

# IMPLEMENTASI STEGANOGRAFI MENYISIPKAN PESAN GAMBAR BERFORMAT BITMAP PADA BERKAS AUDIO BERFORMAT WAV DENGAN METODE END OF FILE

Akhmad Iskhak <sup>1)</sup>, Iman Sapuguh <sup>2)</sup>

Program Studi Teknik Informatika Universitas 45 Surabaya,  
email : [sapuguh@yahoo.com](mailto:sapuguh@yahoo.com)

## Abstrak

Berbagai macam teknik keamanan telah dikembangkan untuk menjaga kerahasiaan dan keamanan informasi agar terhindar dari orang yang tidak berhak, salah satunya yaitu dengan teknik steganografi. Implementasi steganografi saat ini telah mengalami perkembangan yang sangat signifikan, dimana menggunakan media digital sebagai media penampung atau menyembunyi pesan, salah satunya adalah media digital audio.

Implementasi steganografi untuk menyisipkan berkas gambar berformat bitmap pada berkas audio berformat wav dengan metode end of file menggunakan metodologi studi pustaka, analisis dan perancangan sistem, implementasi sistem, pengujian sistem dan dokumentasi. Perancangan aplikasi steganografi dibuat mulai blok diagram, flowchart, context diagram, bagan berjenjang, DFD dan desain interface. Sedangkan pembuatan aplikasi ini menggunakan bahasa pemrograman visual basic.net

Pembuatan aplikasi steganografi menghasilkan form encoding dan form decoding yang digunakan untuk menyisipkan pesan rahasia berformat bitmap pada berkas audio berformat wav dan untuk ekstraksi pesan rahasia. Dengan menggunakan metode end of file berkas audio sebagai berkas penampung tidak mengalami perubahan kualitas audio.

*Keywords: Steganografi, encoding, decoding*

## 1. PENDAHULUAN

Sejalan dengan pesatnya perkembangan teknologi informasi dan komunikasi memberikan pengaruh yang besar bagi kehidupan manusia. Perkembangan teknologi informasi dan komunikasi memungkinkan setiap orang untuk saling bertukar informasi, data, atau pesan kepada orang lain tanpa batasan jarak dan waktu.

Faktor kerahasiaan dan keamanan merupakan aspek yang sangat penting dalam proses pertukaran informasi melalui internet. Hal ini diikuti pula dengan berkembangnya kejahatan teknologi informasi dan komunikasi dengan berbagai teknik, diantaranya adalah penyadapan, modifikasi, maupun perusakan informasi yang berada di dunia maya.

Berbagai macam teknik keamanan telah dikembangkan untuk menjaga kerahasiaan dan keamanan informasi agar terhindar dari orang yang tidak berhak, salah satunya yaitu dengan teknik steganografi. Steganografi merupakan seni menyembunyikan pesan dalam data lain tanpa mengubah data yang ditumpanginya tersebut, sehingga data yang ditumpanginya sebelum dan setelah proses penyembunyian hampir terlihat sama, sehingga keberadaan pesan tersebut tidak dapat diketahui oleh orang yang tidak berhak. Sebagai contoh proses pengiriman pesan melalui email, apabila pesan tersebut hanya ingin diketahui oleh penerima yang berhak

saja, maka pesan tersebut disisipkan ke dalam sebuah media lain, misalnya dalam media berkas audio. Sehingga orang lain yang tidak memiliki hak, tidak akan mengetahui pesan apa yang disisipkan pada berkas audio yang dikirimkan.

Implementasi steganografi saat ini telah mengalami perkembangan yang sangat signifikan. Dimana menggunakan media digital sebagai media penampung atau menyembunyi pesan, salah satunya adalah media digital audio. Steganografi menyisipkan atau menyembunyikan berkas berupa gambar di dalam sebuah berkas audio, agar pihak lain yang tidak berhak tidak menyadari keberadaan pesan yang ada di dalam berkas audio tersebut.

Tujuan mengimplementasikan steganografi dengan metode *end of file* (EOF) menyisipkan berkas gambar berformat bitmap pada berkas audio berformat wav untuk menghasilkan sebuah program aplikasi.

Adapun manfaat dari program aplikasi steganografi yang dibuat adalah memberikan keamanan dan fleksibilitas pada berkas pesan rahasia, dan dengan metode *end of file* ini berkas yang disisipkan dimungkinkan untuk lebih besar dari berkas penampung.

## 2. STEGANOGRAFI

Menurut pengertian dan definisi istilah steganografi adalah paduan antara seni dan ilmu pengetahuan yang mempelajari cara menuliskan pesan tersembunyi. Isi maupun arti pesan tersebut hanya bisa diterjemahkan oleh pembuat dan orang yang berhak menerima pesan (Jubilee Enterprise, 2010:139). Steganografi berasal dari kata Yunani *Stagonos*, yang berarti tertutup atau tercover, dan *Grafi* yang artinya tulisan. Steganografi adalah seni atau ilmu untuk menyamarkan sebuah pesan/ data rahasia di dalam data atau media yang tampaknya biasa saja, sehingga keberadaan pesan rahasia itu sulit diketahui (Arryawan, 2010:104).

