

EXPERT

Jurnal Sistem Informasi



PEMANFAATAN ANIMASI DUA DIMENSI UNTUK PEMBELAJARAN BAHASA JEPANG TINGKAT DASAR

Yuli Syafitri, Umi Anita Sari

IMPLEMENTASI STEGANOGRAFI PADA BERKAS *AUDIO WEV* UNTUK PENYISIPAN PESAN GAMBAR MENGGUNAKAN METODE *LOW BIT CODING*

Dedi Darwis

APLIKASI INTERNAL AUDIT BERBASIS *ISO 9001:2008* DAN *ISO 19011:2002* UNTUK MENUNJANG EFEKTIFITAS PENGENDALIAN MUTU AKADEMIK

Erlangga

ANALISIS PERBANDINGAN METODE TRANSFORMASI *WAVELET* DAN METODE *EIGENFACE* PADA PENGENALAN CITRA WAJAH DENGAN ANALISIS SWOT

Gabriella Genevine, Usman Rizal

ANALISIS *PATTERN & MINUTIAE BASED MATCHING FINGERPRINT JM 250 U* MENGGUNAKAN METODE BIOMETRIK.

Iwan Sugiarto, Taqwan Thamrin

ANALISIS MANAJEMEN RESIKO UNTUK EVALUASI ASET MENGGUNAKAN METODE *OCTAVE ALLEGRO*.

Joshua Jenriwan L Tobing, Ayu Kartika Puspa

ISSN : 2088-5555

Write To Be Experts

Judul	Hal
PEMANFAATAN ANIMASI DUA DIMENSI UNTUK PEMBELAJARAN BAHASA JEPANG TINGKAT DASAR	1 – 5
IMPLEMENTASI STEGANOGRAFI PADA BERKAS <i>AUDIO WAV</i> UNTUK PENYISIPAN PESAN GAMBAR MENGGUNAKAN METODE <i>LOW BIT CODING</i>	6 - 11
APLIKASI INTERNAL AUDIT BERBASIS <i>ISO 9001:2008</i> DAN <i>ISO 19011:2002</i> UNTUK MENUNJANG EFEKTIFITAS PENGENDALIAN MUTU AKADEMIK	12 – 18
ANALISIS PERBANDINGAN METODE TRANSFORMASI <i>WAVELET</i> DAN METODE <i>EIGENFACE</i> PADA PENGENALAN CITRA WAJAH DENGAN ANALISIS SWOT	19 – 22
ANALISIS <i>PATTERN & MINUTIAE BASED MATCHING FINGERPRINT JM 250 U</i> MENGGUNAKAN METODE BIOMETRIK	23 – 27
ANALISIS MANAJEMEN RESIKO UNTUK EVALUASI ASET MENGGUNAKAN METODE <i>OCTAVE ALLEGRO</i>	28 - 30

Fakultas Ilmu Komputer
Universitas Bandar Lampung

JMSIT	Volume 05	Nomor 01	Lampung Juni 2015	ISSN 2088-5555
-------	-----------	----------	----------------------	-------------------

TIM PENYUNTING

Ketua Tim Redaksi:

Taqwan Thamrin,ST,M.Sc

Penyunting Ahli

Mustofa Usman, Ph.D

Dr.Iing Lukman,M.Sc.

Usman Rizal, ST.,MMSI

Penyunting:

Fenty Ariani,S.Kom,M.Kom

Wiwin Susanty,S.Kom,M.Kom

Ayu Kartika Puspa,S.Kom,M.TI

Erlangga,S.Kom,M.Kom

Iwan Purwanto,S.Kom.,MTI

Pelaksana Teknis:

Zulkaisar, S.Kom

Alamat Penerbit/Redaksi:

