Jurnal TEKNOINFO, Vol. 11, No. 1, 2017, 1-6. ISSN 1693 0010 (print)

TEKNIK STEGANOGRAFI UNTUK PENYEMBUNYIAN PESAN TEKS MENGGUNAKAN ALGORITMA *GIFSHUFFLE*

Dedi Darwis

Sistem Informasi, Universitas Teknokrat Indonesia
Jl. H.ZA Pagaralam, No 9-11, Labuhanratu,Bandarlampung
Email : darwisdedi@teknokrat.ac.id

Abstrak

Keamanan informasi merupakan kebutuhan yang sangat penting diera digital ini karena maraknya pengguna internet dapat membuka hal-hal yang bersifat pribadi. Salah satu cara untuk mengamankan informasi adalah menggunakan teknik steganografi menggunakan algoritma gifshuffle, akan tetapi algoritma ini memiliki keterbatasan dalam penampungan data. Tujuan dari penelitian ini adalah mengamankan pesan teks berupa laporan keuangan menggunakan teknik steganografi dengan daya tampung ukuran pada cover image dapat lebih besar lagi. Berdasarkan hasil pengujian yang dilakukan yaitu gambar yang disisipkan pesan teks dapat menampung ukuran data yang relatif lebih besar yaitu mencapai 1,82 MB.

Kata kunci: *Steganografi, Gif, Stego Image, Cover Image, GifShuffle.*

1. Pendahuluan

A. Latar Belakang Masalah

Perkembangan teknologi dan informasi seperti sekarang ini kecepatan dalam memperoleh atau mengakses informasi sangatlah penting. Dengan mengedepankan akurasi sebuah data yang sampai dan dibutuhkan pengiriman informasi yang cepat maka semua pekerjaan juga dapat dilakukan dengan cepat pula.

Oleh karena itu, keamanan data dan informasi menjadi sebuah kebutuhan vital bagi para pengguna internet saat ini agar privasi mereka bisa tetap terjaga. Salah satu teknik pengamanan data yang sering digunakan adalah steganografi. Steganografi adalah teknik menyamarkan atau menyembunyikan pesan ke dalam sebuah media pembawa (*carrier)*[6]. Kelebihan steganografi terletak pada sifatnya yang tidak menarik perhatian atau kecurigaan orang lain[4].

Stegonografi memiliki banyak teknik algoritma dalam proses penyembunyian informasi, pada penelitian ini algoritma steganografi yang digunakan yaitu *GifShuffle* karena algoritma ini memanfaatkan media citra berformat *GIF* yang berukuran relatif kecil dan bersifat *lossless* yang berarti bahwa citra tidak mengalami kehilangan kualitas ketika dikompresi atau disisipi data[6], maka dari itu antara citra yang asli dan citra yang sudah disisipkan data perbedaan ukuran dan kualitas citra tidak terlalu berbeda sehingga pihak lain tidak curiga bahwa gambar berformat *Gif* tersebut sudah disisipi data atau pesan rahasia. Algoritma gifshuffle memiliki kekurangan yaitu hanya dapat disisipkan pesan dengan ukuran yang relatif kecil yaitu 209 *byte*[7], maka tujuan dilakukan penilitian ini adalah membuat algoritma gifshuffle agar dapat disisipkan pesan lebih banyak lagi dengan menggunakan cover image berupa gambar berformat gif dan data yang dijadikan sebagai objek yang akan disembunyikan adalah laporan keuangan berformat *txt.*

B. Landasan Teori

1. Steganografi

Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi rahasia di dalam suatu informasi lainya[2].

Steganografi sudah digunakan sejak dahulu kala sekitar 2500 tahun yang lalu untuk kepentingan politik, militer, diplomatik, serta untuk kepentingan pribadi. Dan sesungguhnya prinsip dasar dalam steganografi lebih dikonsentrasikan pada kerahasian komunikasinya bukan pada datanya[1].

