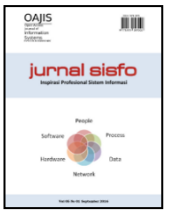
OAJIS
Open Access
Journal of
Information
Systems
is.its.ac.id/pubs/oajis/

jurnal sisfo

Jurnal Sisfo Vol. 06 No. 01 (2016) 1–16



Implementasi Sistem Keamanan File dengan Metode Steganografi EOF dan Enkripsi Caesar Cipher

Bonifacius Vicky Indriyono*

Jurusan Sistem Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer Kadiri (STMIKKA) Kediri

Abstract

Rapid development of data communication raises concern in safety and confidentiality of information. One of those is risk of the confidential information taken by unauthorized parties. Maintaining the security and confidentiality of data in the field of information technology can be made by applying the techniques of cryptography and steganography. Cryptography is the science and art to maintain the confidentiality of the data, while steganography is a science and an art that is used to hide messages in a medium. The medium could be an image, audio, or video. Purpose of this study is combining the Caesar cipher algorithm and steganography End Of File (EOF) into one application. The test phase begins by scrambling the contents of files of type text with Caesar cipher method. After the contents of the file is scrambled, the file is inserted into the image that has been selected. To be able to read the message again, the message is decrypted and the decrypted result will be stored in a new text file.

Keywords: Steganography, Cryptography, End Of File, Caesar cipher, Encryption

Abstrak

Meningkatnya perkembangan komunikasi data membuat aspek keamanan dan kerahasiaan informasi menjadi hal yang penting untuk diperhatikan. Karena banyak informasi yang bersifat rahasia yang beresiko diambil oleh pihak yang tidak berkepentingan. Untuk menjaga keamanan dan kerahasiaan data dalam bidang teknologi informasi dapat dilakukan dengan menerapkan teknik kriptografi dan steganografi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan data, sedangkan steganografi adalah ilmu dan seni yang digunakan untuk menyembunyikan pesan ke dalam suatu media. Media tersebut dapat berupa gambar, audio, atau video. Dalam penelitian ini, akan dibuat sebuah aplikasi yang mengkombinasikan algoritma *Caesar cipher* dan steganografi *End Of File* (EOF). Tahap uji coba dimulai dengan melakukan pengacakan isi file bertipe teks dengan metode *Caesar cipher*. Setelah isi file teracak, selanjutnya file disisipkan ke dalam gambar yang telah terpilih. Untuk dapat membaca pesan kembali, maka pesan didekripsi dan hasil dekripsi akan disimpan pada file teks yang baru.

Kata kunci: Steganografi, Kriptografi, *End Of File*, *Caesar cipher*, Enkripsi

Histori Artikel : Disubmit 11 Maret 2016; Diterima 24 Juni 2016; Tersedia online 11 Agustus 2016

*Corresponding Author
Email address: bonifaciusvicky@gmail.com (Bonifacius Vicky Indriyono)

1. Pendahuluan

Pesatnya perkembangan ilmu pengetahuan dan teknologi dalam dunia informatika membuat faktor pengamanan data menjadi meningkat. Hal ini disebabkan tingginya tingkat kerawanan data maupun informasi penting dapat diakses oleh pihak-pihak yang tidak berkepentingan sehingga data maupun informasi harus diamankan. Terdapat berbagai bentuk pesan rahasia seperti pesan teks (dalam bentuk file), pesan citra, pesan audio dan pesan video yang umum digunakan. Banyak cara atau metode yang digunakan untuk mengamankan informasi atau data agar tidak jatuh ke tangan pihak-pihak yang tidak berkepentingan. Sebagai contoh adalah teknik kriptografi.

Kriptografi adalah ilmu yang digunakan untuk menjaga keamanan dari pihak yang tidak memiliki hak akses terhadap suatu data baik data berupa e-mail, dokumen, maupun berkas pribadi [1]. Selain memiliki sisi positif dari segi keamanan, kriptografi dapat menimbulkan kecurigaan pada orang yang membaca data atau informasi yang terenkripsi. Hal ini yang memungkinkan pihak-pihak yang tidak berkepentingan berusaha untuk memecahkan enkripsi tersebut walau membutuhkan waktu yang cukup lama. Terdapat banyak teknik kriptografi yang dapat digunakan salah satunya adalah teknik subtitusi *Caesar cipher*. Prinsip kerja dari *Caesar cipher* ini adalah substitusi dimana setiap huruf pada teks terang (plaintext) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Pada *Caesar cipher*, tiap huruf disubtitusi dengan huruf ketiga berikutnya dari susunan alphabet yang sama. Dalam hal ini kuncinya adalah pergeseran huruf (yaitu 3) [2].

Selain kriptografi, teknik lain yang dapat digunakan untuk mengamankan data atau informasi adalah steganografi. Menurut Dony Arius [3], teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) didalam file-file lain yang mengandung teks, image, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari file semula.

Dari uraian diatas, maka dapat diambil rumusan masalah sebagai berikut : 1). Bagaimanakah implementasi penggunaan metode steganografi EOF dan *Caesar cipher* dalam proses penyisipan file terenkripsi dalam gambar ?, 2). Bagaimanakah pengaruh dari penggunaan steganografi EOF dan *Caesar cipher* terhadap keamanan file yang berisi informasi ?.

Agar pembahasan dalam penelitian ini lebih terarah, maka diberikan beberapa batasan masalah sebagai berikut: 1). Media penyisipan file yang digunakan adalah gambar dengan format bebas, 2). File yang disisipkan dalam bentuk text file yang berekstensi .doc/.docx dan .txt, 3). Metode steganografi yang digunakan untuk menyisipkan file ke dalam gambar adalah *End Of File* (EOF), 4). Menggunakan teknik subtitusi *Caesar cipher* untuk mengenkripsi isi file yang akan disisipkan. 5). Aplikasi untuk implementasi dibuat dengan menggunakan *compiler* Delphi 2010.

Beberapa tujuan yang ingin dicapai dalam penelitian ini diantaranya: 1). Mengetahui konsep metode *Caesar cipher* dan steganografi EOF untuk mengenkripsi isi file dan menyisipkan file tersebut dalam gambar. 2) Menerapkan algoritma metode steganografi EOF dan *Caesar cipher* dalam sebuah aplikasi berbasis desktop. 3). Dapat merancang sebuah aplikasi yang menerapkan kombinasi teknik kriptografi metode *Caesar cipher* dan steganografi EOF untuk menyisipkan file pesan dalam media gambar.

Sebelum melakukan proses penelitian dan uji coba penyisipan file yang telah terenkripsi ke dalam media gambar, beberapa tahapan telah dilakukan mulai dari studi literatur, kajian pustaka, mengumpulkan beberapa citra gambar dan file-file uji coba, memahami metode *Caesar cipher* dan steganografi *End Of File* (EOF) sampai dengan pada proses perancangan serta implementasi hasil perancangan dengan menggunakan kedua metode tersebut.

2. Tinjauan Pustaka/Penelitian Sebelumnya

Kajian penelitian sebelumnya ini dilakukan dengan tujuan untuk mengetahui dan mempelajari penelitian yang telah dilakukan oleh peneliti sebelumnya yang relevan dengan topik yang sedang diteliti saat ini sekaligus sebagai bahan rujukan untuk pembangunan sistem didalam penelitian ini.