Pusat Studi Teknologi Informasi

Fakultas Ilmu Komputer

Universitas Bandar Lampung

Gedung Business Center Lt.2

Jl,Zainal Abidin Pagar Alam No.26

Bandar Lampung

Telp.0721 – 774626

Email: Journal.expert@ubl.ac.id

IMPLEMENTASI STEGANOGRAFI PADA BERKAS AUDIO WAV UNTUK PENYISIPAN PESAN GAMBAR MENGGUNAKAN METODE LOW BIT CODING

Dedi Darwis

Program Studi Magister Ilmu Komputer, Universitas Budi Luhur
Jalan Raya Ciledug, Petukangan Utara, Jakarta Selatan, 12260

dedikomdat@gmail.com

Abstrak

Dengan semakin berkembangnya teknologi informasi dan telekomunikasi, maka perhatian pada tingkat keamanan akan menjadi semakin penting. Salah satunya adalah tingkat keamanan pengiriman data atau informasi. Peningkatan keamanan pengiriman data dapat dilakukan dengan menggunakan steganografi. Steganografi adalah teknik menyembunyikan pesan ke dalam sebuah media pembawa (*carrier*). Salah satu media yang dapat digunakan adalah berkas audio. Penelitian ini membahas tentang penerapan steganografi pada berkas audio WAV dan metode steganografi yang digunakan adalah metode Low Bit Coding. Dari hasil uji coba, diketahui bahwa dengan metode Low Bit Coding penyisipan dan ekstraksi pesan dapat dilakukan dengan baik. Jenis pesan yang dapat disisipkan adalah pesan gambar dengan format JPEG/ JPG. Penyisipan pesan tidak berpengaruh terhadap ukuran berkas audio, akan tetapi berkas audio yang telah disisipi pesan (*stego*) tidak tahan terhadap kompresi, manipulasi amplitudo dan pemotongan audio.

Kata Kunci : Steganografi, Audio Wav, Low Bit Coding, Pesan Gambar

1. Pendahuluan

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*). Kelebihan steganografi terletak pada sifatnya yang tidak menarik perhatian atau kecurigaan orang lain.

Salah satu media yang dapat digunakan sebagai *carrier* adalah berkas audio. Teknik steganografi pada berkas audio memanfaatkan kelemahan pendengaran manusia, karena kualitas suara antara berkas audio asli dengan berkas audio yang telah disisipkan pesan rahasia tidak jauh berbeda. Salah satu metode steganografi audio yang sering digunakan adalah *Low Bit Coding*. Metode ini diterapkan dengan mengganti bit-bit yang tidak terlalu berpengaruh dari berkas audio dengan bit-bit pesan.

Adapun tujuan dari penelitian ini adalah menerapkan steganografi pada berkas audio WAV dengan menggunakan metode *Low Bit Coding*. Metode *Low Bit Coding* ini diujicoba untuk melakukan proses penyisipan dan ekstraksi pesan.

Dimana jenis pesan yang dapat disisipkan adalah pesan gambar dengan format JPEG/ JPG.

2. Landasan Teori

a. Steganografi

Steganografi merupakan suatu cabang ilmu yang mempelajari tentang bagaimana menyembunyikan suatu informasi rahasia di dalam suatu informasi lainnya. [Ariyus, Dony, 2006].

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 kerahasiaan komunikasinya bukan pada datanya. [Alam, Ibnu.2008].

Seiring perkembangan teknologi terutama teknologi komputasi juga bertambahnya kebutuhan dan keinginan dengan kontinuitas yang tinggi, steganografi merambah juga ke media digital. Ada dua proses utama dalam steganografi digital yaitu penyisipan (*embedding/encoding*) dan penguraian (*extraction/decoding*) pesan. Pesan dapat berupa *plaintext*, *chipertext*, citra, atau apapun yang dapat ditempelkan ke dalam *bit-stream*. *Embedding* merupakan proses menyisipkan pesan ke dalam *berkas* yang belum dimodifikasi, yang disebut media cover (*cover object*). Kemudian media cover dan pesan yang ditempelkan membuat media *stego* (*stego object*). *Extraction* adalah proses menguraikan pesan yang tersembunyi dalam media *stego*. Suatu password khusus (*stego key*) juga dapat

digunakan secara tersembunyi, pada saat penguraian selanjutnya dari pesan. Ringkasnya, steganografi adalah teknik menanamkan *embedded message* pada suatu *cover object*, dimana hasilnya berupa *stego object*.

