

## **OPTIMALISASI STEGANOGRAFI PADA FILE FLV MEMANFAATKAN METODE INJECTED AT END OF ALL VIDEO TAG DENGAN PENAMBAHAN KOMPRESI**

**Dwi Arraziqi<sup>1</sup> dan F. X. Ferdinandus<sup>2</sup>**

<sup>1</sup>Teknik Informatika STIKOM PGRI Banyuwangi

<sup>2</sup>Teknik Informatika Sekolah Tinggi Teknik Surabaya  
banyuwangi.dwi.arraziqi@gmail.com dan ferdi@stts.edu

### **ABSTRAK**

Proses steganografi pada FLV sering kali mengalami kendala. Kendala ini berupa kapasitas file stego yang besar menyulitkan untuk diembedkan ke dalam file video yang berkapasitas kecil. Kendala yang timbul akibat proses steganografi file stego dapat diminimalisir dengan mengkompresi file stego. Pengkompresian ini dapat dilakukan dengan menggunakan metode kompresi Huffman. Kemudian file stego yang telah terkompresi, didistribusikan pada akhir semua tag video secara merata menggunakan metode Injected at End of All Video Tag. Berdasarkan pengujian terhadap 10 (sepuluh) file stego yang terdiri dari Test.txt, Test.png, Test.jpg, Test.pdf, Test.rtf, Test.xls, Test.zip, Test.rar, Test.exe, Test.doc, terdapat 2 (dua) file stego yang tidak dapat dikompresi secara maksimal dengan menggunakan metode kompresi yang diusulkan yaitu Test.zip dan Test.rar sehingga kapasitasnya lebih besar dari file stego aslinya. Sedangkan untuk file stego lainnya, dapat dikompresi secara maksimal. Ini artinya tingkat keberhasilan dari metode kompresi yang diusulkan mencapai 80% dan mampu mengkompresi hingga 57%.

Kata kunci: *steganografi, flash video, kompresi data*

### **ABSTRACT**

*Steganography in FLV often encounter obstacles. This obstacles in the form of large capacity confidential files can be difficult to be embedded into video file that is small capacity. Constraints arising from the stego files steganography can be minimized by compressing the stego files. This compression can be done by using Huffman compression method. Then the stego files that have been compressed, distributed at the end of all video tag evenly using Injected at End of All Video Tag method. Based on the examination of 10 (ten) stego files consisting of Test.txt, Test.png, Test.jpg, Test.pdf, Test.rtf, Test.xls, Test.zip, Test.rar, Test.exe, Test.doc, there are 2 (two) stego files that can not be compressed to the maximum by using proposed compression method that is Test.zip and Test.rar so this capacity greater than original stego file. As for the other stego files, can be compressed to the maximum. This means that the level of success of proposed compression method reaches 80% and capable of compressing up to 57%.*

*Keywords: steganography, flash video, data compression*

## I. PENDAHULUAN

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. Sebaliknya, kriptografi menyamarkan arti dari suatu pesan, tapi tidak menyembunyikan bahwa ada suatu pesan. Kata “steganografi” berasal dari bahasa Yunani *steganos* yang artinya “tersembunyi atau terselubung” dan *graphein* yang artinya “menulis”.

Di dalam teknik steganografi membutuhkan dua properti yaitu media penampung dan pesan stego. Media penampung (covered media) yang umum digunakan antara lain: citra, video, audio dan teks. Pesan stego yang disembunyikan dapat berupa sebuah artikel, gambar, teks dan lain-lain. Kemudian pesan stego disembunyikan di dalam media lain sehingga pesan stego tidak dapat diketahui oleh pihak yang tidak diberi hak untuk mengetahui bahwa ada pesan tersembunyi di dalamnya. Pesan stego yang disembunyikan dapat diekstrak kembali persis sama seperti aslinya.

Proses steganografi pada FLV sering kali mengalami kendala. Kendala ini berupa kapasitas file stego yang besar menyulitkan untuk diembedkan ke dalam file video yang berukuran kecil. Kendala-kendala ini bisa mengakibatkan proses steganografi tidak berjalan maksimal.

Untuk mengembedkan file stego ke dalam file video diusulkan metode *Injected at End of All Video Tag*, memaksimalkan kapasitas file stego yang diembedkan ke dalam file video diusulkan metode kompresi *Huffman*.

