yaitu :

4.4.1 Robustness

Pengujian *robustness* dilakukan untuk melihat apakah gambar hasil steganografi yang telah disisipkan pesan rahasia dapat diungkap kembali menjadi pesan rahasia yang akan disampaikan, jika gambar steganografi tersebut akan dilakukan serangan dengan cara memutarkan maupun gambar mengalami pemotongan. Sehingga dalam tabel 8 akan menunjukkan hasil pengujiannya.

Tabel 9. Hasil Pengujian *Robustness*

Nama Gambar	Ukuran Gambar Asli(kb)	Ukuran Gambar Putar(kb)	Ukuran Gambar Crop(kb)	*Recovery*
	50,0 51,284	66,7 68,326	-	Gagal
	10,4 10,699	27,0 27,741	-	Gagal
	39,9 40,945	56,4 57,779	-	Gagal
	20,0 20,486	46,3 47,512	-	Gagal
	80,4 82,414	87,6 88,857	-	Gagal
	60,3 61,797	-	41,5 42,553	Gagal
	99,6 102,052	-	46,9 48,069	Gagal
	70,4 72,154	-	81,4 83,424	Gagal
	30,4 31,196	-	18,0 18,497	Gagal
	89,7 91,931	-	88,5 90,696	Gagal

Analisis hasil pengujian :
Pengujian dengan melakukan serangan dengan cara *robustness* yaitu ketahanan sebuah citra digital terhadap serangan baik perpotongan maupun perputaran. Hasil pengujian tersebut menunjukan bahwa pada proses perputaran mengakibatkan ukuran pada file citra yang telah disisipkan pesan rahasia mengalami penambahan ukuran file dan berbanding terbalik dengan proses pemotongan citra digital yang mengalami pengurangan ukuran file itu sendiri. Namun dari kesemua pengujian baik pemutaran dan pemotongan citra ditigal mengakibatkan tidak dapat terungkap kembali pesan rahasia yang telah disisipkan sebelumnya.

5. KESIMPULA DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian maka dapat dihasilkan simpulan sebagai berikut :

1. Steganografi dengan menggunakan metode *End of File* (EOF) dapat menyisipkan informasi kedalam media citra digital pada bagian akhir *file* gambar.
2. Pengujian *imperectibility* memberikan hasil steganografi pada gambar, dengan metode kuesioner yang menghasilkan 70% mahasiswa dibidang komputer tidak mengetahui tentang Steganografi dan 100% menyatakan gambar hasil steganografi tidak dapat terlihat oleh indra mata manusia secara kasat mata. Pada proses pengujian tahap *fidelity* tidak nampak nilai MSE yang hanya menghasilkan nilai "0" dan PSNR menghasilkan nilai "∞" (tak hingga) dikarenakan metode yang digunakan menyisipkan pesan di akhir file tanpa merubah nilai intensitas warna pikselnya.
3. Pengujian *recovery* dapat diungkapkan kembali pesan yang disisipkan, sedangkan *robutness* gambar steganografi mengalami penambahan ukuran *file* jika diputar dengan rata-rata mencapai 74,72 % dan berkurang ukuran *file* jika mengalami pemotongan dengan rata-rata mencapai 26,73 %.
4. Pembuatan aplikasi steganografi dapat diterapkan dan dijalankan dengan *mobile smartphone android*.

5.2 Saran

Berdasarkan simpulan dari hasil penelitian yang telah diuraikan, maka saran yang dapat diberikan untuk pengembangan lebih lanjut dari penelitian ini adalah sebagai berikut:

1. Aplikasi steganografi dapat dibangun dengan berbagai macam media penyisipan seperti audio maupun video untuk penelitian selanjutnya.
2. Aplikasi ini diharapkan dapat diterapkan dengan format citra lainnya seperti PNG, GIF, dll. Sehingga menjadikan lebih dinamis.

DAFTAR PUSTAKA

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

[2] Darwis, Dedi., 2015., *Implementasi Steganografi pada Berkas Audio Wav untuk Penyisipan Pesan Gambar Menggunakan Metode Low Bit Coding*., Program Studi Magister Ilmu Komputer Universitas Budi Luhur Jakarta. Jurnal Expert. ISSN : 2088-5555

[3] Iswahyudi, C., Setyaningsih, E., 2012. *Pengamanan Kunci Enkripsi Citra Pada Algoritma Super Enkripsi Menggunakan Metode End Of File*. Jurnal Prosiding Nasional Aplikasi Sains & Teknologi (SNAST) Periode III.

[4] Krisnawati, 2008. Metode Least Significant Bit (LSB) dan End Of File (EOF) untuk Menyisipkan Teks Ke Dalam Citra Grayscale. Jurnal UPN "Veteran" Yogyakarta.

[5] Sembiring, S., 2013. Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End Of File. Jurnal Pelita Informatika Budi Darma, volume : iv, nomor:2.

[6] Sukrisno & Utami, E., 2007. Implementasi Steganografi Teknik End Of File Dengan Gabungan Enkripsi Rijndael, Shift Cipher dan Fungsi Hash MD5. Jurnal Seminar Nasional Teknology.

[7] Wasino & Rahayu., 2012. Implementasi Steganografi Teknik End Of File Dengan Enkripsi Rijndael. Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA). ISSN : 2089-9815.

Redaksi :
Research Of Information Technology Universitas Bandar Lampung
Gedung Business Center Lt. 2
Jl. Zainal Abidin No. 26 Bandar Lampung
Telp. 0721 - 774626
e-Mail : explorer.rit@ubl.ac.id