Penyembunyian pesan harus memenuhi beberapa kriteria diantaranya adalah : (Rinaldi Munir, 2004 : 209)

a. *Fidelity*

Mutu pesan penampung tidak jauh berubah. Setelah penambahan data rahasia, pesan hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam pesan tersebut terdapat data rahasia.

b. *Robustness*

Data yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada citra penampung, seperti perubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, pemotongan (*cropping*), enkripsi, dan sebagainya. Bila pada citra dilakukan operasi-operasi pengolahan citra tersebut, maka data yang disembunyikan seharusnya tidak rusak (tetap valid jika diekstraksi kembali).

c. *Recovery*

Data yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

Terdapat beberapa istilah yang berkaitan dengan steganografi. (Rinaldi Munir, 2006 : 304), diantaranya adalah :

- a. *Hiddentext* atau *embedded message* yaitu pesan atau informasi yang disembunyikan.
- b. *Coverttext* atau *cover-object* yaitu pesan yang digunakan untuk menyembunyikan *embedded message*.
- c. *Stegotext* atau *stego-object* yaitu : pesan yang sudah berisi *embedded message*. Dalam steganografi digital, baik *hiddentext* atau *coverttext* dapat berupa teks, audio, gambar, maupun video.

## 2.1 Metode Steganografi

Implementasi steganografi ada beberapa metode yang digunakan, diantaranya adalah sebagai berikut:

### a) LSB (Least Significant Bit)

Metode ini bekerja dengan cara mengganti bit terakhir dari masing-masing piksel dengan pesan yang akan disisipkan. LSB mempunyai kelebihan yakni ukuran gambar tidak akan berubah. Sedangkan kekurangannya adalah pesan yang akan disisipkan terbatas, sesuai dengan ukuran citra. Selain itu, kualitas citra akan sedikit berubah setelah mengalami proses penyisipan pesan. Namun perubahan yang sedikit itu biasanya tidak nampak oleh mata. (Edisuryana, 2013: 736)

Cara yang dilakukan untuk mengganti binery *cover object* dengan melakukan operasi penambahan atau pengurangan nilai binery pada *cover object* dengan binery pesan rahasia. Misalnya struktur binery dari *cover object* berkas berformat WAV adalah sebagai berikut :

```
01001101 00101110 10101110 10001010 10101111
10100010 00101011 10101011
```

Digunakan untuk menyimpan karakter “H” dengan struktur binery (01001000), maka struktur binery *cover object* tersebut akan diubah menjadi :

```
01001100 00101111 10101110 10001010 10101111
10100010 00101010 10101010
```

### b) EOF (End Of File)

Secara umum metode steganografi menggunakan *redundant bits* sebagai tempat menyembunyikan pesan pada saat dilakukan kompresi data, dan kemudian menggunakan kelemahan indera manusia yang tidak sensitif sehingga pesan tersebut tidak ada perbedaan yang terlihat. Metode EOF atau *End Of File* merupakan salah satu metode yang digunakan dalam steganografi. Metode ini digunakan dengan cara menambahkan data atau pesan rahasia pada akhir file. Perhitungan ukuran file yang telah disisipkan data sama dengan ukuran file sebelum disisipkan data ditambah ukuran data rahasia yang telah diubah menjadi *encoding file*. (Martono, 2013:229)

Cara yang dilakukan untuk mengganti binery *cover object* dengan melakukan operasi penambahan atau pengurangan nilai binery pada *cover object* dengan binery pesan rahasia. Misalnya struktur binery dari *cover object* berkas berformat WAV adalah sebagai berikut :

```
01001101 00101110 10101110 10001010 10101111
10100010 00101011 10101011
```

Digunakan untuk menyimpan karakter “H” dengan struktur binery (01001000), maka struktur binery *cover object* tersebut akan ditambahkan dengan struktur binery karakter “H” pada bagian akhir. Berikut adalah hasil penjumlahannya :

```
01001101 00101110 10101110 10001010 10101111
10100010 00101011 10101011 01001000
```

## 2.2 Format Gambar Digital

Gambar digital merupakan dokumen berbentuk file yang dihasilkan melalui perangkat elektronik atau media digital (Ahmad, 2005:106). Sebuah gambar digital terdiri atas piksel-piksel. Piksel merupakan akronim dari Picture Element. Setiap piksel mengandung informasi mengenai warna piksel tersebut. Warnanya bisa dalam susunan Merah, Hijau, Biru (*Red, Green, Blue* atau RGB).