## II. LANDASAN TEORI

### 1. Format File FLV

Format file FLV terdiri dari header pendek, diikuti oleh metadata, dan kemudian bergantian tag audio dan video atau packet seperti pada gambar 1. Header FLV terdiri dari nilai hex tiga yang pertama dari “46 4c 56” yang diterjemahkan menjadi “FLV” dalam nilai string heksadesimal, yang diikuti oleh version, flags, dan offset. Setelah nilai-nilai ini adalah urutan dari tag sampai akhir dari file (EOF). Jenis tag terdiri dari 0x08 untuk AUDIO, 0x09 untuk VIDEO, dan 0x12 untuk META. Setiap tag berisi type, body length, timestamp, timestamp extended, stream id, dan body (data aktual). Setelah setiap tag adalah previous tag size, yang harus selalu sama dengan size dari data aktual dalam byte ditambah 11 byte yang sesuai dengan type tag (1 byte), size body (3 byte), timestamp (3 byte), timestamp extended (1 byte), dan nilai stream id (3 byte) dari tag.

#### FLV Header ¶

Field	Data Type	Example	Description
Signature	byte3	"FLV"	Always "FLV"
Version	uint8	"\x01" (1)	Currently 1 for known FLV files
Flags	uint8 bitmask	"\x05" (5, audio+video)	Bitmask: 4 is audio, 1 is video
Offset	uint32_be	"\x00\x00\x00\x09" (9)	Total size of header (always 9 for known FLV files)

#### FLV Stream

Field	Data Type	Example	Description
PreviousTagSize	uint32_be	"\x00\x00\x00\x00" (0)	Always 0

Then a sequence of tags followed by their size until EOF.

### FLV Tag

Field	Data Type	Example	Description
Type	uint8	"\x12" (0x12, META)	Determines the layout of Body, see below for tag types
BodyLength	uint24_be	"\x00\x00\xe0" (224)	Size of Body (total tag size - 11)
Timestamp	uint24_be	"\x00\x00\x00" (0)	Timestamp of tag (in milliseconds)
TimestampExtended	uint8	"\x00" (0)	Timestamp extension to form a uint32_be. This field has the upper 8 bits.
StreamId	uint24_be	"\x00\x00\x00" (0)	Always 0
Body	byteBodyLength	...	Dependent on the value of Type

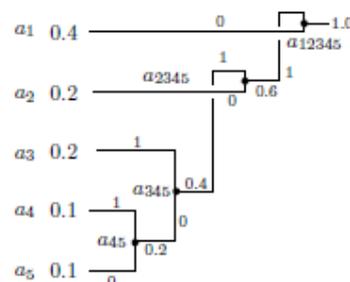
### Previous tag size

Field	Data Type	Example	Description
PreviousTagSize	uint32_be	"\x00\x00\x00\x00" (0)	Total size of previous tag, or 0 for first tag

**Gambar 1. Format File FLV**

## 2. Metode Kompresi Huffman

Algoritma dimulai dengan membangun daftar dari semua simbol alfabet dalam urutan turun dari probabilitasnya. Kemudian dibangun sebuah pohon, dengan simbol pada setiap daunnya, dari bawah ke atas. Hal ini dilakukan dalam langkah-langkah, di mana pada setiap langkah dua simbol dengan probabilitas terkecil dipilih, ditambahkan ke bagian atas dari pohon parsial, dihapus dari daftar, dan digantikan dengan simbol tambahan mewakili dua simbol asli. Ketika daftar direduksi menjadi hanya satu simbol tambahan (mewakili seluruh alfabet), pohon itu lengkap. Pohon kemudian dilalui untuk menentukan kode dari simbol. Proses ini diilustrasikan dengan contoh. Diberikan lima simbol dengan probabilitas seperti ditunjukkan dalam gambar 2.



**Gambar 2. Huffman Code**

## 3. Penelitian Jason Paul Cruz, Nathaniel Joseph Libatique, dan Gregory Tanganon

Metode menyembunyikan file stego pada akhir tag video dan mendistribusikan file stego di antara semua tag video dari carrier (Injected at End of All Video Tag)

direkomendasikan sebagai metode steganografi paling efektif untuk menyembunyikan informasi di dalam carrier FLV karena alasan berikut:

1. Tidak ada distorsi terlihat dapat diamati pada kualitas video dan audio dari FLV stego.
2. File stego disembunyikan di dalam FLV dan bukan pada EOF atau header, sehingga kurang rentan terhadap kecurigaan atau serangan.
3. File stego tidak boleh dirubah bahkan jika FLV stego dikonversi ke format file yang berbeda dan ketika header atau metadata dirubah atau dihapus.
4. Cara untuk memeriksa kecurigaan (yang merupakan evolusi tag video dan histogram) harus disertai dengan software seperti software analisis FLV otomatis yang sulit dilakukan untuk pihak ketiga atau pengamat yang dapat memegang FLV stego.

