Percobaan Steganalisis pada Berkas Audio dengan Memanfaatkan Audio Fingerprint

Diani Pavitri Rahasta 13509021[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13509021@std.stei.itb.ac.id

***Abstrak*—Steganografi merupakan metode penyembunyian pesan yang cukup bisa mengalihkan perhatian para pencuri pesan. Bentuk pesan yang sama sekali berubah terkadang membuat para analis pesan terkecoh dan mengira tidak ada pesan di dalamnya. Salah satu steganografi yang cukup populer adalah steganografi dalam berkas audio. Masih belum banyak yang melakukan penelitian di bidang ini, baik membuat metode maupun membuat pemecahan. Hal ini menyebabkan steganografi dalam berkas audio jadi lebih mudah. Makalah ini akan menawarkan sebuah cara untuk melihat apakah sebuah berkas audio sudah disisipi pesan atau belum dengan bantuan *audio fingerprint*.**

***Termin Indeks*—Steganografi, audio, *audio fingerprint*.**

I. PENDAHULUAN

Salah satu cabang ilmu yang termasuk dalam kriptografi adalah steganografi. Steganografi adalah ilmu yang menyembunyikan sebuah pesan ke dalam media tertentu. Hal yang istimewa dari steganografi adalah bahwa steganografi menyembunyikan pesan menjadi suatu bentuk yang berbeda dengan aslinya, sementara kriptografi mengubah pesan yang ada ke dalam bentuk yang mungkin sama namun tidak memiliki makna yang dimaksud dari pesan sebenarnya. Karena proses penyembunyian dalam steganografi, ada dua hal yang dibutuhkan dalam melakukan steganografi, pesan yang akan disembunyikan serta media untuk menyembunyikan pesan.

Media yang digunakan untuk menyimpan pesan bisa bermacam-macam. Pada saat ilmu ini baru ditemukan, steganografi dilakukan dengan cara menyimpan pesan ke dalam pesan yang lain. Salah satu yang umum adalah dengan menggunakan huruf pertama dari tulisan sebagai tempat penyembunyian pesan. Saat ini, di era *digital*, media untuk steganografi sudah berkembang jauh. Saat ini steganografi bukan hanya disembunyikan dalam teks, tapi juga dalam gambar, audio, maupun video. Jika teknik untuk memecahkan kriptografi dikenal sebagai kriptanalisis, teknik untuk memecahkan steganografi memiliki nama yang mirip, steganalisis.

Salah satu media penyimpanan yang bisa digunakan untuk steganografi adalah audio. Namun tidak banyak eksplorasi yang dilakukan terhadap bidang ini. Orang cenderung berkonsentrasi kepada media gambar dan mungkin video. Hal ini mungkin disebabkan oleh kualitas audio yang cenderung lebih mudah berubah setelah disisipi pesan.

Karena eksplorasi yang kurang terhadap audio sebagai media penyembunyian pesan, media ini justru paling rawan menjadi media penyembunyian yang tidak ketahuan. Tidak banyak penelitian terhadap media ini, terutama di bidang steganalisis, membuat penyembunyian pesan di dalam media ini menjadi semakin mudah. Padahal, pesan yang disembunyikan tidak selamanya merupakan sesuatu yang baik.

Audio fingerprinting adalah sebuah sistem yang memberikan fungsi tertentu kepada sinyal audio sehingga bisa diubah menjadi rangkaian *string*. Sistem ini menerjemahkan sinyal audio sehingga lebih mudah disimpan dan dicari tapi tetap merepresentasikan audio yang sesungguhnya. Sistem ini bisa dimanfaatkan untuk melakukan steganalisis awal terhadap berkas audio, yaitu mencari tahu apakah ada pesan tersembunyi di dalam suatu berkas audio. Jika diketahui ada pesan yang tersembunyi, proses analisis berikutnya untuk mencari pesan apa yang akan disembunyikan akan menjadi lebih mudah.