2. Kriteria Steganografi

Kriteria yang harus diperhatikan dalam melakukan penyembunyian data dengan menggunakan teknik steganografi adalah sebagai berikut :

1) *Imperceptibility* : Keberadaan pesan dalam media penampung tidak dapat dideteksi.
2) *Fidelity* : Mutu media penampung setelah ditambahkan pesan rahasia tidak jauh berbeda dengan mutu media penampung sebelum ditambahkan pesan.
3) *Recovery* : Pesan rahasia yang telah disisipkan dalam media penampung harus dapat diungkap kembali.
4) *Robustness* : Pesan yang disembunyikan harus tahan terhadap berbagai operasi manipulasi yang dilakukan pada media penampung.

Karakteristik steganografi yang baik adalah *imperceptibility* tinggi, *fidelity* tinggi, *recovery* maksimum dan *robustness* tinggi [3].

3. Algoritma GifShuffle

GifShuffle adalah sebuah algoritma *steganography* yang digunakan untuk menyembunyikan pesan dalam berkas citra dengan format *GIF*[7].

Algoritma *gifshuffle* ini ditemukan oleh Matthew Kwan lulusan sarjana ilmu komputer dari *University Of Melbourne*. Beliau pernah belajar seni desain kripto selama 12 bulan bertugas di *Australian Defence Force Academy Centre for Computer Security Research* pada tahun 1991. Beliau juga pemilik dan pendiri *Unicrypt* Pty Ltd, sebuah perusahaan yang bergerak dibidang enkripsi email. *Gifshuffle* adalah program untuk menyembunyikan pesan kedalam gambar gif dengan cara men-*shuffle colourmap*. *Gifshuffle* bekerja disemua gambar gif, termasuk transparasi dan animasi GIF. Algoritma *gifshuffle* pada intinya memanfaatkan *header file* GIF yang menyimpan palet warna sebagai media penyisipan pesan[5]. Gambar Gif berisi *colourmap* sampai 256 entri dan menghasilkan kapasitas penyimpanan maksimum 1683 bit. Pada prosesnya Algoritma *Gifshuffle* memiliki langkah-langkah sebagai berikut[8] :

1) Dimulai dengan pesan yang ingin disembunyikan, kemudian pesan dikonversi kedalam bilangan biner 1 dan 0. Berinama M pada pesan yang telah dikonversi ke dalam bilangan biner, kemudian tambahkan nilai 1 pada bilangan biner.
2) Hitung jumlah warna terkandung dalam gif yang ingin disisipkan, ibaratkan jumlah warna ini dengan "n". Jika nilai m > n!-1 maka proses penyisipan tidak dapat dilakukan.
3) Kemudian urutkan palet warna citra secara natural dan setiap warna dikonversi kedalam bilangan integer dengan rumus sebagai berikut : **(*Red* * 65536 + *Green* * 256 + *Blue*).**
4) Lakukan iterasi *variable* ***i*** mulai dari 1 sampai n (jumlah warna). Warna indeks ke (n – 1) dipindah ke (m mod ***i***), $\frac{m}{i}$.
5) Pada tahap kelima, apabila ada beberapa warna yang menempati indeks yang sama, maka setiap warna yang menempati indeks tersebut akan bergeser sekali ke indeks berikutnya.
6) Urutan palet warna kemudian dimasukkan kedalam berkas citra GIF untuk menghasilkan citra yang telah disisipi pesan.