#### 4. Penelitian Mr. Pravin R. Kamble, Mr. Prakash S. Waghmode, Mr. Vilas S Gaikwad dan Mr. Ganesh B. Hogade

Makalah menyajikan review dari semua metode steganografi yang ada untuk menyembunyikan data di dalam teks, gambar, audio dan video. Beberapa metode steganografi perlu meningkatkan keamanan dengan menggunakan kriptografi terhadap serangan. Teknik steganografi baik gambar dan audio berguna untuk steganografi video untuk meningkatkan kapasitas data aman yang disembunyikan di dalam file carrier. Paralelisasi dari teknik steganografi merupakan tugas penting di lingkup steganografi untuk meningkatkan kecepatan pemrosesan dari algoritma steganografi.

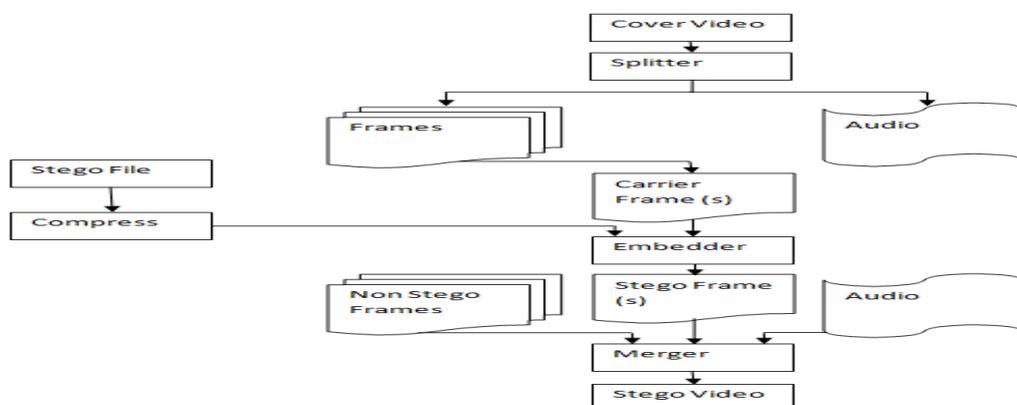
Berbagai teknik steganografi seperti steganografi gambar, audio dan video perlu difokuskan pada kapasitas yang disembunyikan, pendeteksian, tingkat visibilitas dan ketahanan terhadap serangan berbahaya dan disengaja.

#### 5. Penelitian Mridul Kumar Mathur, Seema Loonker dan Dr. Dheeraj Saxena

Metode kompresi Huffman ini sangat cocok untuk gambar bit map grayscale. Untuk gambar grayscale, metode kompresi Huffman ini memberikan penghematan dalam ukuran hingga 60%.

### III. DESAIN SISTEM DAN UJI COBA

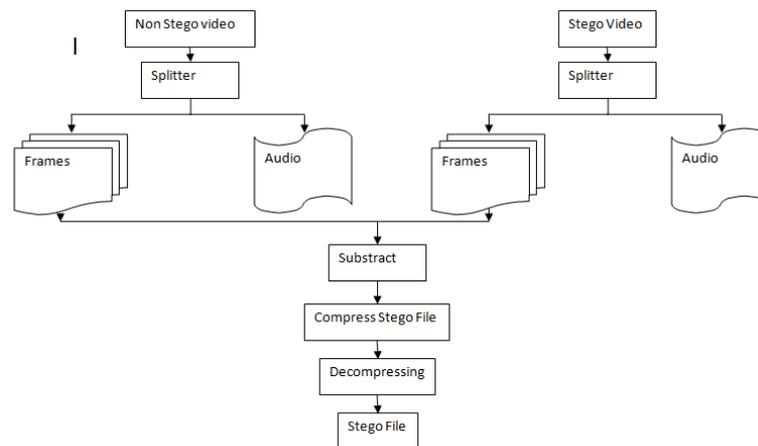
Rancangan proses encode yang digunakan untuk mendapatkan hasil yang sesuai dengan tujuan penelitian ini terlihat pada gambar 3 berikut:



Gambar 3. Blok Diagram Encode

1. Menentukan cover video.
2. Memisahkan antara tag frame (video) dan tag audio.
3. Menentukan file stego yang akan diembedkan.
4. Mengkompres file stego yang akan diembedkan menggunakan metode kompresi *Huffman*.
5. Mengembedkan file stego yang telah terkompresi pada akhir tag video (carrier frame) secara merata (*Injected at End of All Video Tag*).
6. Menggabungkan kembali non stego frame, stego frame (s) dan audio menjadi stego video.