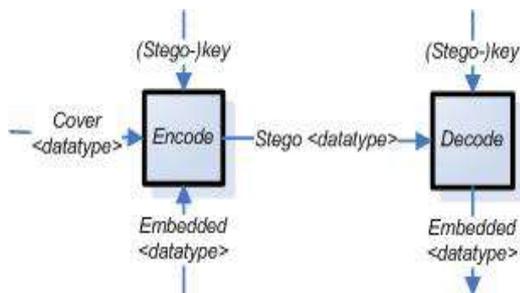
Pihak yang terkait dengan steganografi antara lain *embeddor*, *extractor*, dan *stegoanalyst*. Alam, Ibnu.(2008). *Embeddor* adalah orang yang melakukan *embedding* dengan menggunakan aplikasi steganografi, *extractor* adalah orang yang melakukan *extract stego image* dengan menggunakan aplikasi steganografi. Sedangkan *stegoanalyst* adalah orang yang melakukan steganalisis. Steganalisis merupakan ilmu dan seni untuk mendeteksi pesan yang tersembunyi dalam steganografi.

b. 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. Ariyus, Dony, (2008).



Gambar 1 . Proses Steganografi Modern

c. Metode Low Bit Coding

Dalam steganografi audio terdapat beberapa teknik dalam penyisipan pesan, salah satunya adalah metode *Low Bit Coding*. Metode *Low Bit Coding* adalah cara yang paling sederhana untuk menyimpan data kedalam file audio. Teknik ini diimplementasikan dengan mengganti bit yang paling tidak penting atau *low significant bit*

(*LSB*) pada setiap titik *sampling* dengan string berkode biner (*coded binary string*), kita dapat menyisipkan sejumlah besar data ke dalam suara digital. Kelemahan metode ini adalah lemahnya kekebalan terhadap manipulasi. Pada prakteknya, metode ini hanya berguna pada lingkungan *digital-to-digital* yang tertutup.

d. WAV (Waveform Data)

Format suara *WAV* merupakan standar dari *RIFF (Resource Interchange File Format)* yang dibentuk oleh *Microsoft*. Format suara *WAV* dipilih karena format ini banyak digunakan, dan memiliki kualitas suara yang sangat baik. Metode yang akan digunakan adalah *low bit coding*. Metode ini dipilih, karena metode ini lebih mudah dilakukan, dan metode ini tidak menambah ukuran dari data suara *WAV*.

e. JPEG (Joint Photographich Experts Group)

Dalam ilmu komputer, *JPEG* adalah metode kompresi yang umum digunakan untuk gambar-gambar fotografi. *JPEG* merupakan singkatan dari *Joint Photographic Experts Group*, nama dari komite yang menetapkan standar *JPEG*. Pada tahun 1994, standar *JPEG* disahkan sebagai *ISO 10918-1*.

Standar *JPEG* memberikan spesifikasi *codec* kompresi data ke dalam *stream data byte* dan pendekompresian kembali ke bentuk gambar serta format data penyimpanannya. Metode kompresi data yang digunakan umumnya berupa *lossy compression*, yang membuang detail visual tertentu, dimana hilangnya data tersebut tidak bisa dikembalikan. File *JPEG* memiliki ekstensi *.jpg*, *.jpeg*, *.jpe*, *.jfif*, dan *.jif*.

3. Perancangan

a. Perancangan Spesifikasi Aplikasi

Spesifikasi aplikasi meliputi input (masukan) dan output (keluaran). Masukan bergantung pada proses yang akan dilakukan (proses penyisipan atau ekstraksi). Pada proses penyisipan dibutuhkan dua buah masukan yaitu media cover dan pesan rahasia. Media cover yang digunakan adalah berkas audio *WAV* dan pesan rahasianya berupa gambar dengan format *JPEG/ JPG*. Sedangkan pada proses ekstraksi hanya dibutuhkan satu buah masukan. Masukan tersebut adalah berkas audio *WAV* yang telah disisipi pesan rahasia (*stego*).

Keluaran yang dihasilkan juga bergantung pada proses yang akan dilakukan. Pada proses penyisipan, keluaran yang dihasilkan adalah berkas audio *WAV* yang telah disisipi pesan rahasia (*stego*). Sedangkan pada proses ekstraksi, keluarannya adalah pesan rahasia berupa gambar dengan format *JPEG/ JPG*.

b. Perancangan Algoritma

Metode yang digunakan dalam pembuatan

aplikasi steganografi ini adalah metode *Low Bit Coding*. Ditinjau dari segi fungsional, maka aplikasi yang akan dibangun memiliki 2 proses, yaitu proses *encoding* (penyisipan) dan proses *decoding* (ekstraksi). Algoritma proses penyisipan dan ekstraksi dengan metode *Low Bit Coding* akan dijelaskan pada bagian ini.