Ada beberapa tipe gambar digital yang sering digunakan yaitu:

- a. Bitmap (BMP)  
Tipe file BMP umum digunakan pada sistem operasi Windows dan OS/2.
- b. Joint Photographic Experts Group (JPG/JPEG)  
Tipe file JPG sangat sering digunakan untuk web atau blog. File JPG menggunakan teknik kompresi yang menyebabkan kualitas gambar turun (*lossy compression*).
- c. Graphics Interchange Format (GIF)  
Tipe file GIF memungkinkan penambahan warna transparan dan dapat digunakan untuk membuat animasi sederhana, tetapi saat ini standar GIF hanya maksimal 256.
- d. Portable Network Graphics (PNG)  
Tipe file PNG merupakan solusi kompresi yang kuat dengan warna yang lebih banyak (24 bit RGB + alpha).

## 2.3 Format Berkas Audio

Menurut Iwan Binanto (2010 : 58-59), terdapat beberapa jenis format yang umum untuk suara diantaranya :

- a. WAV  
WAV merupakan standar format file yang digunakan oleh Windows. WAV umumnya digunakan dalam menyimpan audio tak terkompresi untuk menyimpan file suara berkualitas. File .wav juga dapat berisi data terkodekan dengan beraneka ragam codec untuk mengurangi ukuran file.
- b. OGG  
OGG adalah format file yang bebas dan open source yang mendukung beraneka ragam codec.
- c. MPC  
MPC (dikenal juga sebagai MPEGplus, MPEG+, atau MP+) merupakan codec audio yang open source dan lossy. MPC dan OGG merupakan codec terbaik untuk file yang terkompresi dan berkualitas tinggi untuk audio yang lossy.
- d. AIFF  
Format .aiff merupakan standar format file audio yang digunakan oleh Apple.
- e. AAC  
AAC adalah kependekan dari Advanced Audio Coding yang berbasis pada standar MPEG2 dan MPEG4.

f. MP3

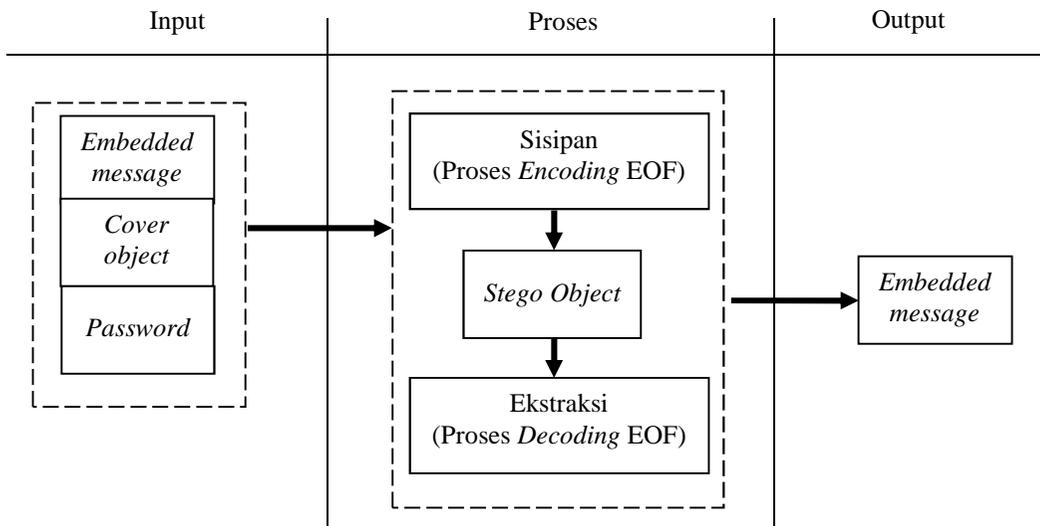
Format MPEG Layer-3 merupakan format yang sangat populer untuk pengunduhan dan penyimpanan musik.

### 3. PERANCANGAN SISTEM

Perancangan sistem yang baik memerlukan beberapa tahapan-tahapan untuk menghasilkan rancangan sistem yang mudah untuk diimplementasikan, diantaranya adalah :

#### 3.1 Block Diagram

*Block diagram* merupakan sebuah gambaran dari sistem, atau program yang masing-masing fungsinya diwakili oleh gambar kotak berlabel dan hubungan diantaranya digambarkan dengan garis penghubung. Berikut adalah *block diagram* dari steganografi :



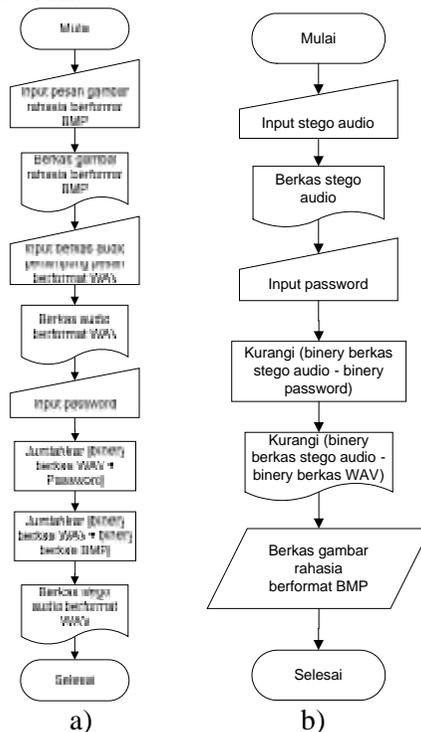
Gambar 1. Blok Diagram Sistem

#### 3.2 Flowchart

Flowchart yang ada dibagi menjadi dua, yaitu flowchart proses encoding dan flowchart decoding. *Encoding* merupakan penyisipan pesan gambar rahasia ke dalam berkas audio, dimana dalam proses penyisipan disertakan juga *password* untuk keamanan dari pihak yang tidak berhak ketika ingin melakukan ekstraksi. Berikut ini adalah *flowchart* yang menggambarkan alur proses penyisipan pesan rahasia yang disisipkan pada berkas audio WAV.

*Decoding* merupakan proses ekstraksi pesan rahasia yang ada pada stego audio. Proses *decoding* diawali menginputkan stego audio dan *password* untuk ekstraksi, kemudian diproses dengan mengurangi binary berkas stego audio dengan binary berkas WAV, sehingga hasil pengurangan tersebut menghasilkan gambar rahasia berformat BMP.

Gambar 2 dibawah ini menunjukkan alur proses encoding dan decoding steganografi menyisipkan pesan gambar berformat bitmap pada berkas audio berformat wav dengan metode end of file.



Gambar 2 a) flowchart proses encoding b) flowchart proses decoding

#### 4. HASIL DAN PEMBAHASAN

Pengujian aplikasi steganografi ini dicoba dengan berbagai ukuran kapasitas berkas yang berbeda-beda. Berikut adalah detail dari pengujian yang telah dilakukan:

##### a) Berkas Yang Akan Diuji

Berikut ini adalah tabel berkas yang digunakan dalam pengujian :

Tabel 1 Berkas yang akan diuji

No	Nama Berkas	Ukuran (MB)	Keterangan
1	Ebiet G. Ade - Berita Kepada Kawan.wav	29.2	Cover Object
2	Ebiet - Elegi Esok Pagi.wav	20.4	Cover Object
3	Ebiet-Rindu Yang Terlarang.wav	23.6	Cover Object
4	Ijazah.bmp	60.8	Embedded Message
5	Surat Keterangan.bmp	45.5	Embedded Message
6	Naskah Soal Fiqih Kelas X.bmp	50.2	Embedded Message

Berdasarkan data Tabel 4.1 dapat dikelompokan :

1. Berkas yang digunakan sebagai *cover object* terdiri dari 3 berkas dengan ukuran 29.2 MB, 20.4 MB dan 10.0 MB.
2. Berkas yang digunakan sebagai *embedded message* lebih besar dari *cover object*.

**b) Hasil Pengujian**

Hasil Pengujian berikut menjelaskan tingkat keberhasilan dari uji coba yang sudah dilakukan. Tingkat keberhasilan setiap berkas telah di uji coba beberapa kali pengujian. Hasil pengujian dari aplikasi steganografi adalah seperti tampak pada Tabel 2 berikut ini:

Tabel 2 Hasil pengujian

No	Cover Object	Embedded Message	Uji Coba	Keterangan	Persentase
1	Ebiet G. Ade - Berita Kepada Kawan.wav	Ijazah.bmp	5 kali	5 kali sukses	100%
2	Ebiet - Elegi Esok Pagi.wav	Surat Keterangan.bmp	5 kali	5 kali sukses	100%
3	Ebiet -Titip_Rindu Buat Ayah.wav	Naskah Soal Fiqih Kelas X.bmp	5 kali	5 kali sukses	100%

Hasil pengujian pada Tabel 4.2 diperoleh hasil pengujian 100% sukses dari masing-masing berkas dengan beberapa kali uji coba.

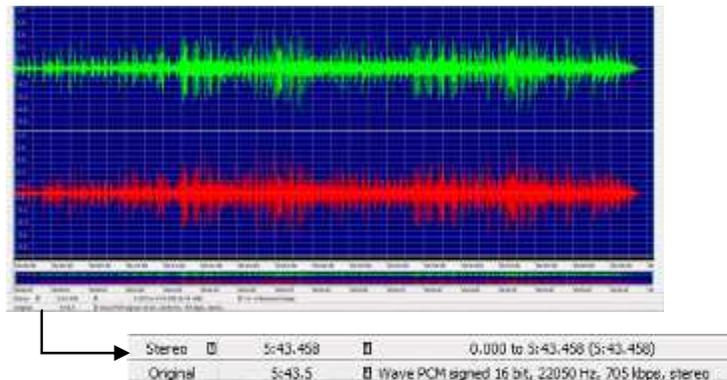
**c) Analisis Hasil Pengujian**

Analisis dalam pengujian ini menggunakan aplikasi Goldwave untuk mengetahui detail perubahan dari kualitas suara, dan aplikasi Notepad++ untuk mengetahui detail perubahan pada struktur *binary* pada *cover object* sebelum disisipi pesan, maupun setelah disisipi pesan.

Berikut adalah analisis proses *encoding* dan *decoding* :

1. Penjelasan Proses *Encoding* Percobaan Ke-1

Proses *encoding* merupakan proses penyisipan *embedded message* ke dalam *cover object*. Berikut ini adalah tampilan detail kualitas suara *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav sebelum disisipi pesan :



Gambar 3 Detail kualitas suara berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav sebelum disisipi pesan.

Detail informasi dari Gambar 3 menjelaskan bahwa berkas *cover object* sebelum disisipi pesan memiliki durasi 5 menit 43,458 detik dengan bit rate 22050 Hz, 705 kbps. Berikut ini adalah tampilan struktur *binary* dari berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav sebelum disisipi pesan :

Address	0	1	2	3	4	5	6	7	8
00000000	01010010	01001001	01000110	01000110	10111000	00111100	11001110	00000001	01010111
00000010	00010000	00000000	00000000	00000000	00000001	00000000	00000010	00000000	00100010
00000020	00000100	00000000	00010000	00000000	01001100	01001001	01010011	01010100	10001100
00000030	01001001	01000001	01010010	01010100	00001101	00000000	00000000	00000000	01000101
00000040	00100000	01000001	01100100	01100101	00000000	00000000	01001001	01000011	01010010
00000050	00110111	00111001	00000000	00000000	01001001	01000111	01001110	01010010	00001100
00000060	00100000	01000010	01100001	01101100	01101100	01100001	01100100	00000000	01001001
00000070	01000010	01100101	01110010	01101001	01110100	01100001	00100000	01001011	01100101
00000080	01110111	01100001	01101110	00000000	01001001	01010000	01010010	01000100	00001100

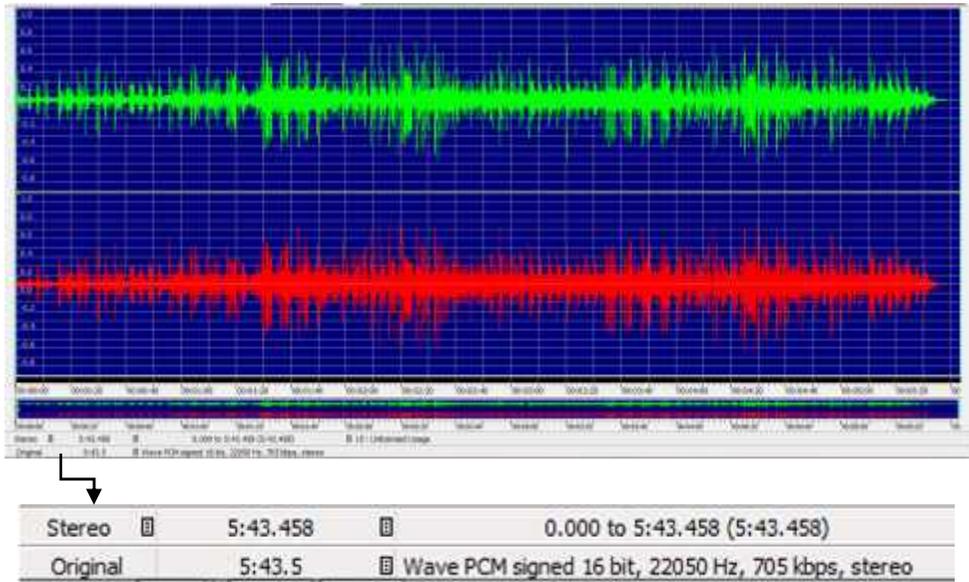
Gambar 4 Detail *binary* berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav sebelum disisipi pesan.

Berikut ini adalah tampilan struktur *binary* dari berkas *embedded message* Ijazah.bmp sebelum disisipan pada *cover object* :

Address	0	1	2	3	4	5	6	7	8
00000000	01111010	11111100	00100001	11111100	00010101	11111011	00001010	11111011	10010110
00000010	11100011	11111000	00110001	11111000	00101010	11111001	11110011	11110110	01101011
00000020	01100011	11111000	00110100	11110110	00100010	11111011	11011011	11110110	10101111
00000030	01101011	11111101	00100001	11111001	10011001	11111100	10111111	11111000	11111001
00000040	10010000	11111100	10110000	11111000	11110101	11111010	00001110	11111000	00010000
00000050	00000010	11111010	11110010	11110111	00000101	11111011	10111111	11110111	10111011
00000060	01111010	11111001	01001111	11111000	01111000	11111001	00110000	11111000	10111111
00000070	00001001	11111010	01011110	11111000	11001111	11111010	01101001	11111001	11110111
00000080	01101110	11111001	01101101	11111011	11110101	11110111	00101100	11111001	00101000
00000090	01111101	11111010	11010111	11111000	00001000	11111001	11110101	11110111	10010010
000000a0	00000000	11111010	10000001	11111000	00000010	11111001	10111011	11111000	00101000
000000b0	11001001	11111011	01001000	11111001	00001110	11111011	10000010	11111000	10100111
000000c0	11010100	11111011	00001000	11111010	10101010	11111111	01010011	11111011	10100101
000000d0	10111011	11111110	01011101	11111100	11011011	11111110	11011000	11111100	00111010
000000e0	10110100	00000010	00101011	11111111	10101011	00000010	00001110	00000000	01000001
000000f0	11011111	00000010	11101110	00000010	00101011	00000101	00101010	00000011	00101111
00000100	11111001	00000101	11011100	00000100	10101001	00000101	11010100	00000101	01100011
00000110	00000101	00001000	10101001	00000110	11010000	00001000	11010010	00000111	01111110
00000120	01101100	00000101	10100111	00000111	00000000	00000111	11110010	00000111	11010110
00000130	01011000	00000101	01111110	00000110	10001011	00000110	01101100	00000110	10100110

Gambar 5 Detail *binary* berkas *embedded message* Ijazah.bmp sebelum disisipkan pada *cover object*.

Berikut ini adalah tampilan kualitas audio berkas *cover object* setelah Ebiet G. Ade – Berita Kepada Kawan.wav setelah disisipi *embedded message* :



Gambar 6 Detail kualitas suara berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah disisipi *embedded message*.

Detail informasi dari Gambar 6 menjelaskan bahwa berkas *cover object* setelah proses *encoding* memiliki durasi 5 menit 43,458 detik dengan bit rate 22050 Hz, 705 kbps yang berarti kualitas suara berkas tersebut tidak mengalami perubahan.

Berikut ini adalah tampilan struktur *binary* dari berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah disisipi pesan :

Address	0	1	2	3	4	5	6	7	8
00000000	01010010	01000100	01000110	01000110	10111000	00111100	11001110	00000001	01010111
00000010	00010000	00000000	00000000	00000000	00000001	00000000	00000010	00000000	00100010
00000020	00000100	00000000	00010000	00000000	01001100	01001001	01010011	01010100	10001100
00000030	01001001	01000001	01010010	01010100	00001101	00000000	00000000	00000000	01000101
00000040	00100000	01000001	01100100	01100101	00000000	00000000	01001001	01000011	01010010
00000050	00110111	00111001	00000000	00000000	01001001	01000111	01001110	01010010	00001100
00000060	00100000	01000010	01100001	01101100	01101100	01100001	01100100	00000000	01001001
00000070	01000010	01100101	01110010	01101001	01110100	01100001	00100000	01001011	01100101
00000080	01110111	01100001	01101110	00000000	01001001	01010000	01010010	01000100	00001100
00000090	01111010	11111100	00100001	11111100	00010101	11111011	00001010	11111011	10001010
000000a0	11100011	11111000	00110001	11111000	00101010	11111001	11110011	11110110	01101011
000000b0	01100011	11111000	00110100	11110110	00100010	11111011	11011011	11110110	10101111
000000c0	01101011	11111101	00100001	11111001	10011001	11111100	10111111	11111000	11111001
000000d0	10010000	11111100	10110000	11111000	11110101	11111010	00001110	11111000	00010000
000000e0	00000010	11111010	11110010	11110111	00000101	11111011	10111111	11110111	10111011
000000f0	01111010	11111001	01001111	11111000	01111000	11111001	00110000	11111000	10111111
00000100	00001001	11111010	01011110	11111000	11001111	11111010	01101001	11111001	11110111
00000110	01101110	11111001	01101101	11111011	11110101	11110111	00101100	11111001	00101000
00000120	01111101	11111010	11010111	11111000	00001000	11111001	11110101	11110111	10010010
00000130	00000000	11111010	10000001	11111000	00000010	11111001	10111011	11111000	00101000
00000140	11001001	11111011	01001000	11111001	00001110	11111011	10000010	11111000	10100011
00000150	11010100	11111011	00001000	11111010	10101010	11111111	01010011	11111011	10100101
00000160	10111011	11111110	01011101	11111100	11011011	11111110	11011000	11111100	00111010
00000170	10110100	00000010	00101011	11111111	10101011	00000010	00001110	00000000	01000001
00000180	11011111	00000010	11101110	00000010	00101011	00000101	00101010	00000011	00101111
00000190	11111001	00000101	11011100	00000100	10101001	00000101	11010100	00000101	01100011
000001a0	00000101	00001000	10101001	00000110	11010000	00001000	11010010	00000111	01111110
000001b0	01101100	00000101	10100111	00000111	00000000	00000111	11110010	00000111	11010110
000001c0	01101000	00000101	01111110	00000110	10001011	00000110	01101100	00000110	10100110

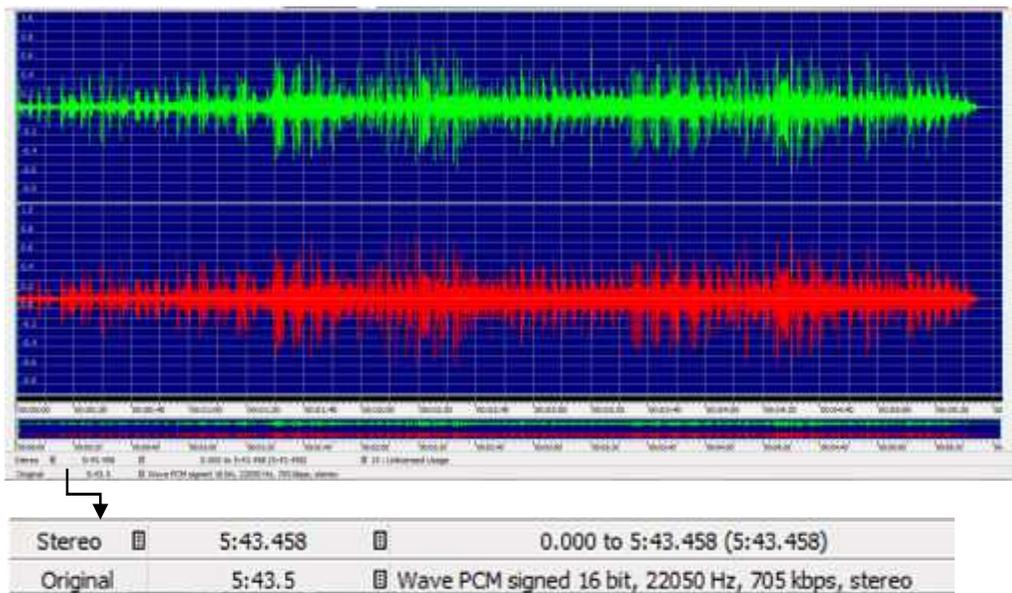
Gambar 7 Detail *binary* berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah disisipi *embedden message*.

Detail informasi dari Gambar 7 menjelaskan bahwa berkas *cover object* setelah disisipi *embedded message* mengalami perubahan struktur *binary*. Jumlah struktur *binary* bertambah karena ada pesan yang telah disisipkan. Penambahan struktur *binary* pada *cover object* terletak pada akhir *address* sesuai dengan metode yang digunakan yaitu *end of file*, dimana proses penyisipan *binary embedded message* diletakkan pada akhir *address cover object*.

## 2. Penjelasan Proses *Decoding* Percobaan Ke-1

Proses *decoding* merupakan proses ekstraksi *embedded message* dari *cover object*, yaitu memisahkan kembali struktur *binary* antara *cover object* dengan *binary embedded message* dengan cara pengambilan kembali struktur *binary embedded message* yang telah disisipkan pada struktur *binary cover object*.

Berikut ini adalah detail tampilan kualitas suara dari berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah proses *decoding* :



Gambar 8 Detail kualitas suara berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah proses *decoding*.

Detail informasi dari Gambar 8 menjelaskan bahwa berkas *cover object* setelah proses *decoding* memiliki durasi 5 menit 43,458 detik dengan bit rate 22050 Hz, 705 kbps yang berarti kualitas suara berkas tersebut tidak mengalami perubahan.

Berikut ini adalah tampilan struktur *binary* dari berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah proses *decoding* :

Address	0	1	2	3	4	5	6	7	8
00000000	01010010	01001001	01000110	01000110	10111000	00111100	11001110	00000001	01010111
00000010	00010000	00000000	00000000	00000000	00000001	00000000	00000010	00000000	00100010
00000020	00000100	00000000	00010000	00000000	01001100	01001001	01010011	01010100	10001100
00000030	01001001	01000001	01010010	01010100	00001101	00000000	00000000	00000000	01000101
00000040	00100000	01000001	01100100	01100101	00000000	00000000	01001001	01000011	01010010
00000050	00110111	00111001	00000000	00000000	01001001	01000111	01001110	01010010	00001100
00000060	00100000	01000010	01100001	01101100	01101100	01100001	01100100	00000000	01001001
00000070	01000010	01100101	01110010	01101001	01110100	01100001	00100000	01001011	01100101
00000080	01110111	01100001	01101110	00000000	01001001	01010000	01010010	01000100	00001100

Gambar 9 Detail *binary* berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah proses *decoding*

Detail informasi dari Gambar 9 menjelaskan bahwa struktur *binary* dari berkas *cover object* Ebiet G. Ade – Berita Kepada Kawan.wav setelah proses *decoding* kembali seperti semula.

Berikut ini adalah tampilan struktur *binary* dari berkas *embedded message* Ijazah.bmp setelah proses *decoding* :

Address	0	1	2	3	4	5	6	7	8
00000000	01111010	11111100	00100001	11111100	00010101	11111011	00001010	11111011	10010110
00000010	11100011	11111000	00110001	11111000	00101010	11111001	11110011	11110110	01101011
00000020	01100011	11111000	00110100	11110110	00100010	11111011	11011011	11110110	10101111
00000030	01101011	11111101	00100001	11111001	10011001	11111100	10111111	11111000	11111001
00000040	10010000	11111100	10110000	11111000	11110101	11111010	00001110	11111000	00010000
00000050	00000010	11111010	11110010	11110111	00000101	11111011	10111111	11110111	10111011
00000060	01111010	11111001	01001111	11111000	01111000	11111001	00110000	11111000	10111111
00000070	00001001	11111010	01011110	11111000	11001111	11111010	01101001	11111001	11110111
00000080	01101110	11111001	01101101	11111011	11110101	11110111	00101100	11111001	00101000
00000090	01111101	11111010	11010111	11111000	00001000	11111001	11110101	11110111	10010010
000000a0	00000000	11111010	10000001	11111000	00000010	11111001	10111011	11111000	00101000
000000b0	11001001	11111011	01001000	11111001	00001110	11111011	10000010	11111000	10100111
000000c0	11010100	11111011	00001000	11111010	10101010	11111111	01010011	11111011	10100101
000000d0	10111011	11111110	01011101	11111100	11011011	11111110	11011000	11111100	00111010
000000e0	10110100	00000010	00101011	11111111	10101011	00000010	00001110	00000000	01000001
000000f0	11011111	00000010	11101110	00000010	00101011	00000101	00101010	00000011	00101111
00000100	11111001	00000101	11011100	00000100	10101001	00000101	11010100	00000101	01100011
00000110	00000101	00001000	10101001	00000110	11010000	00001000	11010010	00000111	01111110
00000120	01101100	00000101	10100111	00000111	00000000	00000111	11110010	00000111	11010110
00000130	01011000	00000101	01111110	00000110	10001011	00000110	01101100	00000110	10100110

Gambar 10 Detail *binary* berkas *embedded message* Ijazah.bmp setelah proses *decoding*.

Detail informasi dari Gambar 10 menjelaskan bahwa struktur *binary* dari berkas *embedded message* Ijazah.bmp setelah proses *decoding* memiliki struktur *binary* yang sama seperti sebelum disisipkan ke dalam berkas *cover object*.

## 5. SIMPULAN

Berdasarkan hasil uji coba dan analisis aplikasi steganografi dengan metode *end of file*, dapat disimpulkan beberapa simpulan diantaranya :

1. Berkas yang menjadi *embedded message* bisa memiliki ukuran yang lebih besar dari pada berkas *cover object*,
2. Kualitas audio dari berkas *cover object* tidak mengalami penurunan.

## DAFTAR PUSTAKA

- Arryawan, Eko. 2010. *Anti Forensik Mengatasi Investigasi Komputer Forensik*. Jakarta : Elex Media Komputindo
- Binanto, Iwan. 2010. *Multimedia Digital – Dasar Teori dan Pengembangannya*. Yogyakarta : Andi
- Edisuryana, Mukharrom. 2013. “Aplikasi Steganografi Pada Citra Berformat Bitmap Dengan Menggunakan Metode End Of File”. *Jurnal Ilmiah Teknik Elektro Undip Semarang*.2(3), 735
- Enterprise, Jubilee. 2010. *13 Ancaman PC dan Cara Mengatasinya*. Jakarta : Elex Media Komputindo
- Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi Pendekatan Terstruktur*. Yogyakarta : Andi
- Martono, Irawan. 2013. “Penggunaan Steganografi dengan Metode End of File (EoF) pada Digital Watermarking”. *Jurnal Teknologi Of Informasi And Communication Aptikom*. 2(1), 229
- Munir, Rinaldi. 2006. *Kriptografi*. Bandung : Informatika
- Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung : Informatika
- Purba Verlando, Jhoni, “Implementasi Steganografi Pesan Text Ke Dalam File Sound (.WAV) Dengan Modifikasi Jarak Byte Pada Algoritma Least Significant Bit (LSB). *Jurnal Dunia Teknologi Informasi USU*. 1(1), 51
- Raymond McLeod, Jr dan P. Schell, George. 2008. *Sistem Informasi Manajemen*. Jakarta : Salemba Empat
- Usman, Ahmad. 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta : Graha Ilmu