Beberapa penelitan terdahulu tentang penerapan metode steganografi *End Of File* (EOF) dan *Caesar cipher* yang sudah pernah dilakukan peneliti sebelumnya diantaranya :

1) Sukrisno dan Ema Utami [4] dengan judul penelitian : "Implementasi Steganografi Teknik EOF Dengan Gabungan Enkripsi Rijndael, Shift cipher Dan Fugsi Hash MD5". Dalam penelitian ini menjelaskan tentang bagaimana menghasilkan sebuah program yang menerapkan algoritma steganografi EOF. Kesamaan dengan penelitian yang dilakukan sekarang adalah menghasilkan program yang menerapkan teknik steganografi EOF, sedangkan perbedaan dengan penelitian sekarang ini adalah pada penelitian sekarang, program yang dihasilkan tidak hanya menerapkan teknik steganografi EOF tetapi juga kriptografi metode *Caesar cipher*. Selain itu, media yang digunakan untuk steganografi pada penelitian sebelumnya adalah *file* teks sedangkan pada penelitian sekarang media steganografi yang digunakan adalah gambar dengan format bebas.
2) Krisnawati [5] dengan judul penelitian : "Metode *Least Significant Bit* (LSB) Dan *End Of File (EOF)* Untuk Menyisipkan Teks Ke Dalam Citra *Grayscale*". Dalam penelitian ini memaparkan proses implementasi metode LSB dan EOF untuk menyisipkan pesan teks ke dalam citra grayscale. Persamaan dengan penelitian yang dilakukan sekarang adalah penerapan metode EOF untuk menyisipkan *file* teks ke dalam media gambar. Perbedaannya terletak pada media yang digunakan. Jika pada penelitian sebelumnya pada citra *grayscale* maka pada penelitian sekarang menggunakan media gambar dengan format bebas. Selain itu aplikasi yang digunakan pada penelitian sebelumnya menggunakan Matlab 6.1, sedangkan pada penelitian sekarang menggunakan aplikasi yang dibuat sendiri.
3) Yogie Aditya, Andhika Pratama dan Alfian Nurlifa [6], dalam penelitian yang berjudul "Studi Pustaka Untuk Steganografi Dengan Beberapa Metode". Penelitian ini membahas tentang analisis terhadap keefektifan metode steganografi dalam proses penyembunyian pesan dengan memanfaatkan sebuah aplikasi. Persamaan dengan penelitian sekarang adalah membahas tentang metode steganografi EOF, sedangkan perbedaannya terletak pada fungsi penelitian. Jika pada penelitian sebelumnya merupakan studi pustaka metode-metode steganografi, maka pada penelitian sekarang lebih kepada implementasi metode steganografi EOF dalam sebuah aplikasi.
4) Ardiyanto [7] dalam tulisan yang berjudul "Implementasi Algoritma Kriptografi *Caesar cipher* Pada Aplikasi Sms Telepon Selular Berbasis J2ME". Dalam artikel ini membahas tentang penerapan *Caesar cipher* pada aplikasi SMS. Kesamaan dengan penelitian sekarang adalah penerapan teknik kriptografi metode *Caesar cipher* dalam sebuah aplikasi. Perbedaan terletak pada antar muka/aplikasi yang digunakan dan pesan yang akan dienkripsi. Pada penelitian sebelumnya menggunakan Netbeans sedangkan penelitian sekarang menggunakan Delphi 2010. Selain itu pada penelitian sebelumnya yang dienkripsi adalah *file* pesan/sms, sedangkan pada penelitian sekarang yang dienkripsi adalah *file* bertipe teks (*text file*).
5) Wasino, Tri Puji Rahayu dan Setiawan [8] dalam penelitian yang berjudul "Implementasi Steganografi Teknik *End Of File* Dengan Enkripsi *Rijndael*". Dalam penelitian ini akan dibangun sebuah aplikasi yang menerapkan steganografi teknik EOF dan Rijndael. Persamaan dengan penelitian sekarang adalah dalam penerapan metode EOF untuk menyisipkan pesan, sedangkan perbedaannya terletak pada metode enkripsi yang digunakan. Pada penelitian sebelumnya menggunakan enkripsi Rijndael sedangkan pada penelitian sekarang menggunakan metode *Caesar cipher.*
6) Sandro Sembiring [9] dengan penelitian yang berjudul "Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End Of File". Penelitian ini membahas tentang bagaimana cara merancang suatu aplikasi steganografi *end of file* dan bagaimana proses penyisipanya dalam gambar. Kesamaan dengan penelitian yang dilakukan sekarang adalah merancang aplikasi

steganografi untuk menyisipkan pesan dalam media gambar. Perbedaannya adalah pada penelitian sebelumnya hanya menggunakan teknik steganografi EOF, sedangkan pada penelitian sekarang teknik steganografi EOF dikombinasikan dengan teknik kriptografi *Caesar cipher*.

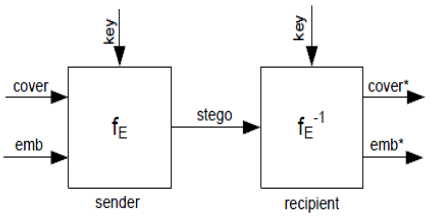
7) Mukharrom Edisuryana, R. Rizal Isnanto, dan Maman Somantri [10] dengan judul penelitian "Aplikasi Steganografi Pada Citra Berformat Bitmap Dengan Menggunakan Metode *End Of File*". Dalam penelitian ini diimplementasikan teknik kriptografi dan steganografi pada citra berformat bitmap dengan menggunakan metode *End Of File* (EOF). Penelitian sebelumnya dengan yang sekarang dilakukan memiliki persamaan yakni menerapkan kombinasi steganografi EOF dan kriptografi *Caesar cipher*, sedangkan perbedaannya adalah pada implementasinya. Penelitian sebelumnya menggunakan MATLAB R2008a, sedangkan pada penelitian sekarang implementasinya menggunakan aplikasi yang dibuat sendiri berdasarkan pada algoritma kedua metode.
8) Yayuk Anggraini dan Dolly Virgian Shaka Yudha Sakti [11] dengan judul penelitian "Penerapan Steganografi Metode *End Of File* (EOF) Dan Enkripsi Metode *Data Encryption Standard* (DES) Pada Aplikasi Pengamanan Data Gambar Berbasis Java Programming". Pada penelitian sebelumnya dan sekarang, memiliki kesamaan penerapan teknik steganografi EOF pada media citra/gambar. Perbedaannya terletak pada perlakuan *file* pesan yang akan disisipkan. Pada penelitian sebelumnya, *file* pesan tidak dienkripsi sedangkan pada penelitian sekarang, pesan yang akan disisipkan terlebih dahulu diacak/dienkripsi dengan metode *Caesar cipher*.

Berikut ini adalah beberapa pustaka yang digunakan untuk menunjang pelaksanaan kegiatan penelitian ini.

2.1 Pengertian Steganografi

Menurut Rinaldi Munir [12], steganografi (*steganography*) adalah suatu teknik yang digunakan untuk menyembunyikan data rahasia di dalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang lain. 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. Menurut Putri Alatas [13], penilaian sebuah algoritma steganografi yang baik dapat dinilai dari beberapa faktor yaitu : 1). *Impercepbility* : Keberadaan pesan rahasia dalam media penampung tidak dapat dideteksi oleh inderawi 2). *Fidelity* : Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan itu tidak dapat dipersepsi oleh inderawi. 3). *Recovery* : Pesan yang disembunyikan harus dapat diungkapkan kembali. Menurut pendapat Dony Arius [3] bahwa tujuan dari steganografi ini adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi.

Beberapa hal yang diperlukan untuk menyembunyikan pesan yaitu Rinaldi Munir [14]: 1). Algoritma Penyisipan : Algoritma ini digunakan untuk menyisipkan suatu pesan yang disembunyikan ke dalam suatu data yang akan dikirim. 2) Fungsi Detektor : 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.. 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* : merupakan pesan rahasia yang akan disisipkan kedalam *carrier document*. Adapun gambaran alur sistem steganogafi diperlihatkan pada Gambar 1.



Gambar 1. Sistem Steganografi [15]

Dimana :
f_E = fungsi steganografi *"embedding"*
f_E^{-1} = fungsi steganografi "*extracting*"
cover = *coverdata* pada *emb* akan di sembunyikan
emb= pesan yang akan disisipakan
key = parameter f_E
stego = *coverdata* dengan pesan yang telah disisipkan

2.2 Metode End Of File (EOF)

Metode *End Of File* (EOF) merupakan metode pengembangan LSB (*Least Significant Bit*). Menurut Yogie Aditya, Andhika Pratama dan Alfian Nurlifa [6], metode EOF adalah metode yang digunakan untuk meyembunyikan pesan rahasia dengan cara manambahkan bit-bit pesan yang akan disembunyikan ke akhir file citra penampung, sedangkan menurut Henny Wandani, Muhammad Andri Budiman dan Amer Sharif [16], metode *End Of File* (EOF) merupakan salah satu metode yang digunakan dalam steganografi dimana teknik ini digunakan dengan cara menyisipkan data pada akhir *file*, sehingga tidak akan mengganggu kualitas data awal yang akan disisipkan pesan. Dalam metode EOF ini pesan disisipkan diakhir berkas. Pesan yang disisipkan dengan metode ini jumlahnya tidak terbatas. Akan tetapi efek sampingnya adalah ukuran berkas menjadi lebih besar dari ukuran semula. Ukuran berkas yang terlalu besar dari yang seharusnya, tentu akan menimbulkan kecurigaan bagi yang mengetahuinya.

