TEKNIK PENYEMBUNYIAN PESAN PDF TERENKRIPSI MENGGUNAKAN ALGORITMA KRIPTOGRAFI VERNAM CIPHER DAN STEGANOGRAFI END OF FILE (EOF) DALAM MEDIA GAMBAR

[1]Marsela Sutikno Dibiyo, [2]Aisyatul Karima, S.Kom, M.Cs
Program Studi Teknik Informatika – S1
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Jl. Nakula 1 No. 5-11. Semarang
[1]111201105905@mhs.dinus.ac.id, [2]aisyatul.karima@gmail.com

ABSTRAK

Seiring perkembangan teknologi yang semakin lama semakin pesat, pengiriman pesan atau informasi yang penting dari satu orang ke orang lain dirasa sangat tidak aman lagi. Berawal dari pesan format word, setelah itu munculah aplikasi PDF yang bisa mengirim pesan dan tidak bisa dirusak oleh orang lain, tetapi lama kelamaan dengan teknologi yang semakin berkembang pesat, pesan PDF juga bisa dirusak serta pesan itu juga tidak dapat dirahasiakan, walaupun orang lain tidak dapat merusaknya, namun dapat mengerti informasi yang terdapat di dalam pesan tersebut. Oleh karena itu, diperlukan suatu sistem yang bisa menjaga keaslian pesan dan kerahasiaan pesan tersebut. Dengan cara mengubah pesan asli menjadi pesan acak yang sudah dienkripsi oleh suatu kunci agar orang yang tidak berwenang tidak dapat mengetahui pesan tersebut. Selain itu pesan yang sudah diubah menjadi pesan acak harus disembunyikan juga agar pihak ketiga atau pihak yang tidak berwenang tidak merasa curiga terhadap pesan yang diacak tersebut. Salah satu metode yang digunakan untuk mengacak pesan rahasia tersebut dengan menggunakan Algoritma Kriptografi Vernam Cipher dan Steganografi End of File untuk menyembunyikan pesan yang sudah dienkripsi ke dalam gambar agar tidak menimbulkan kecurigaan pada pihak ketiga, karena perubahan yang terjadi tidak tampak berbeda secara kasat mata. Dengan adanya sistem tersebut dapat menjaga keamanan data serta kerahasiaan data yang hanya dapat diketahui oleh pihak yang berwenang. Selain itu keaslian data dapat terjaga tanpa adanya kerusakan data yang dikirimkan.

Kata kunci : kriptografi, *vernam cipher*, steganografi, *end of file.*

I. PENDAHULUAN

Seiring perkembangan teknologi, teknik dan metode penyampaian pesan rahasia semakin beragam. Terdapat berbagai bentuk pesan rahasia seperti pesan teks, pesan citra, pesan audio dan pesan video yang umum digunakan.

Berbagai organisasi, perusahaan, atau pihak - pihak lain telah memanfaatkan pesan teks untuk mengirim informasi penting. Saat ini, perkembangan pesan teks semakin pesat. Berawal dari pesan teks yang menggunakan microsoft word ke PDF, dahulu orang-orang menggunakan microsoft word untuk menyimpan pesan yang akan disampaikan ke penerima, karena takutnya pesan itu diganti atau dirusak oleh pihak-pihak lain maka pesan berformat word itu kemudian dirubah menjadi pesan yang berformat PDF.

Namun perkembangan teknologi yang lama kelamaan semakin meningkat, keamanan pesan rahasia dalam format PDF sudah dirasa menjadi tidak aman lagi, karena semakin banyak aplikasi yang bisa merubah atau merusak pesan dalam format PDF. Misalnya dalam hal mengganti nama penulis makalah milik orang lain dalam format PDF, dengan berbagai aplikasi tersebut sangatlah mudah untuk merubah nama yang diinginkan [8]. Oleh sebab itu, si pengirim pesan membutuhkan bantuan untuk memenuhi kebutuhan keamanan akan informasi atau data yang akan dikirim ke penerima.