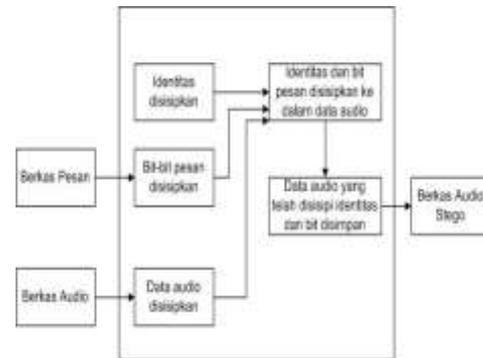
1. Algoritma Proses Penyisipan

Langkah-langkah proses penyisipan bit-bit pesan (gambar) ke dalam berkas audio WAV adalah sebagai berikut :

- Berkas audio WAV dibaca dan byte-byte hasil pembacaan disimpan ke dalam variable "fid1".
- Informasi mengenai berkas audio disiapkan dengan cara mengambil 44 byte pertama dari "fid1" dan pindahkan ke dalam variable "info". Byte-byte ini berisi informasi mengenai *chunk header*, *chunk fmt*, *chunk data* dan tidak dapat disisipi bit pesan.
- Byte sampel data (byte ke 45 sampai byte terakhir) dipindahkan ke dalam variable "dta".
- Pesan berupa gambar diinput.
- 16 bit unik disiapkan dan disimpan ke dalam variable "identitas". Bit-bit ini dibutuhkan sebagai penanda apakah di dalam berkas audio terdapat pesan gambar atau tidak. Identitas ini didefinisikan oleh *programmer* dengan seunik mungkin dan identitas yang digunakan adalah [1100110011001100].
- Pesan gambar dan ukurannya dibaca.
- Sebelum dilakukan penyisipan, terlebih dahulu dicek apakah penyisipan dapat dilakukan atau tidak. Pengecekan dilakukan berdasarkan jumlah byte sampel data dan jumlah bit identitas + bit ukuran pesan + bit pesan. Jika jumlah bit identitas + bit ukuran pesan + bit pesan lebih kecil dari jumlah byte sampel data, maka proses penyisipan dapat dilakukan.
- Apabila pengecekan bernilai ya, maka langkah selanjutnya adalah pesan dan ukurannya diubah ke dalam bentuk bit. Kemudian bit identitas, bit ukuran pesan, dan bit pesan disisipkan ke dalam "dta". Penyisipan dilakukan dengan mengganti bit pertama (bit yang tidak terlalu berpengaruh) dari setiap byte sampel data. Identitas disisipkan pada byte pertama sampai byte keenambelas. Sedangkan bit ukuran pesan dan bit pesan disisipkan pada byte ketujuhbelas dan seterusnya.
- Data audio yang telah disisipi pesan disimpan dengan mengikutsertakan informasi berkas audio dari variable "info". Informasi ini ditulis pada awal berkas. Jika informasi ini tidak diikutsertakan, maka berkas audio tidak

dapat dikenali.

Adapun algoritma proses penyisipan ini dapat dilihat pada gambar berikut :



Gambar 2 . Proses Penyisipan

Berikut ini adalah penggalan program untuk proses *encoding* yang dibuat dengan menggunakan Matlab :

- ```
[handles.fnameinput,handles.pnameinput]=
uigetfile('*.wav','Pilih file audio');
fid1=fopen([handles.pnameinput
handles.fnameinput], 'r');
```

Fungsi *uigetfile* digunakan untuk menampilkan jendela pencarian file audio yang berekstensi \*.wav. Nama file terpilih akan disimpan ke dalam variable *fnameinput* dan pathnya akan disimpan ke dalam variable *pnameinput*. File audio kemudian dibuka dengan menggunakan fungsi *fopen* dan hasil pembacaannya disimpan ke dalam variable *fid1*.
- ```
info=fread(fid1,44,'uint16');
```

Fungsi *fread* digunakan untuk mengambil 44 byte pertama dari "fid1" dan dipindahkan ke dalam variable "info". Byte-byte ini berisi informasi mengenai berkas audio.
- ```
[dta,count]=fread(fid1,inf,'uint16');
fclose(fid1);posisi=1;
```

Fungsi *fread* berikutnya digunakan untuk membaca dan memindahkan byte-byte sampel data ke dalam variable *dta*. Sedangkan variable *count* digunakan untuk membaca dan menyimpan jumlah byte sampel data.
- Algoritma Proses Ekstraksi Langkah-langkah proses ekstraksi bit-bit pesan dari berkas suara stego adalah sebagai berikut ini :

  - Berkas audio stego dibaca dan byte-byte hasil pembacaan disimpan ke dalam variable "fid1".
  - 44 byte pertama dari "fid1" yang berisi informasi mengenai berkas audio dipindahkan ke dalam variable "info".
  - Byte sampel data (byte ke 45

sampai byte terakhir) dipindahkan ke dalam variable “dta”.

- d. Bit pertama dari byte pertama sampai byte keenambelas diekstrak dari dta. Jika hasilnya sama dengan salah satu identitas pada saat penyisipan, maka di dalam berkas tersebut terdapat pesan rahasia dan proses ekstraksi dapat dilakukan.
- e. Apabila pengecekan bernilai ya, maka langkah selanjutnya adalah bit ukuran pesan dan bit pesan diekstrak. Bit-bit pesan yang telah diekstrak dikembalikan ke bentuk semula berdasarkan ukuran pesan.
- f. Pesan gambar hasil ekstraksi terlebih dahulu harus disimpan dengan ekstensi JPEG/ JPG, baru kemudian gambarnya ditampilkan.

Adapun algoritma proses ekstraksi dapat dilihat pada gambar berikut :



Gambar 3 . Proses Ekstraksi

#### 4. Uji Coba dan Analisa Perancangan Sistem

Pengujian dilakukan berdasarkan spesifikasi aplikasi dan juga terhadap ketahanan data. Pengujian berdasarkan spesifikasi aplikasi meliputi pengujian kesesuaian proses, kesesuaian data, dan kualitas suara. Sedangkan pengujian ketahanan data hanya dilakukan terhadap berkas suara WAV *stego*. Untuk melakukan pengujian, digunakan beberapa berkas audio WAV dengan spesifikasi seperti yang terdapat pada Tabel 1.

Tabel 1 . Daftar Berkas Audio

|   | Nama Berkas Audio WAV | Durasi    | Ukuran Data (Byte) | Jenis Suara | Kanal  |
|---|-----------------------|-----------|--------------------|-------------|--------|
| 1 | Kalimba.wav           | 03:27.804 | 36,656,684         |             | Stereo |
| 2 | Flaxen Hair.wav       | 01:41.094 | 17,833,004         |             | Stereo |
| 3 | Sleep Away.wav        | 02:44.700 | 21,081,644         |             | Stereo |

Selain berkas audio, dalam pengujian digunakan juga beberapa sampel gambar. Berikut ini adalah uraian dari pesan gambar :

Pelaksanaan uji coba menggunakan dua buah sampel file gambar, yaitu budiluhur.jpg dan budiluhur\_bw.jpg. Kedua gambar ini berisi logo Universitas Budi luhur yang disajikan dalam

bentuk berwarna (budiluhur.jpg) dan hitam putih (budiluhur\_bw).



Gambar 4 . Gambar yang akan disisipkan

#### a. Pelaksanaan Pengujian

Pada pelaksanaan pengujian dibutuhkan beberapa perangkat lunak pendukung, yaitu *Free WAV to MP3 Converter 7.3.2* dan *Wave Editor 3.1.0.0*. Perangkat lunak *Free WAV to MP3 Converter* digunakan untuk mengkompres audio WAV *stego*, yaitu dengan mengubah format audio WAV *stego* menjadi MP3 (dengan ekstensi .mp3) dan sebaliknya. Sedangkan *Wave Editor* digunakan untuk melakukan manipulasi amplitudo, pemotongan dan dapat juga menampilkan grafik sinyal dari audio WAV *stego*.

Tabel 2. Ukuran File Gambar

|   |                  |                  |         |
|---|------------------|------------------|---------|
| 1 | budiluhur.jpg    | Color/ Wama      | 12,6 KB |
| 2 | budiluhur_bw.jpg | Grayscale/ Hitam | 9,10 KB |

#### b. Pengujian Berdasarkan Spesifikasi Aplikasi

Hasil pengujian terhadap kesesuaian proses, kesesuaian data, dan kualitas suara dapat dilihat pada Tabel 3. Hasil pengujian pada Tabel 3 menunjukkan bahwa perangkat lunak berhasil untuk ketiga faktor pengujian. Pada pengujian terhadap kesesuaian proses, perangkat lunak dapat melakukan proses penyisipan dan ekstraksi dengan baik.

Pengujian terhadap kesesuaian data, menunjukkan bahwa data yang berhasil diekstrak dari audio WAV *stego* bersesuaian dengan data yang disisipkan. Dan pada pengujian terhadap kualitas suara, baik berdasarkan pendengaran maupun secara visual melalui grafik sinyal, dapat dikatakan bahwa kualitas suara antara berkas audio asli dengan berkas audio *stego* adalah sama. Gambar 4 berikut ini adalah salah satu gambar yang menunjukkan grafik sinyal audio sebelum dan sesudah dilakukan

penyisipan.

Setelah proses penyisipan, hal lain yang dapat diperbandingkan adalah ukuran berkas audio sebelum dan setelah penyisipan.

**b. Pengujian Ketahanan Data**

Pengujian ketahanan data dilakukan terhadap dua faktor, yaitu kompresi dan manipulasi amplitudo.

**Tabel 3. Pengujian Ketahanan Data**

| No. | Nama Berkas Audio WAV | Pesan Gambar di Dalamnya | Amplitudo       | Nama Berkas Audio | Ekstraksi |
|-----|-----------------------|--------------------------|-----------------|-------------------|-----------|
| 1   | Kalimba_1.wav         | Budiakur.jpg             | ditambah        | Kalimba_1.wav     | Berhasil  |
|     |                       |                          | dikurangi (90%) | Kalimba_1.wav     | Berhasil  |
| 2   | Kalimba_2.wav         | Budiakur_bv.jpg          | ditambah        | Kalimba_2.wav     | Berhasil  |
|     |                       |                          | dikurangi (50%) | Kalimba_2.wav     | Berhasil  |
| 3   | Fh_1.wav              | Budiakur.jpg             | ditambah        | Fh_1.wav          | Berhasil  |
|     |                       |                          | dikurangi (90%) | Fh_1.wav          | Berhasil  |
| 4   | Fh_2.wav              | Budiakur_bv.jpg          | ditambah        | Fh_2.wav          | Berhasil  |
|     |                       |                          | dikurangi (50%) | Fh_2.wav          | Berhasil  |
| 5   | Sa_1.wav              | Budiakur.jpg             | ditambah        | Sa_1.wav          | Berhasil  |
|     |                       |                          | dikurangi (50%) | Sa_1.wav          | Berhasil  |
| 6   | Sa_2.wav              | Budiakur_bv.jpg          | ditambah        | Sa_2.wav          | Berhasil  |
|     |                       |                          | dikurangi (50%) | Sa_2.wav          | Berhasil  |

- **Kompresi**  
Kompresi audio *stego* dari WAV ke MP3 dan sebaliknya dilakukan dengan menggunakan *Free WAV to MP3 Converter*. Hasil pengujian proses ekstraksi setelah kompresi terhadap berkas audio *stego* dapat dilihat pada Tabel 3.
- **Manipulasi Amplitudo**  
Manipulasi Amplitudo dilakukan dengan menggunakan *Wave Editor*. Hasil pengujian ekstraksi setelah manipulasi amplitudo terhadap beberapa berkas audio *stego* dapat dilihat pada Tabel 4.

**Tabel 4. Hasil Pengujian Ekstraksi Setelah Manipulasi Amplitudo**

| No. | Nama Berkas Audio WAV | Pesan Gambar    | Penyisipan | Nama Berkas Audio | Ekstraksi | Kesesuaian Data | Kualitas Suara |
|-----|-----------------------|-----------------|------------|-------------------|-----------|-----------------|----------------|
| 1   | Kalimba.wav           | budiakur.jpg    | Berhasil   | Kalimba_stabagan  | Berhasil  | Sesuai          | Sesuai         |
|     |                       |                 | Berhasil   | Kalimba_stabagan  | Berhasil  | Sesuai          | Sesuai         |
| 2   | Flauto Hain.wav       | Budiakur_bv.jpg | Berhasil   | Fh_1              | Berhasil  | Sesuai          | Sesuai         |
|     |                       |                 | Berhasil   | Fh_2              | Berhasil  | Sesuai          | Sesuai         |
| 3   | Sleep Away.wav        | budiakur.jpg    | Berhasil   | Sa_1_wandergrah   | Berhasil  | Sesuai          | Sesuai         |
|     |                       |                 | Berhasil   | Sa_2_wandergrah   | Berhasil  | Sesuai          | Sesuai         |

Setelah melakukan pengujian, semua proses ekstraksi gambar dari semua berkas audio *stego* yang telah dimanipulasi amplitudonya dinyatakan berhasil.

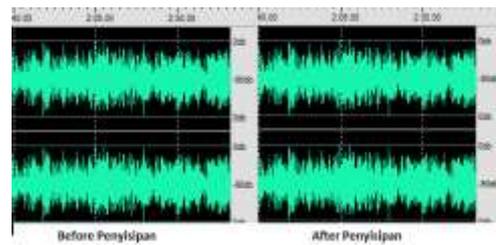
**c. Analisis Hasil Pengujian**

Dari pengujian yang telah dilakukan,

dapat dilakukan beberapa analisis terhadap hasil pengujian tersebut. Berikut ini merupakan analisis terhadap hasil pengujian berdasarkan spesifikasi aplikasi dan ketahanan data.

**a. Analisis Hasil Pengujian Berdasarkan Spesifikasi Aplikasi**

Hasil pengujian berdasarkan spesifikasi aplikasi menunjukkan bahwa perangkat lunak steganografi ini berhasil untuk setiap faktor pengujian yang dilakukan. Faktor yang diuji meliputi faktor kesesuaian proses, kesesuaian data, dan kualitas suara.



**Gambar 5. Grafik Sinyal Sebelum dan Sesudah Penyisipan**

Pada pengujian kesesuaian proses, perangkat lunak dapat melakukan proses penyisipan dan ekstraksi pesan gambar dengan baik. Proses penyisipan berhasil jika ukuran data (jumlah bit identitas + bit ukuran pesan gambar + bit pesan gambar) tidak lebih besar dari jumlah byte sampel data audio yang akan disisipi pesan gambar.

Pada pengujian kesesuaian data, gambar yang berhasil diekstrak bersesuaian dengan gambar yang disisipkan. Kesesuaian ditinjau dari bentuk dan ukuran pesan gambar.

Pengujian kualitas suara menunjukkan bahwa penyisipan bit-bit pesan gambar ke dalam sampel data audio WAV tidak mempengaruhi kualitas suara. Hal ini disebabkan karena perubahan pada bit pertama atau *low significant bit* sangat sulit dideteksi oleh pendengaran manusia. Dan hal ini juga ditunjukkan oleh grafik sinyal audio sebelum dan sesudah penyisipan yang hampir tidak kelihatan perbedaannya.

Selain ketiga faktor di atas, hasil uji coba juga menunjukkan bahwa ukuran berkas audio WAV sebelum dan setelah penyisipan adalah sama. Hal ini terjadi karena penyisipan pesan dilakukan dengan mengganti bit yang tidak terlalu berpengaruh (*low significant bit*) dengan bit-bit pesan gambar, bukan dengan menambah bit baru ke dalam byte sampel data audio.

## b. Analisis Hasil Pengujian Ketahanan Data

Pada pengujian ketahanan data terhadap kompresi dan manipulasi amplitudo, seluruh hasil menunjukkan bahwa pesan gambar tidak dapat diekstrak. Proses kompresi dan manipulasi amplitudo tidak mengubah ukuran berkas audio, akan tetapi menyebabkan perubahan pada bit pertama atau *low significant bit* dari sampel data audio. Dimana bit pertama merupakan tempat untuk menyisipkan bit identitas, bit ukuran pesan, dan bit pesan. Jika terjadi kerusakan pada bit identitas, maka otomatis bit ukuran pesan dan bit pesan dalam keadaan apapun (rusak/ tidak) tidak dapat dibaca dan diekstrak. Dan berdasarkan hasil pengujian, maka dapat disimpulkan bahwa penyisipan pesan gambar dengan metode *Low Bit Coding* tidak tahan terhadap kompresi dan manipulasi amplitudo.