2. Pembahasan

A. Alur Proses Algoritma *GifShuffle*

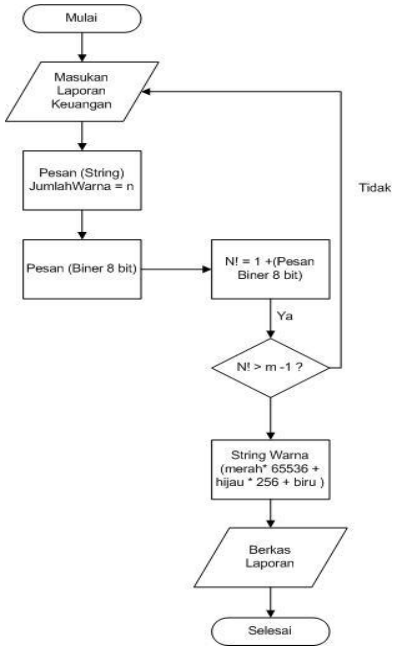
Algoritma *Gifshufle* pada intinya memanfaatkan *header file GIF* yang menyimpan palet warna sebagai media penyisipan pesan. Algoritma ini tidak terjadi perubahan apapaun dalam data berkas dengan format *GIF*. Sehingga menambah aspek *robustness* dari algoritman ini.Sesuai dengan namanya *GifShuffle*akan melakukan "*Shuffle*" terhadap palet warna dari sebuah berkas *GIF*. "Shuffle" jika diterjemahkan ke dalam bahasa Indonesia berari memutar. Sehingga dapat diartikan bahwa *GifShuffle* adalah algoritma yang memanfaatkan penukaran posisi ke 256 palet warna dalam berkas citra berformat *GIF*. Hal tersebut aman dilakukan karena dua buah berkas *GIF* dengan palet warna yang berbeda akan ditampilkan secara sama persis.

Dengan dilakukannya penukaran posisi maka akan dapat diperoleh sebuah informasi berkaitan dengan perbedaap posisi dengan posisi awal. Sebagai contoh jika kita mempunyai 52 kartu remi maka kita akan dapat mengurutkan kartukartu tersebut dalam 52! cara. Dengan kata lain jika kita diberikan n buah kartu maka kita dapat menyimpan log2(n!) bit informasi berdasarkan pengurutannya.

Langkah-langkah Algoritma *GifShuffle* :

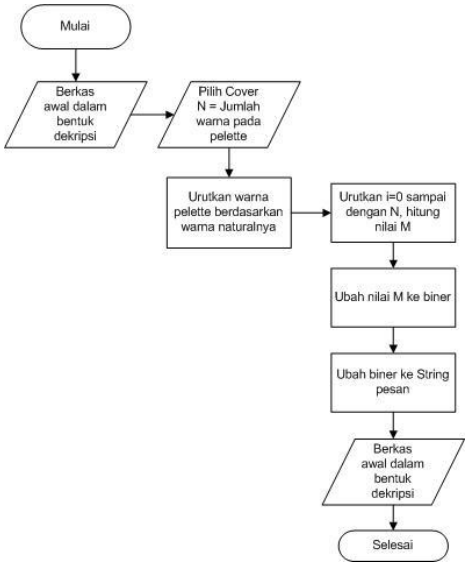
1) Dimulai dengan dengan pesan yang akan disisipkan. Pesan tersebut akan diubah kedalam sebuah bentuk biner dengan representasi 1/0.
2) Anggap kumpulan representasi biner yang tadi diperoleh sebagai sebuah angka. Biasanya langkah ini akan menghasilkan sebuah bilangan yang sangat besar karena konversi dari biner yang besar . Namakan bilangan yang diperoleh ini sebagai M.
3) Hitung jumlah warna yang terkandung dalam berkas GIF yang ingin disisipkan. Namakan jumlah yang diperoleh ini sebagai N. Apabila M > N!-1 maka pesan yang ingin disisipkan berukuran terlalu besar sehingga proses penyisipan tidak dapat dilakukan.
4) Urutkan warna dalam palet warna sesuai dengan urutan yang "natural". Setiap warna dengan format RGB dikonversikan ke bilangan integer dengan aturan (merah* 65536 + hijau * 256 + biru). Kemudian diurutkan berdasarkan besar bilangan interger yang mewakili warna tersebut.
5) Lakukan iterasi terhadap variabel I dengan nilai I dari 1 sampai N. Setiap warna dengan urutan N-i dipindahkan keposisi baru yaitu M mod i, kemudian M dibagi dengan i.
6) Kemudian palet warna yang baru hasil iterasi pada langkah 5 dimasukkan ke dalam palet warna berkas GIF. Apabila ada sebuah tempat yang diisi oleh 2 buah warna maka warna yang sebelumnya menempati tempat tersebut 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 GIF ini akan dikompresi ulang dengan palet warna yang baru untuk menghasilkan berkas yang baru dengan ukuran dan gambar yang sama namun telah disisipi pesan.