Salah satu sistem keamanan yang digunakan pada saat ini adalah kriptografi. Keunggulan dari kriptografi adalah kemampuan penyandian pesan sehingga pesan terlihat sepeti diacak. Kriptografi tidak sekedar berupa kerahasiaan data (privacy) saja, tapi juga bertujuan untuk menjaga integritas data (data integrity), keaslian data (authentication) dan anti penyangkalan (non-repudiation) [1] [2].

Algoritma kriptografi yang akan digunakan adalah algoritma kriptografi simetris dan bersifat *vernam cipher* sehingga data hasil enkripsi (cipherteks) mempunyai ukuran yang sama dengan data asli (plainteks). Teknik kriptografi simetris dipilih karena diharapkan dengan algoritma ini proses enkripsi – dekripsi data dapat dilakukan dengan waktu yang lebih cepat dibandingkan dengan algoritma kriptografi kunci publik (asimetris).

Namun, kelemahan dari algoritma *Vernam Cipher* ini adalah hasil enkripsi

yang masih tampak oleh mata manusia, sehingga mudah dikenali sebagai data yang telah mengalami proses enkripsi. Oleh sebab itu, perlu menyembunyikan data yang sudah dienkripsi ke dalam gambar supaya pihak yang tidak berkepentingan tidak merasa curiga dalam melihat gambar tersebut. Dan teknik penyembunyian data ke dalam gambar disebut dengan teknik steganografi.

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Salah satu algoritma yang digunakan dalam steganografi yaitu *End of File* (EoF).

II. DASAR TEORI

A. VERNAM CIPHER

Vernam cipher merupakan algoritma kriptografi yang ditemukan oleh Mayor J. Maugborne dan G. Vernam. Algoritma ini merupakan algoritma berjenis *symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara *stream cipher* dimana *cipher* berasal dari hasil XOR antara bit *plainteks* dan bit *key* [1]. Algoritma *vernam cipher* diadopsi dari *one-time pad cipher*, dimana dalam hal ini karakter diganti dengan bit (0 atau 1). Dengan kata lain, *vernam cipher* merupakan versi lain dari *one-time pad cipher*.

Dalam proses enkripsi, *cipherteks* diperoleh dengan melakukan penjumlahan modulo 2 satu bit plainteks dengan satu bit kunci, seperti terlihat pada rumus di bawah ini :

$$\mathbf{c_1 = (p_1 + k_1) \bmod 2} \qquad \mathbf{(3)}$$

Dimana :
c_1 = cipherteks
p_1 = plainteks
k_1 = kunci

Sedangkan dalam proses dekripsi, untuk mendapatkan kembali *plainteks*, diperoleh dengan melakukan penjumlahan modulo 2 satu bit *cipherteks* dengan satu bit kunci :

$$\mathbf{p_1 = (c_1 - k_1) \bmod 2} \qquad \mathbf{(4)}$$

Pada *cipher* aliran, bit hanya mempunyai dua buah nilai, sehingga proses enkripsi hanya menyebabkan dua keadaan pada bit tersebut, yaitu berubah atau tidak berubah. Dua keadaan tersebut ditentukan oleh kunci enkripsi yang disebut dengan aliran-bit-kunci (*keystream*). Oleh karena operasi penjumlahan *modulo* 2 identik dengan operasi bit dengan operasi XOR, maka persamaan 3 dapat ditulis secara sederhana sebagai berikut [1]:

$$c_1 = p_1 \text{ XOR } k_1 \quad (5)$$

Sedangkan pada proses pendekripsian dituliskan :

$$p_1 = c_1 \text{ XOR } k_1 \quad (6)$$

Dalam operator logika XOR, hasil akan T (benar) apabila salah satu dari kedua operand (tetapi tidak keduanya) bernilai T atau 1. Atau dengan kata lain, apabila diaplikasikan dalam bit maka operator XOR akan menghasilkan 1 jika dan hanya jika salah satu operand bernilai 1.
Contoh :

X : 00111010 10101011
Y : 10100100 01010101
Hasil : 10011110 11111110

Sedangkan suatu bilangan dalam biner apabila di XOR-kan dengan dirinya sendiri akan menghasilkan 0.
Contoh :

X : 01010101 10101010
Y : 01010101 10101010
Hasil : 00000000 00000000

B. END OF FILE

Metode *End of File* (EoF) merupakan salah satu metode yang digunakan dalam steganografi. Metode ini menyembunyikan pesan rahasia dengan cara menambahkan bit-bit pesan yang akan disembunyikan ke akhir file citra penampung.

Teknik ini menggunakan cara dengan menyisipkan data pada akhir file. Sehingga, tidak akan mengganggu kualitas data awal yang akan disisipkan pesan. Namun, ukuran file setelah disisipi pesan rahasia akan bertambah [2] [5]. Sebab, ukuran file yang telah disisipkan pesan rahasia sama dengan ukuran file sebelum disisipkan pesan rahasia ditambah dengan ukuran pesan rahasia yang disisipkan. Untuk mengenal data yang disisipkan pada akhir file, diperlukan suatu tanda pengenal atau simbol pada awal dan akhir data yang akan disisipkan.

Proses penyisipan pesan dengan metode EoF dapat dituliskan dalam algoritma sebagai berikut [5] :

1. Inputkan *cipherteks* yang akan disisipkan.
2. Inputkan citra yang akan menjadi media penyisipan *cipherteks*.
3. Baca nilai setiap *pixel* citra.
4. Tambahkan *cipherteks* sebagai nilai akhir *pixel* citra dengan diberi karakter penanda sebagai penanda akhir *cipherteks*.
5. Petakan menjadi citra baru.

Proses pengambilan *cipherteks* dari media menggunakan metode *End of File* adalah sebagai berikut :

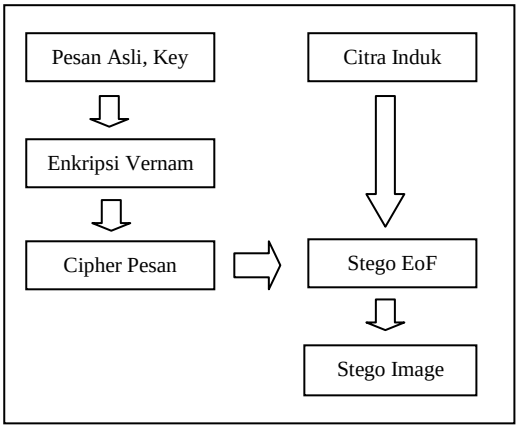
1. Inputkan citra yang telah disisipkan *cipherteks* (*stego image*).
2. Baca nilai *pixel stego image* yang terdapat pada baris terakhir matriks *pixel* citra.
3. Ambil *cipherteks* yang terdapat pada *stego image*, yaitu nilai *pixel* awal yang terdapat pada baris terakhir matriks *pixel* citra sampai nilai desimal karakter penanda.

III. METODE PENELITIAN

Berikut ini model yang diusulkan :

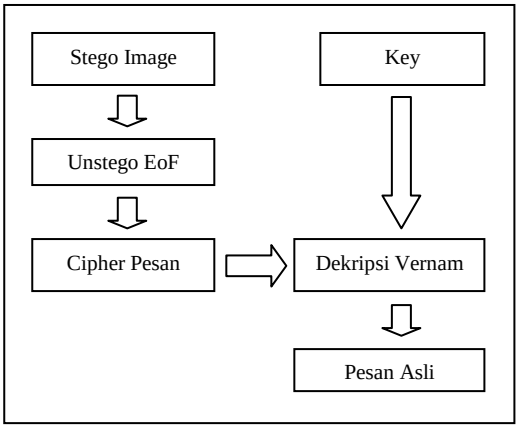
Pesan Asli atau *plainteks* dengan kunci dienkripsikan menggunakan algoritma kriptografi *vernam cipher* dan menghasilkan data berupa *cipherteks.* Selanjutnya *cipherteks* disisipkan ke dalam sebuah citra mengunakan steganografi *end of file* yang akan menghasilkan *Stego Image* yang terdapat pada gambar 1.