Rancangan proses decode yang digunakan untuk mendapatkan hasil yang sesuai dengan tujuan penelitian ini terlihat pada gambar 4 berikut:



**Gambar 4. Blok Diagram Decode**

1. Menentukan non stego video dan stego video.
2. Memisahkan antara tag frame (video) dan tag audio dari non stego video dan stego video.
3. Kurangkan antara tag frame (video) dari stego video dengan tag frame (video) dari non stego video menjadi file stego yang terkompresi.
4. File stego yang terkompresi, didekompresi menggunakan metode *Huffman* menjadi file stego yang asli.

Tabel 1 merupakan hasil pengujian terhadap 10 (sepuluh) file stego yang terdiri dari Test.txt, Test.png, Test.jpg, Test.pdf, Test.rtf, Test.xls, Test.zip, Test.rar, Test.exe, Test.doc terhadap file FLV yang memiliki kapasitas 654 KB, tag audio 649 dan tag video 424.

**Tabel 1. Uji Coba**

File FLV	File Stego Uji Coba	Kompresi Huffman	Hasil FLV Stego
1.FLV 654 KB A : 649 V : 424	Test.txt 16780 B	10083 B	Tidak Dicurigai
	Test.png 885 B	866 B	Tidak Dicurigai

Test.jpg 28996 B	28917 B	Tidak Dicurigai
Test.pdf 42517 B	40056 B	Tidak Dicurigai
Test.rtf 60814 B	36919 B	Tidak Dicurigai
Test.xls 70656 B	40183 B	Tidak Dicurigai
Test.zip 78865 B	78866 B	Tidak Dicurigai
Test.rar 445673 B	445674 B	Tidak Dicurigai
Test.exe 115516 B	1155090 B	Tidak Dicurigai
Test.doc 45056 B	23621 B	Tidak Dicurigai

#### IV. PENUTUP

Kesimpulan yang diperoleh dari penelitian yang dilakukan adalah sebagai berikut:

1. Metode kompresi *Huffman* yang diusulkan mampu mengkompres kapasitas file stego yang diembedkan hingga 57%.
2. Metode steganografi *Injected at End of All Video Tag* yang diusulkan mampu tidak ada distorsi terlihat dapat diamati pada kualitas video dan audio dari FLV stego.
3. Berdasarkan pengujian terhadap 10 (sepuluh) file stego yang terdiri dari Test.txt, Test.png, Test.jpg, Test.pdf, Test.rtf, Test.xls, Test.zip, Test.rar, Test.exe, Test.doc, terdapat 2 (dua) file stego yang tidak dapat dikompresi secara maksimal dengan menggunakan metode kompresi yang diusulkan yaitu Test.zip dan Test.rar sehingga kapasitasnya lebih besar dari file stego aslinya. Sedangkan untuk file stego lainnya, metode kompresi yang diusulkan mampu mengkompresi file stego sehingga kapasitasnya lebih kecil dari file stego aslinya. Ini artinya tingkat keberhasilan dari metode kompresi yang diusulkan mencapai 80%.

#### V. DAFTAR PUSTAKA

- [1] Cay S. Horstmann, Gary Cornell, *Core Java™ 2 Volume I - Fundamentals*, 7<sup>th</sup> Edition, Prentice Hall PTR, 2004.
- [2] David Salomon, *Data Compression The Complete Reference*, 4<sup>th</sup> Edition, Springer, 2007.
- [3] Jason Paul Cruz, Nathaniel Joseph Libatique, Gregory Tangonan, *Steganography and Data Hiding in Flash Video (FLV)*, TENCON, 2012.
- [4] Mridul Kumar Mathur, Seema Loonker, Dr. Dheeraj Saxena, *LOSSLESS HUFFMAN CODING TECHNIQUE FOR IMAGE COMPRESSION AND RECONSTRUCTION USING BINARY TREES*, IJCTA, 2012.
- [5] Mr. Pravin R. Kamble, Mr. Prakash S. Waghmode, Mr. Vilas S Gaikwad, Mr. Ganesh B. Hogade, *Steganography Techniques: A Review*, IJERT, 2013.