Gambar 1 adalah *flowchart* Algoritma *GifShuffle* proses enkripsi :



Gambar 1. *Flowchart GifShuffle Proses Enkripsi*

Proses algoritma *gifshuffle* setelah melalui tahap enrkipsi di mana memasukan file ke dalam media citra digital selanjutnya adalah tahap dekripsi, proses ini yaitu melakukan *extraction* atau pemisahan isi laporan dari media citra digital, proses dekripsi dari algoritma *gifshuffle* dapat diperhatikan pada gambar 2.



Gambar 2. *Flowchart GifShuffle proses Dekripsi*

B. Simulasi dan Pengujian Enkripsi *GifShuffle*

Pada proses simulasi dan pengujian enkripsi steganografi akan dilakukan dengan cara memasukkan citra digital berformat *gif* dan data yang akan disisipkan berformat *txt*. Pengujian ini akan dilakukan dengan memperhatikan apakah gambar dan *file* dapat dilakukan proses *embedding* dan jika berhasil maka akan dilihat tiga aspek yaitu *Imperect*, *Fidelity*, dan *Recovery*. Adapun proses pengujian dapat dilihat pada gambar 3.

Gambar 3. *Proses Pengujian Enkripsi GifShuflle*

Pada gambar 3 menunjukkan bagaimana cara menggunakan enkripsi *GifShuffle*, langkah pertama pada proses ini adalah mengambil gambar berformat *gif*, gambar yang dipilih merupakan gambar animasi atau gambar yang dapat bergerak, lalu ketika dipilih tombol Simpan Gambar maka gambar akan disimpan sesuai dengan nama laporan dan periode laporan, gambar dipisah dari gambar aslinya karena nantinya untuk membandingkan antara gambar asli dan gambar yang sudah dienkripsi. *Password* dan konfirmasi *password* diisi oleh *user*. Setelah semua sudah diisi lalu pilih tombol Proses Enkripsi agar antara gambar dan *file* dapat dilakukan proses *embedding.* Pengujian berikutnya akan dilakukan apabila ukuran *file* terlalu besar maka laporan keuangan tidak dapat dienkripsi, hasil pengujianya dapat dilihat pada gambar 4.

Gambar 4. *Pengujian dengan ukuran File Terlalu Besar*

Hasil dari proses enkripsi akan diuji melalui 2 aspek sebagai berikut :

1) Imperceptibility

Pada pengujian ini akan dilakukan pengujian secara manual dengan melibatkan 30 responden yang akan menilai perbedaan citra asli dan citra stego. Responden terdiri dari 50% mahasiswa ilmu

komputer dan 50% mahasiswa di luar ilmu komputer. Pesan yang akan disisipkan berukuran 1.868 *byte*, sedangkan cover image berukuran 20.959 *byte* dan memiliki dimensi 289 x 197 piksel. *File* dipilih dengan ukuran yang besar karena semakin besar ukuran *file* laporan maka semakin kecil kualitas gambar yang dihasilkan. Hal ini cukup mewakili penilaian tahap pengujian menggunakan *imperceptibilty*.

Penilaian pada kuisioner dibagi menjadi tiga kriteria, yaitu berbeda, sedikit berbeda, dan tidak berbeda. Kriteria berbeda dapat dipilih apabila responden dapat melihat adanya perubahan warna yang sangat nyata terhadap *cover-image*. Perubahan warna yang dihasilkan terjadi pada hampir semua bagian gambar. Dengan kata lain, *stego image* yang dihasilkan tampak jauh berbedadari *cover-image*. Kriteria sedikit berbedadapat dipilih apabila responden hanya melihat beberapa perbedaan warna. Perubahan warna yang dihasilkan hanya pada tempat-tempat tertentu saja. Responden hanya melihat sedikit perbedaan pada gambar. Kriteria tidak berbeda dapat dipilih apabila responden tidak melihat adanya perbedaan warna yang dihasilkan. Responden tidak melihat perubahan warnapada seluruh bagian gambar. Hasil kuisioner untuk pengujian *imperectibility* dapat dilihat pada tabel 1.