II. STEGANOGRAFI

Steganografi merupakan ilmu yang sudah tua dalam mengamankan informasi. Steganografi bisa dibilang bersaudara dengan kriptografi. Jika kriptografi mengatur isi pesan sehingga tidak bisa dibaca, steganografi menyembunyikan pesan di dalam suatu media. Steganografi membutuhkan dua hal untuk bisa dilakukan, yaitu pesan yang akan disembunyikan dan media untuk menyembunyikannya.

Steganografi sampai sekarang masih dianggap efektif sebab pesan disembunyikan dalam media tertentu yang tidak tampak mencolok. Hal ini membuat para kriptanalis yang berusaha untuk mengambil pesannya tidak menyangka bahwa pesannya disembunyikan di media tersebut. Steganografi bisa dilakukan dengan memanfaatkan berbagai macam media penyimpanan, dari teks lain (pesan disisipkan sebagai huruf ke-sekian), audio, gambar, sampai video.

Sampai saat ini, penelitian masih banyak dilakukan untuk mencari cara untuk menyelipkan pesan di dalam

media berupa gambar, audio, maupun video. Ketiga media ini dipilih sebab jika suatu pesan disisipkan di dalamnya, akan sangat kecil kemungkinan seseorang untuk curiga. Selain itu, metode penyisipan pesan yang tepat akan membuat media tersebut seolah-olah tidak pernah diutak-atik sebelumnya.

Salah satu metode yang bisa digunakan untuk menyisipkan pesan ke dalam media-media tersebut adalah metode penyisipan *Least Significant Bit* (LSB). Metode ini memanfaatkan bit dalam media yang tidak digunakan dalam menyimpan data yang sebenarnya untuk menyimpan bit dari pesan. Secara garis besar, prosesnya adalah sebagai berikut. Pesan pertama diubah ke dalam bentuk bit. Bit-bit pesan ini kemudian diberikan sebuah kunci acak untuk bisa disimpan ke dalam bit-bit media yang digunakan. Proses sebaliknya dilakukan untuk menemukan pesan di dalam media tersebut.

III. *AUDIO FINGERPRINT*

Metode *fingerprint* adalah metode yang biasa digunakan untuk menyederhanakan sebuah *file* dengan ukuran yang besar. Untuk kasus kali ini, *fingerprint* digunakan untuk merepresentasikan data audio. Kata *fingerprint* dipilih karena analogi yang dimilikinya. Pada manusia, *fingerprint* atau sidik jari menunjukkan sebuah penanda yang unik dan representatif dari seseorang. Begitu juga dengan *fingerprint* pada audio. *Fingerprint* pada audio biasanya merupakan kumpulan kode yang dihasilkan dari ekstraksi fitur-fitur tertentu pada data audio. Setiap kumpulan kode merepresentasikan satu *file* audio dan tidak ada dua *file* audio yang akan memiliki rangkaian kode yang sama.

Ada beberapa teknik yang dilakukan untuk mengekstrak *fingerprint* dari sebuah *file* audio. Salah satu teknik *fingerprint* yang digunakan dalam industri adalah teknik *fingerprint* yang digunakan oleh aplikasi Shazam [1]. Shazam adalah salah satu aplikasi pengenal lagu yang menerima *query* berupa rekaman dari lagu yang sudah direkam sebelumnya [2]. Aplikasi ini menggunakan teknik *fingerprint* dalam pemrosesan data audio yang dia miliki. Teknik yang digunakan oleh Shazam adalah pengamatan pada titik puncak *spectogram* dari audio. Pencetakan *fingerprint* akan dilakukan dengan cara menyimpan setiap titik puncak dan merumuskannya sehingga menghasilkan rangkaian *string* yang kemudian menjadi *fingerprint*.

Shazam bukanlah pencetus ide pemanfaatan puncak *spectogram* dalam pembuatan *fingerprint*. Teknik serupa diperkenalkan pada tahun 2002 oleh Jaap Haitsma dan Tom Kalker [3]. Teknik ini, yang sepertinya merupakan ide yang dianggap memang memiliki ketahanan yang tinggi untuk *fingerprint* dari sebuah audio, kemudian banyak diadaptasi oleh orang lain sesuai kebutuhan mereka masing-masing dalam membuat pencetak *fingerprint* yang tepat.

A. *Echoprint*

Echoprint bukanlah sebuah aplikasi langsung jadi yang siap pakai, namun Echoprint merupakan sebuah sistem yang bisa tinggal ditempelkan kepada antarmuka yang dikustomisasi sendiri oleh pengguna. Echoprint bersifat *open source*, sehingga siapapun yang ingin membuat aplikasi pengenal musik bisa langsung menggunakan sistem yang sudah disediakan oleh Echoprint. Echoprint juga mengijinkan para pengembang aplikasi untuk menjual aplikasi mereka, meskipun sistem yang mereka sediakan untuk ditempelkan itu bersifat gratis. Pengembang aplikasi hanya perlu memberitahukan kepada Echoprint sebelumnya, dan tentunya mencantumkan bahwa pengembang memanfaatkan Echoprint dalam aplikasinya.

Echoprint sendiri menyatakan bahwa ada tiga bagian dalam Echoprint [8]. Bagian pertama adalah bagian *code*, yang bekerja sebagai penerjemah lagu masukan pengguna ke dalam *fingerprint* yang akan dibandingkan dengan basis data. Bagian *code* dari Echoprint merupakan mesin sederhana yang dibuat dengan menggunakan bahasa pemrograman Python. Bagian ini akan menghasilkan *fingerprint* yang berupa rangkaian *string* acak yang merepresentasikan potongan lagu yang dimasukkan oleh pengguna. Bagian ini yang dibanggakan oleh Echoprint sebab bagian inilah yang mampu merekam dan mengubah lagu ke *fingerprint* tanpa terpengaruh oleh *noise* yang ada saat lagu direkam.

Seperti sudah dituliskan sebelumnya, Echoprint memperlakukan audio dalam bentuk *fingerprint*. *Fingerprint* yang dihasilkan oleh bagian *code* merupakan representasi dari bentuk sinyal audio. Dalam [12] dijelaskan bahwa Echoprint hanya memanfaatkan waktu relatif di antara *onset* yang tertangkap seperti *beat* pada audio. Deteksi *onset* dilakukan di antara 0 sampai 5512.5 Hz dan ada sebuah *threshold* untuk menangkap *onset* tersebut. Selanjutnya sebuah algoritma digunakan untuk mengambil target IOI (inter onset interval). Pasangan IOI yang berhasil diambil dikuantisasi ke dalam unit yang berukuran 23.2 ms, kemudian dikombinasikan untuk membentuk *fingerprint*. Untuk memastikan ketepatan penangkapan *onset*, setiap *onset* memiliki empat pengganti. Enam pasangan IOI dibuat dengan cara memilih pasangan yang mungkin dari empat pengganti tersebut. Ukuran rangkaian *string* dalam *fingerprint* untuk satu detik audio kira-kira 48 karakter.

IV. PERCOBAAN YANG DILAKUKAN

Percobaan dilakukan dengan menggunakan sebuah berkas audio dengan format .mp3. Format ini dipilih sebab berkas audio dengan format .mp3 sangat mudah untuk ditemukan. Tentunya hal ini juga menyebabkan steganografi yang dilakukan menjadi lebih mudah karena tidak perlu terlalu lama mencari tempat penyembunyian untuk pesan.

Penyisipan pesan dilakukan dengan bantuan kakas MP3Stego. MP3Stego merupakan sebuah aplikasi berbasis *command prompt* di komputer yang bisa membantu kita untuk menyisipkan pesan ke dalam sebuah berkas audio. Aplikasi ini bisa didapat secara gratis

dengan mudah di http://www.petitcolas.net/fabien/steganography/mp3stego/. Aplikasi dapat diunduh dan dijalankan setelah sebelumnya melakukan proses kompilasi kode dari aplikasi tersebut. Hal ini perlu dilakukan sebab pengembang aplikasi ini tidak menyediakan aplikasi diap pakainya.

Berkas audio diubah menjadi *audio fingerprint* dengan bantuan kakas dari Echoprint. Bagian yang diperlukan dari Echoprint adalah bagian *codegen* nya saja, yaitu bagian yang mangubah berkas audio menjadi *fingerprint*. Untuk bisa digunakan, bagian *codegen* dari Echoprint ini juga harus dikompilasi terlebih dahulu.

Untuk proses pembandingan kedua *audio fingerprint*, dapat dilakukan secara manual atau dengan memanfaatkan sebuah program kecil. Pembuatan program kecil perlu sebab terkadang akan terjadi kesulitan saat melakukan perbandingan. Hal ini disebabkan oleh ukuran berkas yang sangat besar sehingga merepotkan untuk diperhatikan satu persatu.

Berikut langkah-langkah percobaan yang dilakukan:

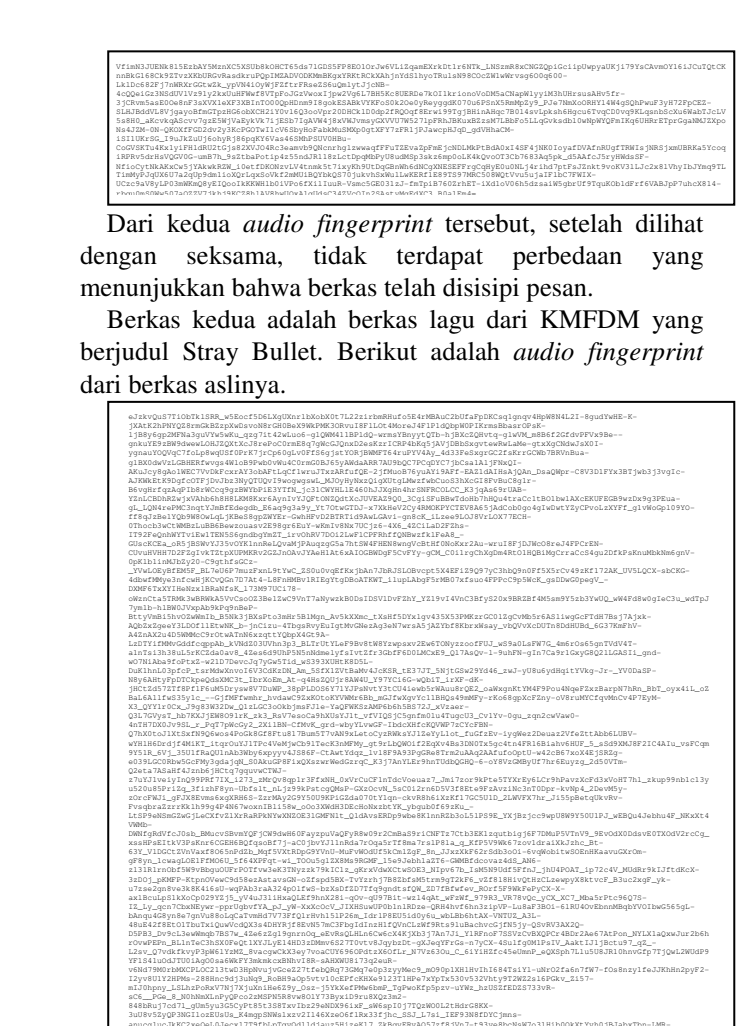
1. Pilih berkas audio untuk disisipi pesan. Berkas harus berformat .mp3 karena baik MP3Stego maupun Echoprint hanya bisa memproses berkas audio dengan format .mp3.
2. Lakukan penyisipan pesan dengan menggunakan MP3Stego. Bagian ini sedikit sulit sebab aplikasi ini berbasis *command prompt*.
3. Lakukan pemrosesan berkas audio menjadi *fingerprint* dengan menggunakan Echoprint. Jangan lupa untuk mengubah kedua berkas audio, yang belum disisipi pesan dan yang sudah disisipi pesan.
4. Bandingkan kedua *audio fingerprint* tersebut.

V. Hasil Percobaan

Percobaan dilakukan pada tiga buah berkas audio. Hal ini dilakukan sebab jika hanya satu berkas yang diuji coba, hasilnya akan cenderung meragukan. Dipilih tiga sebab jika terjadi perbedaan di antara ketiganya, bisa diputuskan mana yang benar dengan mengambil hasil yang lebih banyak. Ketiga berkas yang digunakan merupakan lagu dengan tiga *genre* yang berbeda. Tidak ada alasan khusus, namun mungkin saja *genre* bisa berpengaruh.

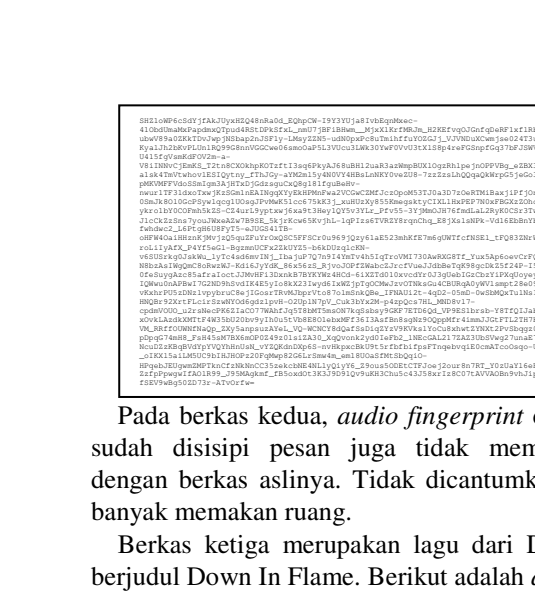
Berkas pertama adalah berkas lagu dari Eminem yang berjudul If I Had. Berikut adalah *audio fingerprint* dari lagu tersebut.

Selanjutnya, berikut adalah *audio fingerprint* dari berkas tersebut yang sudah disisipi pesan.



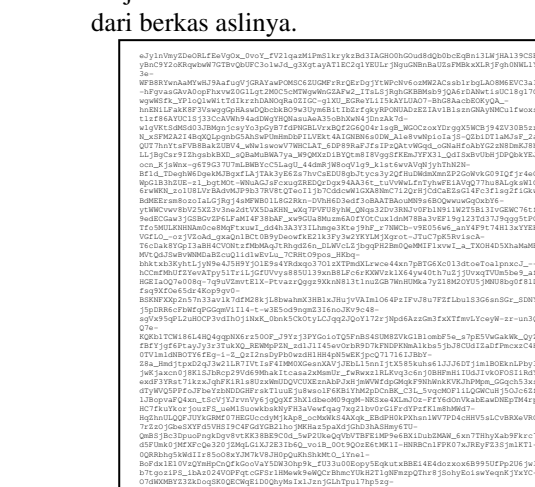
Dari kedua *audio fingerprint* tersebut, setelah dilihat dengan seksama, tidak terdapat perbedaan yang menunjukkan bahwa berkas telah disisipi pesan.

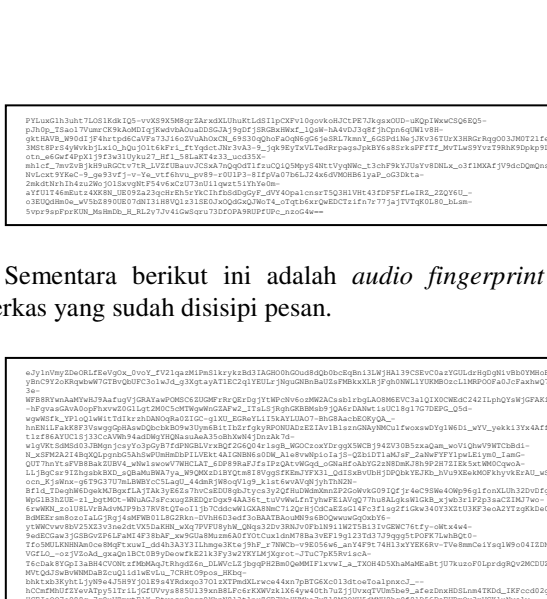
Berkas kedua adalah berkas lagu dari KMFDM yang berjudul Stray Bullet. Berikut adalah *audio fingerprint* dari berkas asalnya.



Pada berkas kedua, *audio file* juga tidak memiliki perbedaan dengan berkas aslinya. Tidak dicantumkan sebab terlalu banyak memakan ruang.

Berkas ketiga merupakan lagu dari Greg yang berjudul Down In Flame. Berikut adalah *audio file* dari berkas aslinya.





Sementara berikut ini adalah *audio file* dari berkas yang sudah disisipi pesan.

Kedua *audio file* tersebut juga tidak memiliki perbedaan.

Setelah dilihat dari tiga berkas yang digunakan, ternyata *audio fingerprint* sebuah lagu tidak berubah setelah disisipi pesan dengan steganografi. Hal ini mungkin terjadi disebabkan adanya mekanisme yang berbeda dari penyisipan pesan ke dalan berkas audio dan pembuatan *audio fingerprint*.

Pada saat pesan disisipkan ke dalam berkas audio menggunakan metode penyisipan ke *least significant bit*, yang berubah dalam berkas tersebut adalah bit bit dari berkas audio tersebut. Pesan disisipkan sehingga secara fisik, jika berkas audio didengarkan tidak akan tertangkap hal yang aneh.

Hal ini yang pada akhirnya menyebabkan *audio fingerprint* dari berkas yang sudah disisipi pesan menjadi tidak ada bedanya. Pembuatan *fingerprint* dari sebuah berkas audio memanfaatkan cara yang paling mirip dengan bagaimana audio tersebut didengar oleh telinga manusia. Echoprint, yang digunakan dalam percobaan ini, menggunakan perbedaan yang terjadi pada puncak gelombang dari berkas audio yang ada. Gelombang yang sudah disisipi pesan mungkin berubah, tapi perubahannya tidak cukup signifikan untuk bisa ditangkap oleh Echoprint saat melakukan pembuatan *audio fingerprint* dari suatu berkas lagu.

VI. KESIMPULAN

A. Steganografi pada berkas audio dilakukan dengan metode penyisipan bit-bit pada *least significant bit* pada berkas audio tersebut.
B. Audio *fingerprint* mengubah berkas audio berdasarkan data sinyal pada audio tersebut.
C. Penyisipan pesan pada berkas audio tidak akan mempengaruhi *audio fingerprint* yang dibentuk terhadap berkas audio tersebut.

REFERENSI

[1] Avery Li-Chun Wang, "An Industrial Strength Audio Search Algorithm," *Proc. 4th ISMIR*, pp. 7-13, 2003.

[2] Shazam Entertainment Ltd. (2013) Shazam. [Online]. http://www.shazam.com/

[3] Jaap Haitsma and Ton Kalker, "A Highly Robust Audio Fingerprinting System," *Proc. 3rd ISMIR*, pp. 144-148, 2002.

[4] Jijun Deng, "An Audio Fingerprinting System Based On Spectral Energy Structure," in *Smart and Sustainable City (ICSSC 2011), IET International Conference*, China, 2011, pp. 1 - 4.

[5] Brian Whitman. (2010) Brian Whitman @ variogr.am. [Online]. http://notes.variogr.am/post/544559482/the-echo-nest-musical-fingerprint-enmfp

[6] Daniel P. W. Ellis, Brian Whitman, Tristan Jehan, and Paul Lamere, "The Echo Nest Musical Fingerprint," 2010.

[7] Brian Whitman. (2012, August) Brian Whitman @ variogr.am. [Online]. http://notes.variogr.am/post/27796385927/the-audio-fingerprinting-at-the-echo-nest-faq

[8] Echoprint. (2010) Echoprint - Open source music identification. [Online]. http://echoprint.me/how

[9] FAL Labs. (2010, August) Tokyo Tyrant Website. [Online]. http://fallabs.com/tokyotyrant/spex.html

[10] FAL Labs. (2010) Tokyo Cabinet Website. [Online]. http://fallabs.com/tokyocabinet/spex-en.html

[11] The Apache Software Foundation. (2011) Apache Solr. [Online]. http://lucene.apache.org/solr/

[12] Daniel P. W. Ellis, Brian Whitman, and Alastair Porter, "Echoprint - An Open Music Identification Service," 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Maret 2013

Diani Pavitri R 13509021