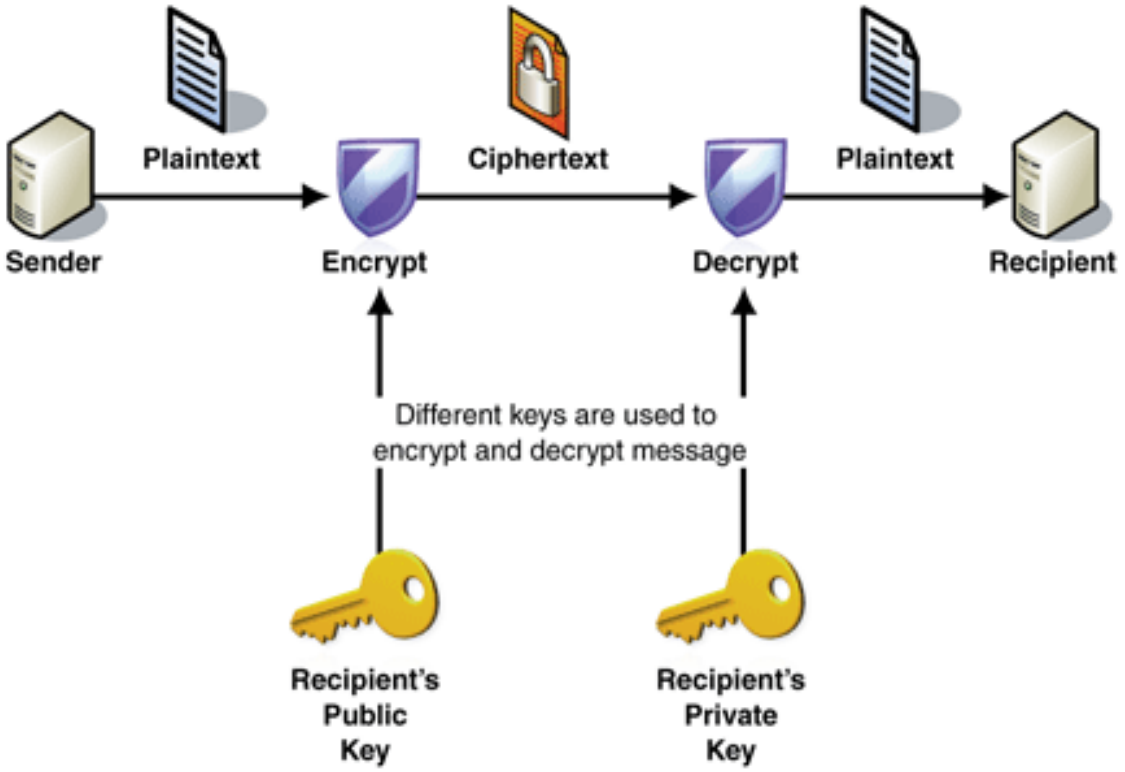
Adapun langkah-langkah *encoding* dengan metode *End of File* adalah sebagai berikut [9]: 1). Proses *encoding* dimulai dengan pesan yang akan disisipkan. Pesan diubah kedalam bentuk biner dengan representasi 1 atau 0. 2). Kemudian disisipkan angka 1 di depan rangkaian biner tersebut. Langkah selanjutnya rangkaian biner tersebut dikonversikan menjadi bilangan desimal dan menghasilkan sebuah bilangan yang dinamakan dengan *m*. 3). Menghitung jumlah warna yang terdapat pada berkas RGB yang menjadi objek steganografi.dan akan menghasilkan sebuah bilangan.Bilangan tesebut dinamakan dengan *n*, maka apabila $m>n!-1$ maka pesan yang akan disisipkan berukuran terlalu besar sehingga proses penyisipan tidak dapat dilakukan. 4). Warna dalam palet warna diurutkan sesuai dengan urutan yang "natural". Setiap warna dengan *format* RGB dikonversikan kedalam bilangan integer dengan aturan (Merah * 65536 + Hijau * 256 + Biru). Kemudian diurutkan berdasarkan besar bilangan integer yang mewakili warna tersebut. 5). Setelah itu lakukan proses iterasi terhadap variabel *i* dengan nilai *i* adalah dari 1 sampai *n*. Setiap warna dengan urutan $n - i$ dipindahkan ke posisi baru yaitu $m \bmod i$, kemudian *m* dibagi dengan *i*. 6). Kemudian palet warna yang baru hasil iterasi pada langkah ke – 4 dimasukkan ke dalam palet warna berkas RGB. Apabila ada tempat yang diisi oleh dua buah warna, maka warna sebelumnya yang menempati tempat tesebut akan digeser satu tempat ke samping. 7). Apabila ternyata besar dari palet warna yang baru lebih kecil dari 256 maka palet warna akan diisi dengan warna terakhir dari palet warna sebelumnya. 8). Kemudian berkas RGB akan dikompresi ulang dengan palet warna yang baru, untuk menghasilkan berkas yang baru dengan ukuran dan gambar yang sama, namun telah disisipi pesan.

Langkah- langkah proses *decoding* atau mengekstrak pesan dari citra RGB yang telah disisipi pesan dengan metode *End Of File* adalah sebagai berikut [9]: 1). Masukkan nomor sesuai dengan posisi setiap warna pada palet warna citra RGB yang telah disisipkan pesan. 2). Warna diurutkan berdasarkan konversi RGB ke nilai integer dengan rumus: (Merah * 65536 + Hijau * 256 + Biru). 3). *m* diberi nilai 0. 4). Iterasi variabel *i* dari *i+1* sampai *n-1*. m=m*(n-1) + posisi warna ke *i* iterasi variabel *j* dari *i* +1 sampai n-1 jika posisi warna ke $j >$ nilai posisi warna ke-*i*, maka posisi warna ke *i* dikurangkan 1.

2.3 Pengertian Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu "*cryptos*" yang berarti rahasia dan "*graphein*" yang berarti tulisan. Jadi, kriptografi adalah tulisan rahasia. Secara umum, kriptografi diartikan sebagai ilmu dan seni

untuk menjaga kerahasiaan berita. Menurut Kurniawan [17] dalam bukunya yang berjudul "*Kriptografi Keamanan Internet dan Jaringan Komunikasi"*, menjelaskan bahwa kriptografi merupakan seni dan ilmu untuk menjaga keamanan pesan. Menurut Sentot Kromodimoeljo [18] kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Enkripsi adalah proses merubah *plainteks* (pesan asli) menjadi suatu pesan tersandi (*cipherteks*). Gambar 2 memperlihatkan skema aliran proses teknik kriptografi :



Gambar 2. Skema Aliran Proses Kriptografi

2.4 Pengertian Enkripsi

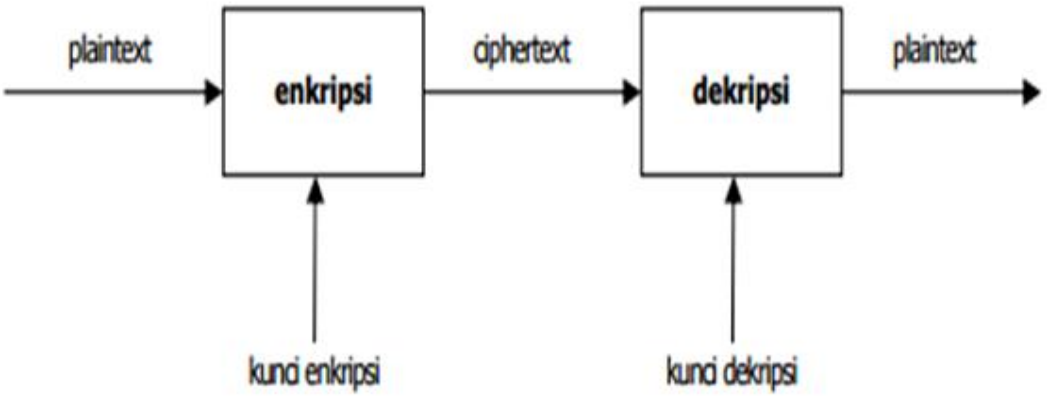
Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Enkripsi dapat diartikan sebagai kode atau *cipher*. Sebuah *cipher* menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti (*unitelligible*). Karena teknik *cipher* merupakan suatu sistem yang telah siap untuk di automasi, maka teknik ini digunakan dalam sistem keamanan komputer dan *network*. Menurut Munir [14], proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering* (standar nama menurut ISO 7498-2), sedangkan proses mengembalikan *ciphertext* menjadi *plaintext*-nya disebut dekripsi (*decryption*) atau *deciphering* (standard nama menurut ISO 7498-2). Gambar 3 menunjukkan diagram proses enkripsi dan dekripsi.