Gambar 1. Metode Enkripsi Data



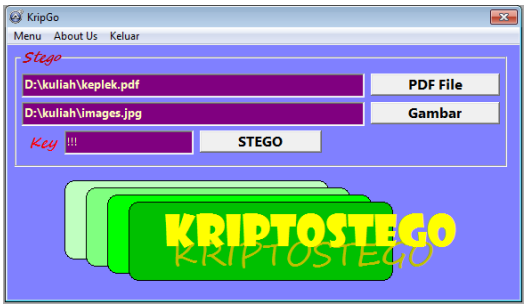
Dekripsi Data dengan cara unstego *stego image* hasil enkripsi diatas akan menghasilkan *cipherteks*, lalu dengan kunci yang sama untuk mengenkripsi pesan awal tadi digunakan untuk mendekripsi *cipherteks* dan menghasilkan pesan asli atau *plainteks*. Terdapat dalam gambar 2 dibawah ini.

Gambar 2. Metode Dekripsi Data



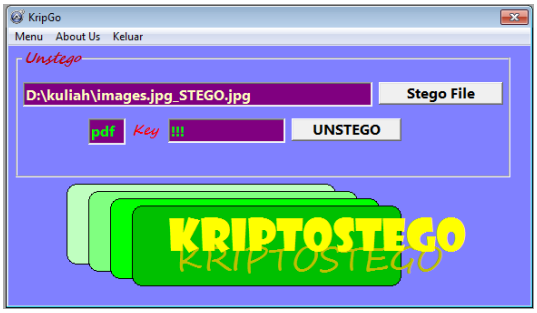
IV. HASIL PENGUJIAN

Berikut ini merupakan tampilan aplikasi :



Gambar diatas merupakan proses stego. File pdf yang akan disembunyikan adalah "keplek.pdf" pada partisipasi "D" dalam *directory* "kuliah".

Gambar merupakan file induk dimana file ini sebagai tempat persembunyian dari file asli. Tahap kedua adalah memilih file induk dimana file asli akan disembunyikan. Dalam hal ini, penulis memilih gambar "images.jpg" sebagai tempat penyembunyian file asli. Setelah itu masukan kata kunci dan tekan tombol stego untuk memproses stego file tersebut.



Pada gambar diatas merupakan proses unstego file yang akan mengembalikan file stego ke file asli.

Pengujian aplikasi dilakukan dengan metode *random sampling*, yakni mengambil masing – masing 15 sampel file pdf dan 15 sampel data gambar guna diujikan pada aplikasi.

File kemudian diujikan pada aplikasi untuk melihat hasil akhir dari enkripsi dan stego yang diproses pada aplikasi. Untuk pengujian, penulis memasukkan kunci sama untuk tiap-tiap percobaan, kunci yang dimasukkan adalah "123".

Masing – masing data akan diujikan ke aplikasi. Pertama, data pdf akan diproses dengan algoritma *vernam cipher*, proses ini disebut dengan proses enkripsi. Pada proses enkripsi bertujuan mengacak *file* pdf atau dapat disebut pesan rahasia agar pihak lain yang tidak berwenang tidak dapat menemukan makna atau pesan rahasia di dalamnya.

Kemudian hasil dari proses enkripsi tadi akan diproses guna menyembunyikan *file* hasil enkripsi tersebut dalam sebuah citra atau gambar. Proses itu sendiri disebut dengan proses stego. Untuk kedua proses ini, baik enkripsi maupun stego,

atau sebaliknya proses dekripsi maupun unstego, pengguna akan diminta memasukkan sebuah kunci. Dimana kunci ini sama antara kedua proses tersebut, untuk mempercepat waktu dalam proses enkripsi maupun dekripsi.

