PENYEMBUNYIAN PESAN TEXT PADA FILE WAV DENGAN METODE LEAST SIGNIFICANT BIT

Andy Gunawan [1], Nugroho Agus Haryono[2], Junius Karel T[3]

Abstrak: Pertukaran informasi pada saat ini sangat mudah untuk dilakukan. Bentuk informasi digital yang dapat ditukar berupa data text, gambar, gambar bergerak dan suara. Seiring dengan perkembangan tersebut, tingkat keamanan atau kerahasiaan data juga semakin lemah. Orang dapat dengan mudahnya melakukan "pencurian" data, untuk itu diperlukan teknik untuk menyembunyikan data, dengan harapan agar data yang sifatnya rahasia tidak diketahui oleh orang yang tidak berkepentingan. Pada kasus ini data yang akan disembunyikan adalah data text atau pesan text, sebagai tempat untuk menyembunyikan pesan ini adalah *file audio* dengan format *wav*. Metode untuk menyembunyikan pesan menggunakan metode *Least Significant Bit*, cara kerja dari metode ini yaitu dengan mengubah nilai bit-bit terakhir dari data WAV dengan data pesan rahasia. Hasil penyembunyian pesan menggunakan metode LSB pada file WAV tidak berpengaruh secara *significant* terhadap kualitas file WAV.

Kata Kunci: *Least Significant Bit (LSB), Penyembunyian Pesan, WAV.*

A. Pendahuluan

Seni penyembunyian pesan ke dalam pesan lain telah ada sejak jaman sebelum masehi, teknik ini dikenal dengan istilah *steganografi*. Teknik penyembunyian pesan kini juga telah banyak diterapkan pada data digital. *Steganografi* banyak dimanfaatkan untuk mengirim pesan melalui jaringan Internet tanpa diketahui oleh orang lain dengan media digital berupa *file* gambar, gambar bergerak maupun suara. Penggunaan steganografi juga menjadi daya tarik bagi banyak orang pada peristiwa penyerangan gedung WTC, 11 September 2001. Pada peristiwa tersebut beredar isu bahwa para teroris menyembunyikan informasi-informasi tentang target penyerangan melalui gambar porno pada *website* tertentu.

Ketika pesan telah sampai ditangan penerima, maka penerima pesan harus mengetahui kunci atau *password* agar dapat membuka data yang disembunyikan. Dengan menggunakan steganografi maka kekhawatiran tentang pencurian data dapat dihindari, karena pesan rahasia tersebut menjadi satu dengan *file* pembawanya dan diproteksi menggunakan *password*.

Untuk dapat melakukan penyembunyian pesan dengan metode LSB diperlukan tempat sebagai media penyimpan. Dalam tulisan ini digunakan file audio dengan format wav seabgai media penyimpan pesan. Format wav ini dipilih karena cenderung memiliki ukuran file yang besar. Dalam kasus ini, pesan text diubah ke dalam bentuk biner dan kemudian disisipkan ke dalam bentuk biner dari file WAV (*digital WAV*)

[1] *Andy Gunawan , Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[2] *Nugroho Agus Haryono,S.Si.,M.Si., Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[3] *Junius Karel T,S.Si.,M.T., Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

B. Pembahasan

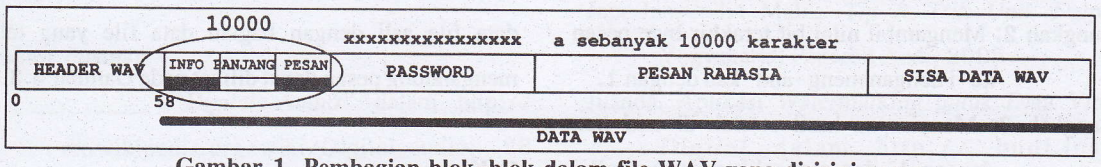
Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling berarti *most significant bit* atau *MSB* dan bit yang paling kurang berarti *least significant bit* atau LSB.

```
10110100
/      \
MSB    LSB
```

Nilai desimal dari MSB di atas adalah 128, sedangkan nilai desimal dari LSB adalah 0, *kemungkinan besarnya nilai dari LSB hanyalah 1 dan 0. Sebelum* masuk pada proses penyisipan, maka harus dilakukan terlebih dulu proses konversi dari file WAV ke dalam bentuk binernya. Untuk melakukan ini diperlukan informasi dari audio wav berupa ukuran sample size dan amplitudo yang digunakan agar proses perubahan LSB nantinya tepat pada bit yang terakhir.

Di bawah ini adalah langkah-langkah untuk menyembunyikan pesan :

1. Langkah pertama adalah membagi file wav menjadi beberapa blok. Untuk Lebih jelasnya dapat dilihat pada Gambar 1.



Gambar 1. Pembagian blok-blok dalam file WAV yang disisipi pesan

Blok pertama adalah header dari *file* wav, panjangnya yaitu 58 byte. Pada bagian ini tidak dapat dilakukan perubahan LSB karena akan merusak file wav (*file* lagu tidak dapat dimainkan).

Blok kedua adalah blok data WAV. Blok data WAV ini kemudian dibagi-bagi lagi menjadi empat bagian, yaitu: bagian pertama untuk menyimpan informasi panjang karakter pesan, bagian kedua untuk menyisipkan password, bagian ketiga untuk menyisipkan pesan rahasia dan bagian keempat adalah sisa data file wav yang tidak diubah.

Dari gambar diatas dapat dilihat bahwa pesan rahasia yang disisipkan adalah huruf a sebanyak 10000 karakter dengan *password* x sebanyak 16 karakter.

2. Langkah kedua adalah membaca membaca password, membaca panjang pesan, dan menghitung panjang pesan, Ketiga input ini kemudian dikonversi ke dalam bentuk *binernya. Dan kemudian menaruhnya ke dalam* bagian masing-masing dengan menggunakan metode LSB.

3. Cara kerja metode LSB dijelaskan berikut ini.

Misalnya akan disisipkan karakter abc ke dalam bagian pesan, maka langkah pertama adalah membaca nilai biner dari nilai ASCII karakter a sebagai berikut.

a -> kode ASCII=97 nilai biner 01100001

Langkah kedua adalah mengambil satu persatu bagian dari setiap bit, yaitu 0,1,1,0,0,0,0,1. Kemudian untuk setiap, akan disisipkan pada data dengan metode LSB, misalnya:

Data 1: 10111111, disisipi bit 0 menjadi 10111110

Data 2: 10101111, disisipi bit 1 menjadi 10101111

Data 3: 10101100, disisipi bit 1 menjadi 10101101

Dan seterusnya.

Jadi pada prinsipnya adalah mengganti bit terakhir dari data dengan nilai bit yang akan disisipkan.

Algoritma penyisipan LSB adalah sebagai berikut:

Misalikan Nilai Bit yang disisipkan =0, dan data yang disisipi misalnya 10111111

Data : 10111111
254 : 11111110 and
Hasil Awal : 11111110,

(tujuannya adalah membuat bit terakhir menjadi 0)

nilai ini kemudian di "or" kan dengan bit yang akan disisipkan

Hasil Awal : 11111110,
Bit : 0 or
Data baru : 10111110

Setelah pesan text disisipkan pada file wav maka file wav dapat dikirimkan pada orang lain. Orang yang dapat membaca pesan adalah orang yang mengetahui password dari pengirim pesan. Langkah untuk membaca pesan adalah sebagai berikut:

Langkah 1: membuat blok-blok data ke dalam 8 byte per blok. Untuk setiap blok dikerjakan langkah 2 sampai dengan langkah 3 untuk i=0,1,2,3, ..., 7.

Langkah 2: Mengambil nilai bit terakhir byte pesan ke-i dengan meng-*and*-kan dengan 1.

Langkah 3: Menyimpan hasil setelah di-*and*-kan dengan 1, dan mengalikan dengan nilai posisi bit, yaitu : $(2^{(7-i)})$

Langkah 4: Menjumlahkan semua hasil perhitungan untuk i=0 sampai dengan i=7.

Langkah 5: Menentukan karakter ASCII yang bersesuaian dengan hasil perhitungan.

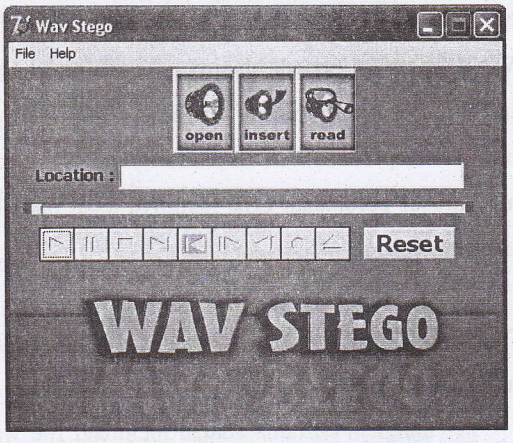
Sebagai contoh pembacaan pesan diberikan berikut ini.

Proses pengambilan nilai LSB dari pesan:

01101010 and 1 = 0 nilai = 0 x 2^7 = 0
00111011 and 1 = 1 nilai = 0 x 2^6 = 64
00110110 and 1 = 0 nilai = 0 x 2^5 = 0
00101110 and 1 = 0 nilai = 0 x 2^4 = 0
10100100 and 1 = 0 nilai = 0 x 2^3 = 0
11001010 and 1 = 0 nilai = 0 x 2^2 = 0
01010101 and 1 = 1 nilai = 0 x 2^1 = 2
10100101 and 1 = 1 nilai = 0 x 2^0 = 1 +
= 67,

karakter ASCII dengan nilai 67 adalah "c", jadi pesan yang terbaca adalah karakter "c".

Tampilan halaman depan aplikasi yang dibuat diberikan dalam Gambar 3.2.



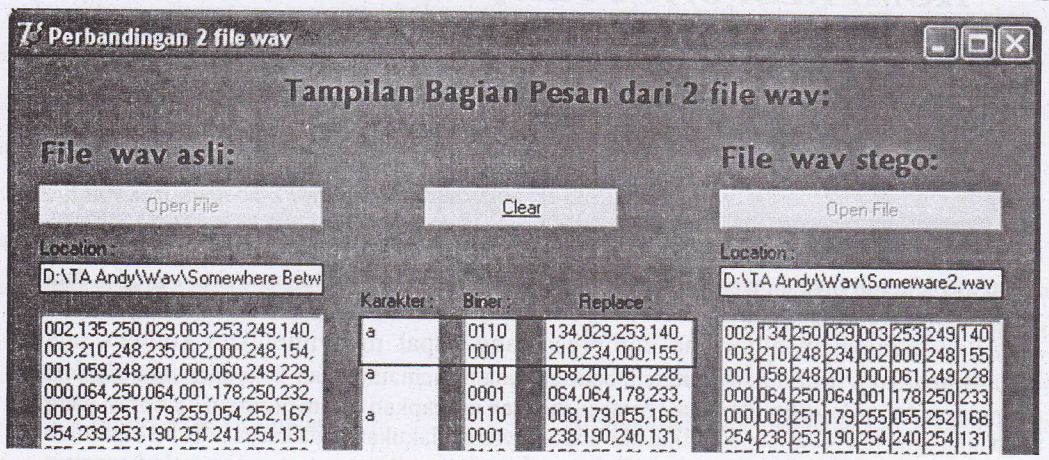
Gambar 3.2. Tampilan halaman depan aplikasi

Untuk dapat melihat perbedaan antara bagian data file asli dengan bagian data file yang telah mengandung pesan dapat dilihat pada Gambar 3.3.

C. Kesimpulan

Kesimpulan dari aplikasi steganografi yang dibuat adalah sebagai berikut :

a. Aplikasi steganografi yang dibuat dapat menyisipkan karakter pesan text dalam jumlah yang banyak.
b. Aplikasi steganografi yang dibuat membutuhkan waktu proses yang relatif lama saat melakukan penyimpanan file, terutama pada *file* wav dengan ukuran yang sangat besar.
c. Proses penyisipan text yang terjadi pada *file* wav



Keterangan gambar:

1. Karakter "a" adalah pesan yang disisipkan dalam file wav
2. Perubahan nilai bit terakhir dapat dilihat pada tampilan file wav stego, contoh: nilai 135 berubah menjadi 134, jika terjadi perubahan nillainya hanya sebesar 1
3. Tiga karakter desimal di atas bernilai 1byte (8 bit), perubahan data terjadi tiap kelipatan 2 dikarenakan nilai samplesize dari file wav adalah 16bit maka perubahan juga dilakukan tiap 2byte (16bit)

Gambar 3.3. Perbandingan bagian data 2 file WAV

tidak menyebabkan perubahan yang berarti pada kualitas suara, sehingga suara yang terdengar tidak dapat dibedakan dengan *file* wav aslinya, hal ini disebabkan dari kecilnya nilai LSB yang terubah.

Daftar Pustaka

_____. "*Digital Audio*" dalam http://en.wikipedia.org/wiki/Digital_audio; 28 Maret 2007

_____. "*Digital Audio Best Practice*" dalam http://www.cdpheritage.org/ digital/audio/documents/CDPDABP_1-2.pdf; 28 Maret 2007

_____. "*the information hiding homepage digital watermarking & steganography*" dalam http:// www.cl.cam.ac.uk/~fapp2/steganography/index.html; 10 November 2006

_____. "*Wave File Format*" dalam http://www.sonicspot.com/guide/wavefiles.html ; 10 November 2006

Antonius Rahmat. *Suara dan Audio* dalam http://lecturer.ukdw.ac.id/ anton/ download/multimedia3.pdf; 11 November 2006

Jaja Jamaludin Malik. *Tip & Trik Unik Delphi.* Yogyakarta: Penerbit Andi, 2005

Juanda. *Aplikasi Watermarking untuk Data Video Digital* dalam http:// budi.insan.co.id/courses/el695/projects/report-juanda.doc,hal 3; 21 Januari 2002

Biodata Penulis:

Nama : Andy Gunawan

NIM : 22012787

TTL : Purbalingga / 12 September 1983

Email : Pecel_lele83@yahoo.com

No Hp : 08175480787