2.5 Enkripsi Caesar Cipher

Dalam buku Practical Workbook: Information Theory, 4th edition, Department of Computer & Information System Engineering NED University of Engineering & Technology, Karachi, Pakistan [19],

dijelaskan bahwa metode *Caesar cipher* yang digunakan menggunakan prinsip modulo 26. Secara matematis dapat dituliskan sebagai berikut :

cipher = rem ((x+kunci-97,26)+97) (1)



Gambar 3. Diagram proses enkripsi dan dekripsi

Dalam MATLAB, fungsi modulo dapat dilakukan dengan menuliskan fungsi mod‟ atau rem‟. *x* merupakan pesan asli yang akan disandikan dan kunci yang digunakan pada saat enkripsi sama dengan kunci dekripsi. *Caesar cipher* terbatas pada penyandian huruf alfabet dari a‟ hingga z‟ dimana dalam ASCII berada pada posisi 97 sampai 122. Mengurangkan 97 diterjemahkan ke dalam kisaran 0 sampai 25 (0 sebagai a‟ hingga 25 sebagai z‟). Kunci digunakan sebagai pergeseran dan mengambil sisanya pada pembagian dengan bilangan 26 melalui fungsi *rem‟* (modulo 26). Menambah 97 diterjemahkan kembali pada kisaran 97 sampai 122.

Teknik enkripsi substitusi yang pertama kali dikenal dan paling sederhana ditemukan oleh Julius Caesar. Metode yang digunakan dalam *Caesar cipher* ini adalah dengan mempertukarkan setiap huruf dari *plaintext* dengan huruf lain dengan interval 3 huruf atau beberapa interval huruf dari huruf *plaintext*. Sebagai contoh dapat dilihat pada Gambar 4:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	O	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Gambar 4. Contoh subtitusi Caesar cipher dengan interval 3 huruf

Secara umum, *Caesar cipher* adalah *cipher* pergeseran karena alphabet *ciphertext* diambil dari alphabet *plaintext* dengan menggeser masing-masing huruf dengan jumlah pergeseran tertentu.

3. Metodologi

Berikut adalah metodologi yang digunakan dalam menunjang kegiatan penelitian ini .

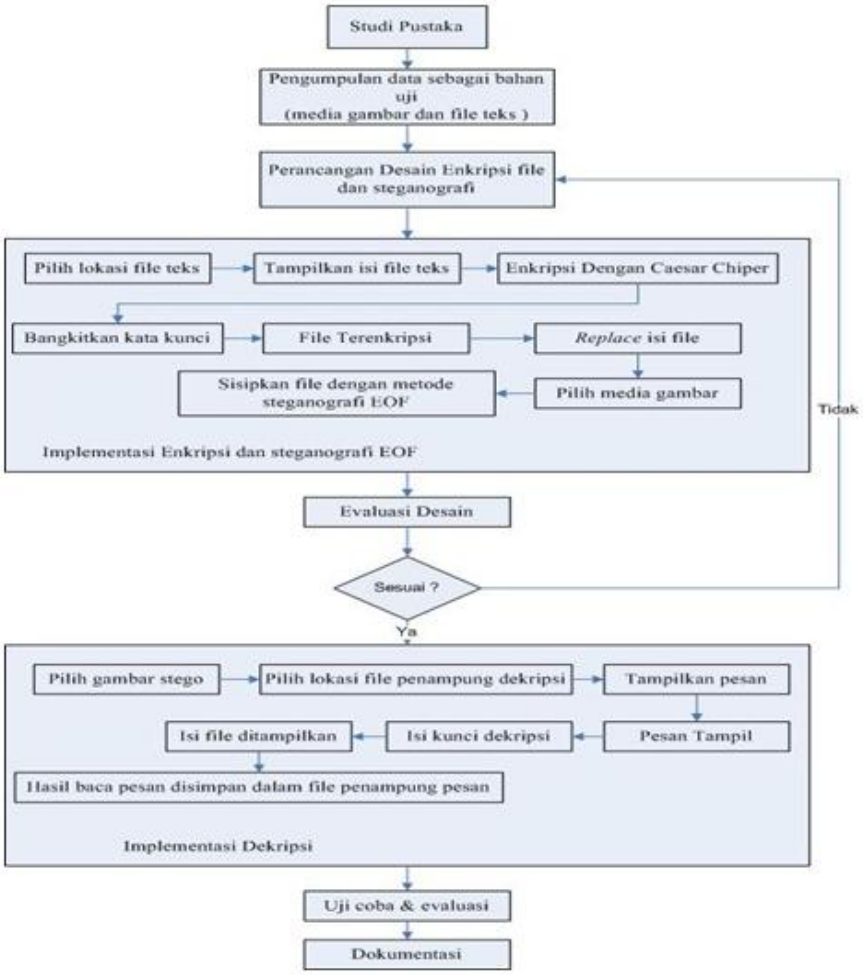
3.1 Metode Penelitian

Dalam penelitian ini metode yang digunakan adalah deskriptif analitik. Langkah yang dilakukan adalah mengumpulkan bahan penelitian, studi literatur, mempelajari algoritma steganografi EOF dan kriptografi

Caesar cipher, melakukan perancangan dan implementasi/pengujian sistem berdasarkan algoritma steganografi EOF serta *Caesar cipher* yang digunakan. Secara umum, langkah-langkah implementasi yang dilakukan dalam penelitian ini adalah sebagai berikut: 1. Melakukan pemilihan *file* tipe teks. 2. Mengacak isi *file* tersebut dengan metode *Caesar cipher*. 3. Memilih media gambar. 4. Melakukan penyisipan *file* terenkripsi dengan steganografi EOF. 5. Mendekripsi *file* teks yang berisi pesan/informasi dan menampilkan informasi yang berada dalam gambar *stego*.

3.2 Alur Penelitian

Alur penelitian adalah tahapan proses yang dilakukan oleh peneliti dalam mengkaji dan membangun aplikasi penyisipan *file* dalam media gambar yang menerapkan kombinasi metode steganografi *End Of File* dan teknik kriptografi *Caesar cipher*. Tahapan dalam penelitian ini diperlihatkan seperti pada Gambar 5.



Gambar 5. Alur Penelitian

Gambar 5 menjelaskan tahapan proses yang diterapkan dalam penelitian ini. Tahapan proses dimulai dari studi pustaka dimana pada bagian ini dilakukan kajian terhadap pustaka yang digunakan dalam penulisan naskah dan mempelajari pustaka tentang metode steganografi EOF dan *Caesar cipher*. Tahap berikutnya adalah mengumpulkan media gambar sebagai media yang akan disisipi *file* pesan dan *file* teks yang akan

disisipkan dalam gambar. Setelah tahap pertama dan kedua selesai, selanjutnya dilakukan tahap desain dan perancangan enkripsi *Caesar cipher* dan steganogafi EOF. Evaluasi dilakukan dari perancangan tersebut dan jika sesuai maka dilanjutkan dengan perancangan desain untuk melakukan dekripsi *Caesar cipher*.

3.3 Analisis Proses Embedding Metode End Of File (EOF)

Menurut Henny Wandani [16], tahapan proses *embedding* atau penyisipan pesan menggunakan metode *End of File* adalah sebagai berikut : 1). Inputkan *ciphertext* yang akan disisipkan. 2). Inputkan citra yang akan menjadi media penyisipan *ciphertext* (*cover image*). 3). Baca nilai setiap *pixel* citra. 4). Baca nilai setiap *pixel* citra. 5). Petakan menjadi citra baru.