No	File PDF	Gambar	Ukuran File PDF	Ukuran Gambar	Ukuran Gambar Stego	Ukuran File PDF Unstego
1	Keplek.pdf	images.jpg	68 KB	8 KB	75 KB	68 KB
2	Abstrak.pdf	aman.jpg	5 KB	87 KB	91 KB	5 KB
3	Eof.pdf	burung.jpg	240 KB	9 KB	248 KB	240 KB
4	Eof2.pdf	gembok.jpg	311 KB	199 KB	510 KB	311 KB
5	Eof3.pdf	kelinci.jpg	238 KB	6 KB	244 KB	238 KB
6	Judul TA.pdf	krip.jpg	101 KB	42 KB	142 KB	101 KB
7	Pengajuan.pdf	kucing.jpg	101 KB	5 KB	106 KB	101 KB
8	Stream1.pdf	panda.jpg	187 KB	9 KB	196 KB	187 KB
9	Tes.pdf	penguin.jpg	299 KB	7 KB	305 KB	299 KB
10	Test.pdf	security.jpg	279 KB	300 KB	579 KB	279 KB
11	Serpen.pdf	angry.jpg	40 KB	8 KB	48 KB	40 KB
12	Mars.pdf	bibo.jpg	126 KB	7 KB	132 KB	126 KB
13	Pesan.pdf	bird.jpg	282 KB	11 KB	293 KB	282 KB
14	Rijndael.pdf	bob.jpg	279 KB	9 KB	288 KB	279 KB
15	Cipher.pdf	cool.jpg	103 KB	7 KB	109 KB	103 KB

Berdasarkan tabel 4 diatas menunjukkan adanya perbedaan antara ukuran gambar sebelum proses stego dan setelah proses stego, dimana ukuran akan bertambah besar. Hal ini disebabkan karena prinsip metode steganografi *End of File* ini sederhana, yaitu menambahkan *file* pdf dibelakang *file* gambar. Dengan proses seperti itu, tentunya jumlah ukuran *file* itu akan bertambah. Pertambahan ukuran *file* ini dapat dirumuskan yaitu besar ukuran *file* pdf ditambahkan dengan besar ukuran *file* gambar itu sendiri. Seperti dapat dilihat di tabel 7, ukuran *file* images.jpg setelah proses stego menjadi 75 KB. Jika dilihat ukuran data sebelumnya, yaitu *file* pdf yang telah dienkripsi sebesar 68 KB jika ditambah dengan ukuran *file* gambar sebelum stego sebesar 8 KB maka angka 75 KB adalah hasil dari penambahan ukuran kedua *file* tersebut.

Namun, meskipun terjadi penambahan besar ukuran *file* gambar, metode steganografi *end of file* ini memiliki kelebihan dibandingkan dengan metode lainnya yaitu secara kasat mata, tidak ada perbedaan dari kedua gambar sebelum dan sesudah proses stego.

Untuk membuktikan kelebihan metode steganografi *end of file* ini, perlu dilakukan perbandingan antara citra sebelum dan sesudah proses stego.

Maka, berikut penulis tampilkan beberapa citra induk sebelum dan sesudah stego :

Gambar 3. Kelinci.jpg sebelum stego

Gambar 4. Kelinci.jpg setelah stego

Dari pengujian yang telah dilakukan, dapat menunjukkan bahwa untuk proses stego dengan metode steganografi *end of file* tidak akan merubah kualitas gambar sehingga pada gambar yang sudah di stego tidak akan menimbulkan kecurigaan untuk pihak yang tidak berwenang maupun yang akan merusak *file* tersebut, namun akan memperbesar ukuran *file* karena merupakan penambahan antara ukuran *file* pesan rahasia dan *file* induk berupa *file* gambar.

V. KESIMPULAN

A. Kesimpulan

Dari penelitian yang telah dilakukan, maka penulis mendapatkan beberapa kesimpulan, yaitu sebagai berikut :

1. Aplikasi KriptoStego dapat mengamankan data rahasia dengan baik menggunakan kriptografi *vernam cipher* dan steganografi *end of file*, karena pihak ketiga tidak menyadari adanya pesan rahasia dalam sebuah gambar yang sudah disembunyikan.
2. Hasil gambar dengan metode *end of file* setelah disisipkan pesan hasil enkripsi *vernam cipher* dari proses KriptoStego tidak mengalami perubahan gambar secara kasat mata karena metode steganografi *end of file* yag digunakan tidak akan mengubah kualitas gambar / citra.
3. Aplikasi kriptografi *vernam cipher* dan steganografi *end of file* ini dapat mengenkripsi *file* dan mendekripsi *file* dengan baik. Karena terbuktinya bahwa dengan aplikasi ini dapat menyimpan pesan rahasia dari pengirim untuk sampai ke penerima tanpa adanya kerusakan pesan setelah proses unstego dan pengirim dapat menjamin tidak adanya perubahan file setelah sampai ke penerima. Namun, ukuran *file* gambar

setelah proses KriptoStego mengalami perubahan bertambah besarnya ukuran tersebut karena terjadi pertambahan ukuran dari kedua *file* tersebut.

B. Saran

Dari kesimpulan yang telah diuraikan penulis diatas, maka penulis memberikan saran untuk pengembangan aplikasi KriptoStego adalah sebagai berikut :

1. Aplikasi dapat dikembangkan lagi menggunakan metode lain yang juga cocok untuk teknik penyembunyian pesan.
2. Aplikasi dapat dikembangkan dengan bahasa pemrograman lain yang lebih powerfull dan lebih cepat.

VI. REFERENSI

[1] Nathasia, N. D., & Wicaksono, A. E. (2011). Penggunaan Teknik Kriptografi Stream Cipher untuk Pengamanan Basis Data. *Jurnal Basis Data, ICT Research Center UNAS*, 6(1), 1-22.

[2] Sukrisno, & Utami, E. (2007). Implementasi Steganografi Teknik EoF dengan Gabungan Enkripsi Rijndael, Shift Cipher dan Fungsi Hash MD5. *Seminar Nasional Teknologi 2007 (SNT 2007)*, (November), 1-16.

[3] Iswahyudi, C., Setyaningsih, E., & Widyastuti, N. (2012). Pengamanan Kunci Enkripsi Citra pada Algoritma Super Enkripsi Menggunakan Metode End Of File. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III*, (November), 278-285.

[4] Aditya, Y., Pratama, A., & Nurlifa, A. (2010). Studi Pustaka untuk Steganografi dengan Beberapa Metode. *Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*, 2010, 32-35.

[5] Wandani, H., Budiman, M., & Sharif, A. (2012). Implementasi Sistem Keamanan Data dengan Menggunakan Teknik Steganografi End Of File (EOF) dan Rabin Public Key Cryptosystem. *Alkhawarizmi*. Retrieved from http://jurnal.usu.ac.id/index.php/alkhawarizmi/article/view/500

[6] Anonimous, ASCII Table and Extended ASCII Table, www.asciitable.com, 10 Agustus 2009

[7] Arifpriyanto, B. (2013). Penyembunyian Pesan Text

Terenkripsi Menggunakan Metode Kriptografi Stream Cipher dan Steganografi End Of File (EOF) dengan File Induk PDF. *Dokumen Karya Ilmiah Tugas Akhir Program Studi Teknik Informatika – S1 Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semaranng 2013*, 2013, 1-6.

[8] Ramnul. (2012). Cara Mengedit File PDF. Retrieved from http://ramnul.us/2012/07/cara-edit-file-pdf-tanpa-adobe-acrobat.html

[9] Sholeh, M., & Hamokwarong, J.V. (2011). Aplikasi Kriptografi Dengan Metode Vernam Cipher dan Metode Permutasi Biner. *Momentum*, 7(2), 8-13.