Tabel 1. *Hasil Kuisioner Pengujian imperceptibilty*

Stego Key	Jumlah		
	Berbeda	Sedikit Berbeda	Tidak Berbeda
123	0	2	28
Budilhur	0	2	28
Bmtmentari	0	4	26
dedidarwis-ok	0	6	24
Teknokrat	0	6	24
Ilmukomputer	0	4	26
Steganog	0	2	28
Password	0	2	28

Berdasarkan pada tabel 1, dapat dilihat hasil kuisioner, yaitu 93% responden berpendapat bahwa *stego-image* dengan *stego-key* "123" tidak berbeda dengan *cover-image*, 93% responden berpendapat bahwa *stego-image* dengan *stego-key*"budiluhur" tidak berbeda dengan *coverimage*, 87% responden berpendapat bahwa *stego-image* dengan *stego-key* "bmtmentari" tidak berbeda dengan *cover-image*, 80% responden berpendapat bahwa *stego-image* dengan *stego-key* "dedidarwis-ok" tidak berbeda dengan *cover-image*, 80% responden berpendapat bahwa *stego-image* dengan *stego-key* "teknokrat" tidak berbeda dengan *cover-image*, 87% responden berpendapat bahwa *stego-image* dengan *stego-key* "ilmukomputer" tidak berbeda dengan *cover-image*, 93% responden berpendapat bahwa *stegoimage* dengan *stego-key* "steganog" tidak berbeda dengan *cover-image*, 93% responden berpendapat bahwa *stego-image* dengan *stego-key* "password" tidak berbeda dengan *cover-image*.

2) Fidelity

Merupakan pengujian terhadap aspek mutu citra hasil steganografi dengan mengukur nilai *MSE (Mean Square Error)* dan *PSNR (Peak Signal to Noise Ratio)*. Keduanya merupakan sebuah nilai yang memiliki satuan *db(desibels)*. Semakin rendah nilai *MSE* maka kualitas citra semakin baik, sedangkan mutu *stego-image* dikatakan baik jika nilai *PSNR* 40 *db* atau lebih. Cara menghitung nilai *MSE* yaitu didapat dengan rumus 1.

$$MSE = \frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N}(S_{xy} - C_{xy})^2 \quad \ldots\ldots (1)$$

Sedangkan menghitung *PNSR* didapat dengan rumus 2.

$$PNSR = 10\,log_{10}(\frac{C_{max}^2}{MSE}) \quad \ldots\ldots (2)$$

Keterangan :
Cmax adalah nilai *pixel* terbesar dari keseluruhan citra
X dan Y adalah koordinat suatu titik pada citra
M dan N adalah dimensi dari citra
S adalah citra tersisipi (*stego-image*)
C adalah citra asli (*Cover Image*)

Pada perhitungan *MSE* dan *PSNR* dalam penelitian ini dilakukan penyisipan pesan yang sama yaitu *file* jurnal umum juli 2015.txt.huff (1,82 KB) sedangkan pada citra *cover* dengan ukuran yang berbeda beda yaitu : lebaran.gif (20,4 KB), puasa.gif (17,9 KB), mudik.gif (17,3 KB), love.gif (30,8 KB) di mana semua gambar merupakan format *gif* animasi. Hasil dari pengujian *MSE* dan *PSNR* dapat dilihat pada tabel 2