Sebagi contoh diberikan kasus penyisipan *cipher text* menggunakan metode *End Of File* dengan sebuah citra RGB 8x8 yang memiliki nilai setiap *pixel* seperti diperlihatkan pada Gambar 6.

104	38	55	104	96	96	77	92
80	93	60	60	60	51	56	94
91	79	16	62	90	69	73	87
97	98	70	52	60	62	52	99
85	83	37	18	82	88	51	56
87	84	56	65	68	39	106	101
69	37	44	74	80	68	99	99
66	62	60	32	105	88	71	77

Gambar 6. Contoh Matriks Pixel Citra RGB

Dari citra tersebut akan disisipkan *ciphertext* dengan nilai : “101 120 97 109 112 108 101”. Nilai tersebut akan ditambahkan sebagai nilai akhir pada *pixel* citra RGB. Pada akhir *ciphertext* diberi karakter penanda “y” yang memiliki nilai desimal “255”. Maka akan didapatkan nilai matriks *pixel* seperti pada Gambar 7 dibawah ini :

104	38	55	104	96	96	77	92
80	93	60	60	60	51	56	94
91	79	16	62	90	69	73	87
97	98	70	52	60	62	52	99
85	83	37	18	82	88	51	56
87	84	56	65	68	39	106	101
69	37	44	74	80	68	99	99
66	62	60	32	105	88	71	77
101	**120**	**97**	**109**	**112**	**108**	**101**	**255**

Gambar 7. Matriks Pixel Citra RGB yang disisipi ciphertext

Dari proses penyisipan nilai *ciphertext* tersebut akan dibaca nilai *pixel image stego* seperti yang diperlihatkan pada Gambar 8 dibawah ini :

101	**120**	**97**	**109**	**112**	**108**	**101**	**255**

Gambar 8. Matriks Pixel Image Stego

3.4 Analisis Caesar Cipher

Di dalam *cipher* subtitusi, setiap unit *plainteks* diganti dengan satu unit *cipherteks*. Satu “unit” di sini bisa berarti satu huruf, pasangan huruf, atau kelompok lebih dari dua huruf. Adapun langkah-langkah yang

dilakukan untuk membentuk *plainteks* ke *cipherteks* begitu juga sebaliknya dengan menggunakan *Caesar cipher* adalah: (1). Menentukan berapa pergeseran karakter (kunci) yang digunakan dalam membentuk *cipherteks* ke *plainteks*. (2). Menukarkan karakter pada *plainteks* menjadi *cipherteks* dengan berdasarkan pada pergeseran karakter (kunci) yang telah ditentukan sebelumnya.

Contoh penyandian sebuah pesan singkat yang diuji coba dalam penelitian ini dengan pergeseran sebanyak 1 huruf adalah sebagai berikut. Diketahui *Plainteks*: DIBERITAHUKAN BAHWA AKAN ADA PENERIMAAN KARYAWAN BARU. Maka *cipherteks* yang dihasilkan adalah sebagai berikut: Ejcfsjubivlbo cbixb blbo beb qfofsjnbbo lbszbxbo cbsv.

Dengan mengkodekan setiap huruf alfabet dengan integer : 'A'= 0 , 'B'= 1,…, 'Z'= 25, maka secara matematis pergeseran 1 huruf *alfabetik ekivalen* dengan melakukan operasi *modulo* terhadap *plainteks* P menjadi *cipherteks C* dengan persamaan berikut :

$$C = E(P) = (P + 1) \bmod 26 \quad (2)$$

Karena ada 26 huruf didalam alphabet, maka penerima pesan mengembalikan lagi *cipherteks* dengan operasi kebalikan, secara matematis dapat dinyatakan dengan persamaan:

$$P = D(C) = (C - 1) \bmod 26 \quad (3)$$

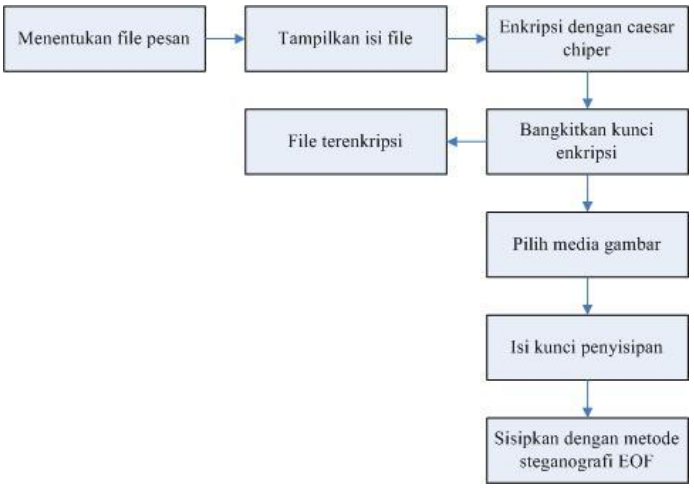
Dimana: ***P*** = *Plainteks*; ***C*** = *Ciperteks*; ***D*** = Dekripsi; ***E*** = Enkripsi; ***K*** = Kunci

4. Hasil dan Pembahasan

Pada bagian hasil dan pembahasan ini, peneliti akan menguraikan tahap-tahap analisis, perancangan dan uji coba berdasarkan pada tujuan dan alur penelitian yang telah digambarkan:

4.1 Analisis dan Rancangan Enkripsi Caesar Cipher dan Steganografi EOF

Pada bagian ini akan digambarkan tahapan dari proses enkripsi file teks dengan *Caesar cipher* dan bagaimana *file* tersebut disisipkan dalam gambar. Alur pada bagian ini diperllihatkan pada Gambar 9 dibawah ini:

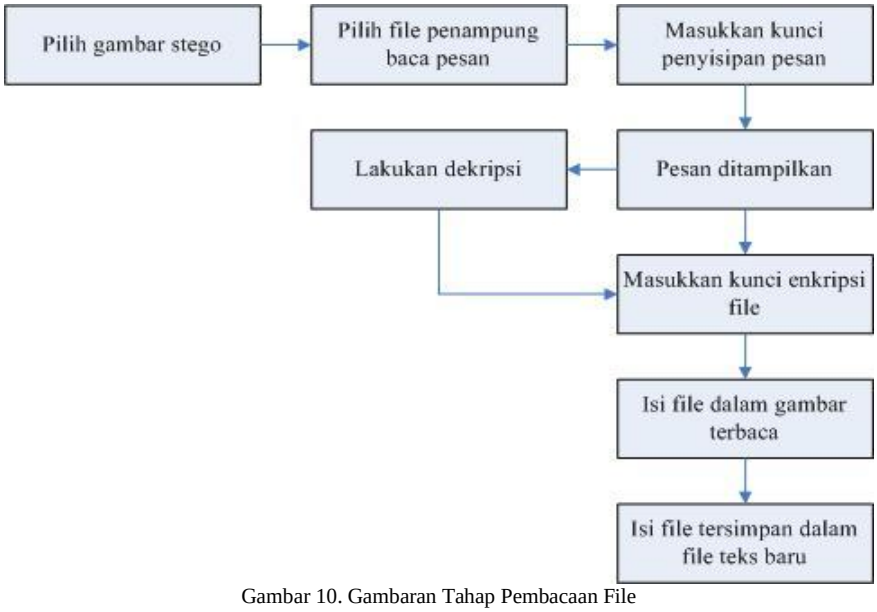


Gambar 9. Gambaran umum proses enkripsi dan steganografi EOF

Dalam Gambar 9 terlihat bahwa penyisipan dengan metode steganografi EOF dalam penelitian ini dilakukan terhadap *file* teks. Sebelum *file* teks tersebut disisipkan, dilakukan terlebih dahulu proses enkripsi dengan *Caesar cipher*. Pada saat proses enkripsi dilakukan, maka dibangkitkan sebuah *key* untuk men-subtitusi-kan isi *file* dengan algoritma *Caesar cipher*. Setelah *file* terenkripsi, maka selanjunya dipilih media gambar dengan sembarang ekstensi *file*. Kemudian apabila gambar sudah terpilih, *file* teks yang isinya telah terenkripsi dapat disisipkan ke dalam gambar terpilih dengan terlebih dahulu memasukkan kunci penyisipan gambar.

4.2 Analisis dan Rancangan Pembacaan File Pesan

Pada bagian ini, akan digambarkan alur/tahapan dimana isi *file* yang disisipkan dalam gambar dimunculkan kembali. Isi *file* yang telah terbaca, hasilnya akan disimpan didalam *file* baru dengan ekstensi teks (.txt dan .doc/.docx). Gambar 10 memperlihatkan alur dari proses pembacaan isi *file* tersebut:



Gambar 10. Gambaran Tahap Pembacaan File

Gambar 10 menunjukkan bagaimana jalannya proses pembacaan *file* yang berada dalam gambar. Proses dimulai dari memilih gambar yang sudah berisi *file* pesan. Kemudian menentukan dimana hasil baca pesan tersebut akan disimpan. Selanjutnya, mengisi kunci yang dibentuk pada saat *file* disisipkan dalam gambar. Apabila kunci benar, maka isi *file* akan ditampilkan. Isi *file* yang ditampilkan masih terenkripsi. Untuk menampilkan isi *file* pesan yang sebenarnya maka perlu diisikan kunci enkripsi (kunci pergeseran) yang dibentuk saat *file* dienkripsi. Apabila kunci benar, maka isi *file* yang sebenarnya tampil dan hasilnya akan disimpan pada *file* teks yang baru.

4.3 Analisis Kebutuhan Jalannya Perangkat Lunak Aplikasi

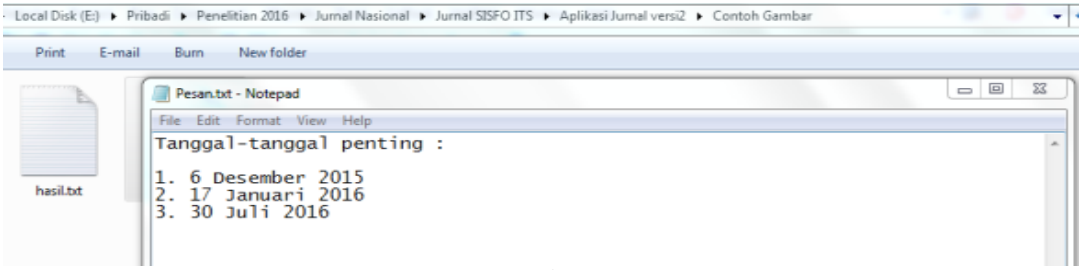
Dalam penelitian ini, aplikasi yang dibangun untuk mengimplementasikan kombinasi steganografi *EOF* dengan Enkripsi *Caesar cipher* memiliki kebutuhan sebagai berikut: 1). Aplikasi dapat membaca *covertext* dan *file* yang akan disembunyikan. 2). Dapat melakukan enkripsi sesuai dengan algoritma yang digunakan. 3). Dapat memasukan kata sandi agar hanya orang yang berkepentingan saja yang dapat mengungkap pesan rahasia. 4). Dapat menyimpan hasil enkripsi atau *stegotext* kedalam *file*. 5). Dapat melakukan dekripsi pada *stegotext* untuk mengeluarkan pesan yang tersembunyi. 6). Dapat menyimpan hasil dari dekripsi sebagai suatu *file* baru.7). Tampilan menggunakan GUI (*Graphical User Interface*).

4.4 Implementasi

Pada bagian ini, rancangan tahapan baik untuk memulai proses enkripsi *Caesar cipher* terhadap *file* teks, penyisipan dalam gambar sampai dengan proses pembacaan isi *file* diterapkan dalam rancangan antar muka. Gambar 11 memperlihatkan tampilan awal menu aplikasi.

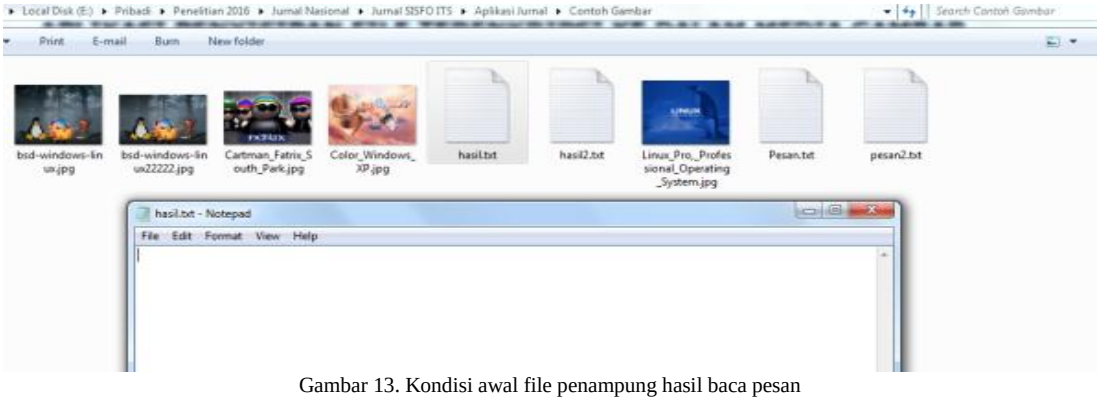
Gambar 11. Tampilan awal menu aplikasi

Kondisi awal isi *file* teks yang akan disisipkan diperlihatkan pada Gambar 12:



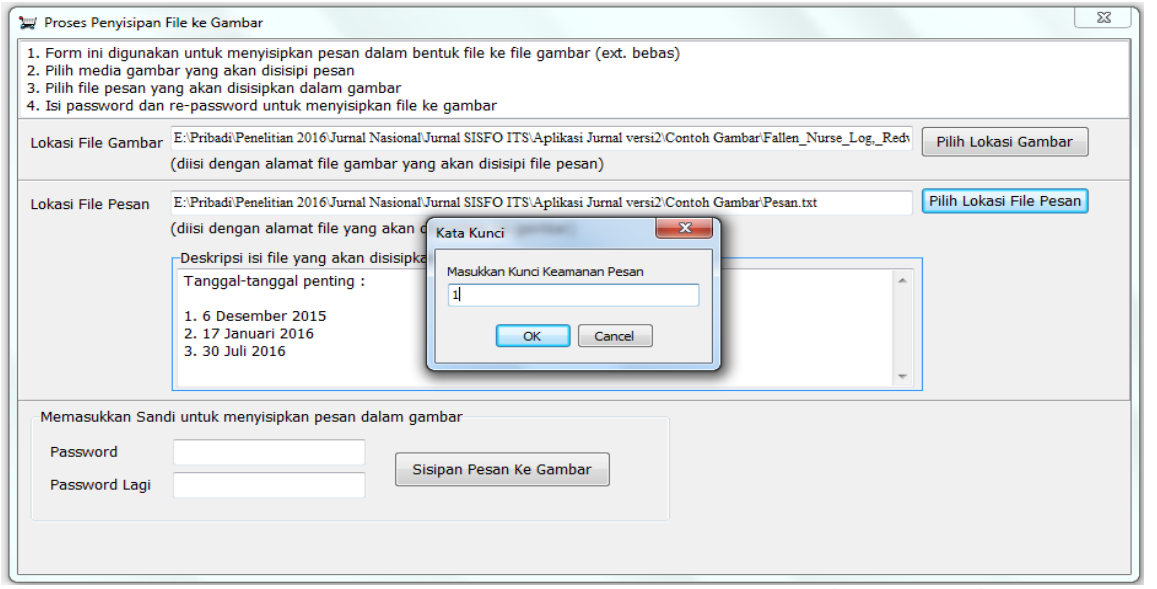
Gambar 12. Kondisi awal file yang akan disisipkan

Gambar 12 memperlihatkan kondisi awal dari *file* yang akan disisipkan ke dalam gambar, sedangkan kondisi awal file baru sebagai penampung hasil baca pesan diperlihatkan pada Gambar 13:



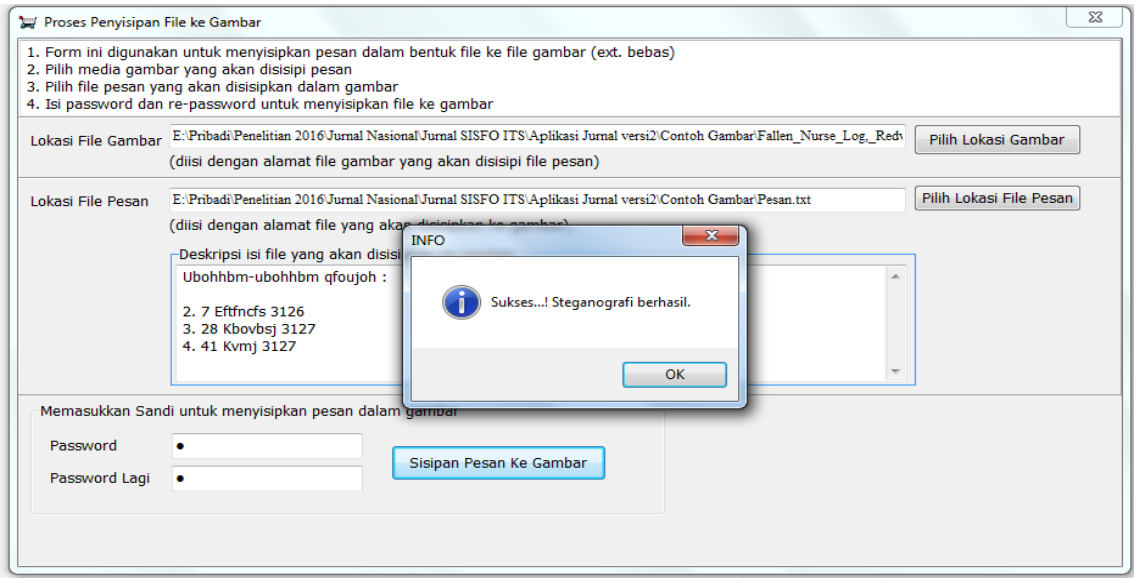
Gambar 13. Kondisi awal file penampung hasil baca pesan

Proses berikutnya adalah enkripsi dengan *Caesar cipher* terhadap *file* pesan dan menyisipkannya kedalam gambar. Tampilan proses diperlihatkan pada Gambar 14.



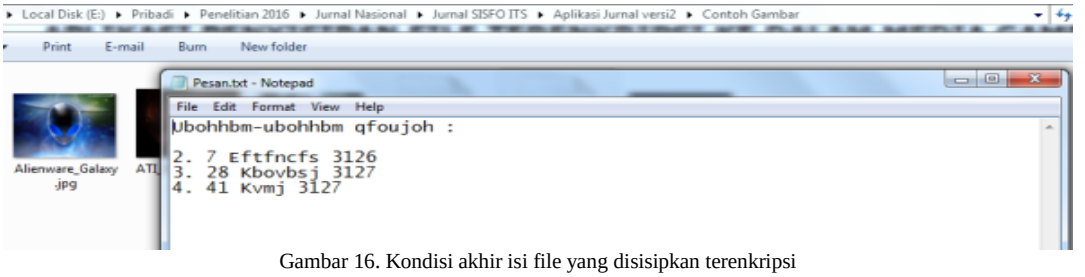
Gambar 14. Form untuk proses enkripsi Caesar cipher dengan pergeseran 1 huruf dan penyisipan file

Dari Gambar 14 dapat dijelaskan tahapan proses enkripsi dan penyisipan *file* dalam gambar. Langkah awal adalah menentukan lokasi gambar. Kemudian menentukan *file* yang akan disisipkan. Setelah *file* terpilih, maka secara otomatis isi *file* awal akan dimunculkan. Pada saat isi *file* muncul, maka ditampilkan juga kotak *key* subtitusi *Caesar cipher* yang harus diisi angka antara 1-26. Contoh diatas adalah dengan menggunakan kunci pergeseran 1 huruf. Selanjutnya memasukkan kunci penyisipan *file* dalam gambar. Hasil dari proses ini diperlihatkan pada Gambar 15.



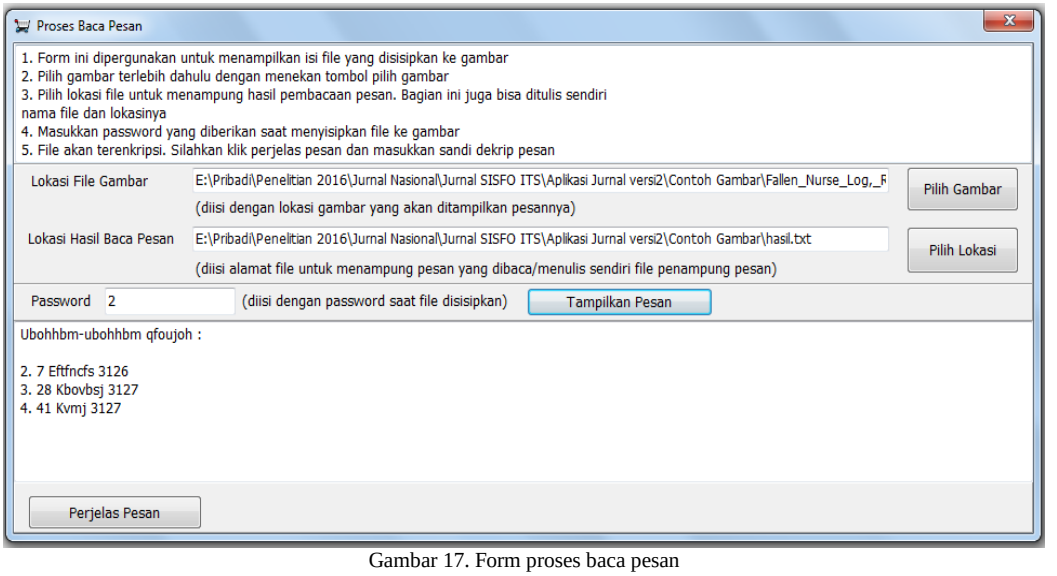
Gambar 15. Hasil enkripsi dengan kunci pergeseran 1 huruf dan penyisipan file

Pada waktu proses enkripsi dan penyisipan *file* berhasil, maka terjadi perubahan pada isi *file* awal seperti pada Gambar 16.

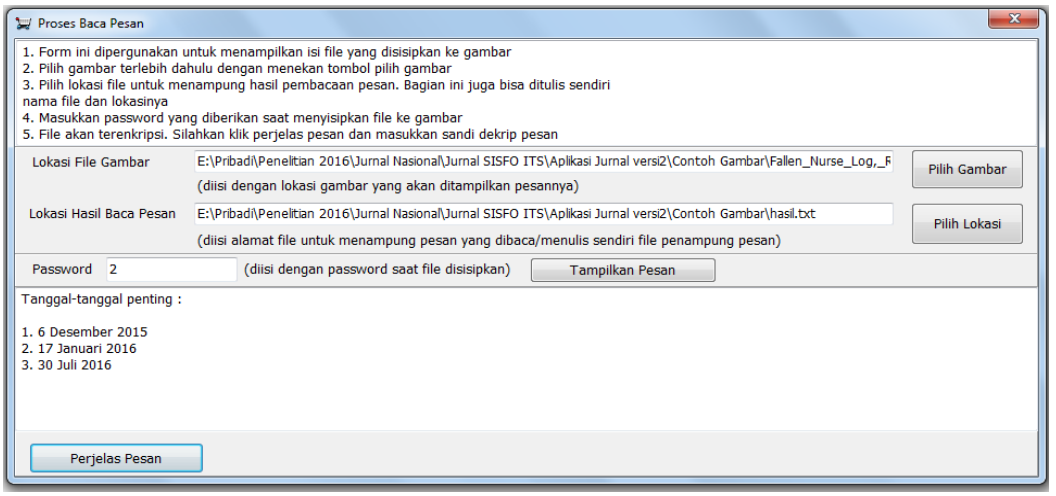


Gambar 16. Kondisi akhir isi file yang disisipkan terenkripsi

Proses selanjutnya setelah penyisipan *file* adalah membaca isi *file* tersebut yang sudah tersimpan dalam gambar. Gambar 17 dibawah ini memperlihatkan antar muka proses baca pesan yang disisipkan.



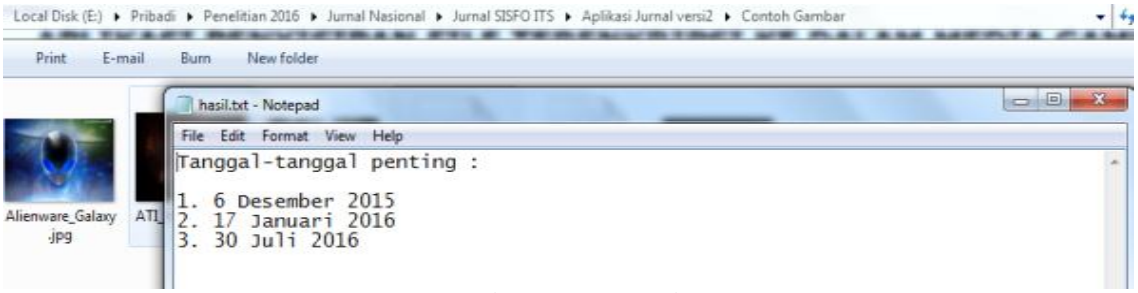
Gambar 17. Form proses baca pesan



Gambar 18. Hasil proses baca pesan gambar

Dari Gambar 17 dapat dijelaskan langkah pembacaan pesan sebagai berikut: langkah pertama memilih gambar yang telah berisi pesan. Kemudian menentukan lokasi *file* yang akan digunakan untuk menyimpan hasil baca pesan. Setelah itu, memasukkan kunci pada saat penyisipan gambar. Apabila kunci sesuai, maka pesan akan ditampilkan. Pesan yang muncul adalah pesan yang terenkripsi. Agar pesan yang sesungguhnya bisa tampil, maka harus memasukkan angka kunci saat enkripsi dengan klik tombol **perjelas pesan.** Tampilan akhir dari proses ini diperlihatkan pada Gambar 18.

Pada saat pesan asli dapat ditampilkan, maka hasil baca pesan ini otomatis akan tersimpan dalam *file* teks yang baru. Hasilnya diperlihatkan pada Gambar 19.



Gambar 19. Isi file tersimpan dalam file teks baru

5. Kesimpulan

Berdasarkan pada rumusan masalah dan hasil penelitian, maka dapat ditarik kesimpulan sebagai berikut:

5.1 Simpulan

1) Teknik steganografi metode *End Of File* (EOF) dan algoritma subtitusi *Caesar cipher* dapat diimplementasikan dalam sebuah aplikasi keamanan data berbasis *desktop*. Aplikasi ini dibangun berdasarkan pada bagaimana subtitusi *Caesar cipher* untuk melakukan enkripsi isi *file* ini berjalan dan metode steganografi EOF untuk menyisipkan *file* pada akhir gambar.
2) *Caesar cipher* yang digunakan untuk melakukan enkripsi file memiliki pengaruh yang signifikan terhadap kemanan atau kerahasiaan file yang akan disisipkan karena isi file tidak mudah terbaca oleh pihak yang tidak berkepentingan karena di enkripsi dengan kunci pergeseran/subtitusi tertentu. Demikian juga dengan metode steganografi EOF juga memiliki pengaruh terhadap kualitas gambar yang disisipi file pesan karena pada saat penyisipan, file pesan ditempatkan pada akhir baris *pixel* gambar. Teknik steganografi EOF tidak menyebabkan kualitas citra berubah, sehingga tidak menimbulkan kecurigaan pihak-pihak lain.

5.2 Saran

Agar penelitian dengan topik yang relevan ini dapat berkembang, maka dapat diberikan saran-saran sebagai berikut:

1) Media yang digunakan untuk menampung pesan yang disisipkan masih berupa media gambar, sehingga untuk penelitian selanjutnya dapat digunakan media selain gambar agar dapat diketahui lebih dalam efek dari steganografi EOF pada media lain.
2) File yang disisipkan ke media gambar dalam penelitian ini hanya menggunakan *file* teks yang berekstensi .txt dan .doc/.docx. Isi *file* teks pun tidak menyertakan simbol-simbol, sehingga tidak dapat terenkripsi dengan baik jika terdapat simbol dalam *file* pesan. Oleh karena itu untuk penelitian

selanjutnya agar dapat menggunakan *file* selain bertipe .txt dan .doc/.docx serta mengandung simbol-simbol dalam *file* tersebut untuk mengetahui hasil enkripsi simbol dengan *Caesar cipher*.

6. Daftar Rujukan

[1] Nurhayati, O.D., 2010. *Keamanan Multimedia*, Program Studi S1 Sistem Komputer: Universitas Diponegoro.

[2] Seftyanto, D.; Apriani, M.; Haryanto, T., 2012. Peran Algoritma *Caesar Cipher* Dalam Membangun Karakter Akan Kesadaran Keamanan Informasi. *Seminar Nasional Matematika dan Pendidikan Matematika*. Jurusan Pendidikan Matematika FMIPA UNY 10 November 2012. Yogyakarta

[3] Arius, D., 2006. *Computer Security*. Yogyakarta: ANDI.

[4] Sukrisno; Utami, E., 2007. Implementasi Steganografi Teknik EOF dengan Gabungan Enkripsi Rijndael, Shift Cipher dan Fungsi Hash Md5. *Seminar Nasional Teknologi 2007 (SNT 2007)*. Yogyakarta, 24 November 2007. ISSN : 1978 – 9777.

[5] Krisnawati, 2008. Metode Least Significant Bit (LSB) dan End Of File (EOF) Untuk Menyisipkan Teks ke Dalam Citra Grayscale. *Seminar Nasional Informatika 2008 (semnasIF 2008)*. UPN "Veteran" Yogyakarta, 24 Mei 2008. ISSN: 1979-2328.

[6] Aditya, Y.; Pratama, A.; Nurlifa, A., 2010. Studi Pustaka Untuk Steganografi Dengan Beberapa Metode. *Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*. Yogyakarta, 19 Juni 2010. ISSN: 1907-5022.

[7] Ardiyanto, 2011. *Implementasi Algoritma Kriptografi Caesar Cipher Pada Aplikasi SMS Telepon Selular Berbasis J2ME*. Naskah Publikasi : STMIK AMIKOM Yogyakarta.

[8] Wasino; Rahayu, T.P; Setiawan., 2012. Implementasi Steganografi Teknik End Of File Dengan Enkripsi Rijndael. *Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012)*. Yogyakarta, 10 Maret 2012. ISSN: 2089-9815.

[9] Sembiring, S., 2013. Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End Of File. *Pelita Informatika Budi Darma*, IV (2). ISSN : 2301-9425.

[10] Edisuryana, M; Isnanto, R.R; Somantri, M., 2013. Aplikasi steganografi pada citra berformat bitmap dengan menggunakan metode end of file. *TRANSIENT*, 2 (3) , 735.

[11] Anggraini, Y; Sakti, D.V.S.Y., 2014. Penerapan steganografi metode *End Of File* (EOF) dan Enkripsi metode *Data Encryption Standard* (DES) pada aplikasi pengamanan data gambar berbasis java programming. *Konferensi Nasional Sistem Informasi 2014 (KNSI 2014)*. STMIK Dipanegara Makassar, 27 Februari – 01 Maret 2014.

[12] Munir, R., 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung: Informatika.

[13] Alatas, P., 2009. *Implementasi Teknik Steganografi Dengan Metode LSB Pada Citra Digital*. Jakarta: Universitas Gunadarma.

[14] Munir, R., 2006. *Kriptografi*. Bandung: Informatika.

[15] Sutoyo, T., 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: ANDI.

[16] Wandani, H.; Budiman, M.A; Sharif, A., 2012. Implementasi Sistem Keamanan Data dengan Menggunakan Teknik Steganografi End of File (EOF) dan Rabin Public Key Cryptosystem. *Jurnal Alkhawarizimi*, 1 (1).

[17] Kurniawan, Y. 2004. *Keamanan Internet dan Jaringan Telekomunikasi.* Bandung: Informatika.

[18] Kromodimoeljo, S., 2010. *Teori & Aplikasi Kriptografi*. Jakarta: SPK IT Consulting.

[19] Department of Computer & Information System Engineering, 2012. *Practical Workbook: Information Theory*. 4th edition. Karachi, Pakistan: NED University of Engineering & Technology.