Pada pengujian dengan pemotongan, secara keseluruhan hasil menunjukkan dua kemungkinan, yaitu pesan gambar dapat diekstrak atau tidak sama sekali. Semua pemotongan audio diawal menyebabkan pesan tidak dapat diekstrak. Hal ini dikarenakan perubahan atau kerusakan pada bit identitas. Kerusakan pada bit identitas ini menyebabkan bit ukuran pesan dan bit pesan dalam keadaan apapun (rusak/tidak) tidak dapat diekstrak. Sedangkan ekstraksi setelah pemotongan ditengah dan akhir audio bisa berhasil dan gagal. Ekstraksi berhasil jika bagian audio yang dibuang tidak termasuk byte sampel data audio yang telah disisipi bit identitas, bit ukuran pesan, atau bit pesan.

## 5. Kesimpulan

Steganografi adalah teknik menyembunyikan pesan ke dalam sebuah media pembawa (*carrier*). Penelitian ini membahas tentang penerapan steganografi pada berkas audio WAV, yang diimplementasikan dengan pengembangan aplikasi menggunakan Matlab 7.0. Metode steganografi yang digunakan adalah metode *Low Bit Coding*. Penyisipan pesan dengan metode ini dilakukan dengan mengganti bit pertama atau bit

yang tidak terlalu berpengaruh dengan bit-bit pesan.

Dari hasil uji coba berdasarkan spesifikasi aplikasi, dengan metode *Low Bit Coding* proses penyisipan dan ekstraksi pesan dapat dilakukan dengan baik. Penyisipan dapat dilakukan jika jumlah bit pesan dan bit informasi pendukung tidak lebih besar dari jumlah byte sampel data audio. Untuk kesesuaian data, pesan hasil ekstraksi dan pesan awal adalah sama. Penyisipan pesan dengan metode *Low Bit Coding* juga tidak berpengaruh terhadap kualitas suara karena sulit dideteksi oleh pendengaran manusia. Proses penyisipan juga tidak mengubah ukuran berkas audio atau dengan kata lain, ukuran berkas audio sebelum dan setelah penyisipan adalah sama. Dan dari hasil pengujian ketahanan data, dapat disimpulkan bahwa berkas audio yang telah disisipi pesan (*stego*) tidak tahan terhadap tiga faktor, yaitu kompresi, manipulasi amplitudo dan pemotongan audio. Pesan sama sekali tidak dapat diekstrak setelah audio *stego* dikompres atau dimanipulasi amplitudonya. Sedangkan setelah pemotongan, masih ada kemungkinan pesan dapat diekstrak. Hal ini disebabkan karena bagian audio yang dipotong tidak termasuk byte sampel data audio yang telah disisipi bit pesan dan bit informasi pendukung.

## 6. Daftar Pustaka

- [1] Alam, Ibnu., *Aplikasi Kode Huffman dalam Kompresi Gambar Berformat JPEG*, Makalah, Institut Teknologi Bandung.
- [2] Alatas, Putri. 2009, *Implementasi Teknik Steganografi Denganmetode LSB Pada Citra Digital*, Tugas Akhir, Universitas Gunadarma.
- [3] Ariyus, Dony, 2006. *Kriptografi Keamanan Data dan Komunikasi*. Graha Ilmu, Yogyakarta.
- [4] Ariyus, Dony, 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Andi Offset, Yogyakarta.
- [5] Sadikin, Rifki, 2012. *Kriptografi untuk Keamanan Jaringan*. Andi Offser, Yogyakarta.
- [6] Anhar, (2010). *Membuat Aplikasi DataBase MySQL*. Yogyakarta: Gava Media.

Redaksi :  
Pusat Studi Teknologi Informasi (PSTI).  
Gedung Business Center Lt 2  
Jl. Zainal Abidin No. 26 Bandar Lampung  
Telp. 0721 - 774626  
SistemInformasi@ubl.ac.id



9 772088 555000