Tabel 2. *Hasil Pengujian Fidelity*

Citra Cover	File Pesan	Citra Stego	MSE (db)	PSNR(db)
Lebaran.gif (20,4 KB)	Jurnal umum juli.txt	Jurnal1.gif (22,3 KB)	6,34	40,14
Puasa.gif (17,9 KB)	Jurnal umum juli.txt	Jurnal2.gif (19,8 KB)	5,13	41,06
Mudik.gif (17,3 KB)	Jurnal umum juli.txt	Jurnal3.gif (19,1 KB)	5,54	40,73
Love.gif (30,8 KB)	Jurnal umum juli.txt	Jurnal4.gif (32,7 KB)	4,83	41,33

Berdasarkan tabel 2 dapat dilihat nilai rata-rata *MSE* adalah 5,46 *db* dan nilai rata-rata *PSNR* adalah 40,815 hal ini menunjukkan semakin rendah nilai *MSE* maka semakin bagus kualitas citra yang dihasilkan dan mutu *stego-image* dikatakan baik jika nilai *PSNR* 40 db atau lebih.

C. Simulasi dan Pengujian *Recovery*

Pengujian ini dengan melakukan proses *decoding* pada steganografi, sebuah citra harus dapat dipisahkan dari *stego-image*-nya. Pengujian dapat dilakukan dengan melihat keutuhan pesan yang diekstraksi dengan pesan yang asli. Pada tahap pengujian ini akan diambil stego image yang sudah dienkripsi pada file proses sebelumnya, di mana gambar tersebut berisi laporan keuangan yaitu jurnal juli 2015 (Ukuran File 1,82 KB) maka setelah dilakukan proses ekstraksi ukuran file harus utuh beserta isi filenya. Proses pengujian ini dapat dilihat pada gambar 5.

Gambar 5. *Pengujian Recovery*

Pada gambar 5 menunjukkan apabila *stego image* berisi pesan, serta password yang dimasukkan sesuai maka pembacaan steganografi berhasil dan file sudah dapat dipisahkan dari stego image, file hasil dekripsi yaitu jurnal juli 2015.txt.huff (Ukuran File 1,82 KB) antara ukuran file enkripsi dan hasil dekripsi adalah sama.

3. Kesimpulan

Penerapan *gifshuffle* merupakan solusi alternatif yang sesuai untuk penyembunyian pesan teks agar dapat meningkatkan kemananan informasi. Berdasarkan hasil pengujian yang dilakukan dengan metode *imperceptibility* menunjukkan 93% responden berpendapat bahwa stego image tidak berbeda dengan *cover image*, hal ini menunjukkan algoritma *gifshuffle* tidak menurunkan kualitas citra.

Dari pengujian yang sudah dilakukan membuktikan bahwa algoritma *gifshuffle* dapat menampung pesan dengan ukuran yang lebih besar yaitu mencapai 1, 82 MB.

Daftar Pustaka

[1] Alam, Ibnu, "Aplikasi Kode *Huffman* dalam Kompresi Gambar Berformat JPEG", Makalah, 2013, Institut Teknologi Bandung.

[2] Ariyus. Dony, 2006. "Kriptografi Keamanan Data dan Komunikasi". Graha Ilmu, Yogyakarta.

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

[4] Darwis. Dedi "Implementasi Steganografi Pada Berkas Audio Wav Untuk Penyisipan Pesan Gambar Menggunakan Metode Low Bit Coding", Expert, 2015, ISSN 2088-5555, pp. 6-11

[5] Kwan. Matthew, 2010, The *gifshuffle steganography program*. http *://www.darside.com.au/gifshuffle*. 04-04-2017.

[6] Mufida. Khairani, Sajadin. Sembiring, "Analisis dan Implementasi Steganografi Pada Citra GIF Menggunakan Algoritma GifShufle", SNASTIKOM, 2013, ISBN 978-602-19837-3-7.

[7] Penalosa, "Steganografi Pada Citra dengan Format GIF Menggunakan Alogoritma *GifShuffle*", 2005, Teknik Informatika Institut Teknologi Bandung.

[8] Tesla. Yanuar, Pakereng, "Penyembunyian Pesan Terenkripsi pada Citra Gif Menggunakan Algoritma Gifshuffle*"*, Makalah, 2012, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana.