JOINT

Journal of Informatics and Technology, Vol 2, No 3, Tahun 2013, p 1-12
http://ejournal-s1.undip.ac.id/index.php/joint

Implementasi dan Studi Perbandingan Steganografi pada File Audio WAVE Menggunakan Teknik *Low-Bit Encoding* dengan Teknik *End Of File*

ARTIKEL ILMIAH

Disusun Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer
pada Jurusan Ilmu Komputer / Informatika

Disusun oleh :

Isa Kurniawan

J2F007024

JURUSAN ILMU KOMPUTER / INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO

2013

JOINT

Journal of Informatics and Technology, Vol 2, No 3, Tahun 2013, p 1-12
http://ejournal-s1.undip.ac.id/index.php/joint

IMPLEMENTASI DAN STUDI PERBANDINGAN STEGANOGRAFI PADA FILE AUDIO WAVE MENGGUNAKAN TEKNIK *LOW-BIT ENCODING* DENGAN TEKNIK *END OF FILE*

Isa Kurniawan
Jurusan Ilmu Komputer / Informatika FSM UNDIP Semarang
Email : izhae_present@yahoo.co.id

Drs. Eko Adi Sarwoko, M.Kom.
Dosen Jurusan Ilmu Komputer / Informatika FSM UNDIP Semarang

Drs. Djalal Er Riyanto, MI.Komp.
Dosen Jurusan Ilmu Komputer / Informatika FSM UNDIP Semarang

ABSTRAK

Menjaga keamanan informasi/data yang bersifat pribadi/rahasia dari pihak yang tidak berkepentingan terhadap data tersebut dapat dilakukan dengan berbagai cara, diantaranya dengan steganografi. Steganografi adalah ilmu yang menyembunyikan data di dalam media penampung sehingga keberadaan data tersebut tidak dapat diketahui. Untuk penerapannya dapat digunakan Teknik *Low-Bit Encoding* dan Teknik *End Of File*. Pada tugas akhir ini diimplementasikan steganografi menggunakan kedua teknik tersebut dengan file audio WAV sebagai media penampung dan data yang disembunyikan berupa sebuah file. Di dalam implementasi diperlukan juga data lainnya untuk disisipkan, yaitu *mark* (5 byte), nama file (32 byte), ekstensi file (4 byte), panjang file (4 byte), dan MD5*Password* (32 byte). Hasil perbandingan yang diperoleh, ukuran data (dalam byte) yang dapat disisipkan menggunakan Teknik *Low-Bit Encoding* bergantung pada file WAV (tidak dapat melebihi jumlah sampel audio file WAV yang telah dibagi delapan), sedangkan untuk Teknik *End Of File* ukuran data yang disisipkan dapat melebihi ukuran file WAV. Untuk ukuran file WAV setelah disisipkan data pada Teknik *Low-Bit Encoding* tidak mengalami perubahan, sedangkan untuk Teknik *End Of File* ukuran file WAV mengalami perubahan, yaitu bertambah dengan ukuran data.

Kata kunci : Steganografi, *Low-Bit Encoding*, *End Of File*, WAV

1. PENDAHULUAN

Pada umumnya seseorang berharap pesan yang dikirim kepada orang lain tidak dibaca oleh orang yang tidak berhak, terutama pesan yang bersifat rahasia atau pribadi yang hanya boleh diketahui oleh pihak pengirim dan pihak penerima pesan atau kalangan terbatas saja. Untuk menjaga kerahasiaan pesan yang dikirimkan tersebut diperlukan suatu teknik, salah satunya adalah menggunakan steganografi. Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui [7].

Dengan perkembangan teknologi komputasi, steganografi sudah banyak diimplementasikan pada media digital. Steganografi membutuhkan dua properti, yaitu media penampung (*cover-object*) dan data/informasi rahasia. Steganografi digital menggunakan media digital sebagai penampung, seperti citra digital, audio digital, dan video digital. Data yang disembunyikan juga berbentuk digital, seperti teks, citra digital, audio digital, dan video digital.

Banyak teknik yang dapat digunakan untuk aplikasi steganografi. Salah satu teknik yang digunakan yaitu Teknik *End Of File*. Teknik *End Of File* merupakan teknik penyembunyian data dengan menyisipkan data pada akhir file media penampung [9]. Sedangkan Teknik *Low-Bit Encoding* merupakan teknik penyembunyian data dengan memodifikasi *Least Significant Bit* (LSB) file media penampung. Teknik *Low-Bit Encoding* dilakukan dengan memodifikasi bit terakhir dari setiap *sampling point* (sampel) file media penampung dengan bit-bit data [2].

Pemilihan file audio WAVE sebagai media penampung dilatarbelakangi karena file WAVE merupakan file format standar untuk penyimpanan audio digital pada PC. Selain itu, penggunaan Sistem Operasi Windows oleh kebanyakan orang merupakan salah satu pendukung, karena file ini merupakan format utama penyimpanan audio digital pada Windows. File WAVE menyimpan audio tak termampatkan (*uncomppressed*), dalam artian file WAVE berisi langsung representasi digital dari suara asli (sinyal analog).

Penelitian menggunakan Teknik *End Of File* pernah dilakukan oleh Sukrisno dengan data yang disisipkan berupa teks [9]. Sedangkan untuk penyisipan data teks pada LSB file WAVE pernah dilakukan oleh Utami dengan metode yang digunakan yaitu menyisipkan data pada LSB byte-byte audio file WAVE [10]. Berbeda dengan penelitian yang dilakukan oleh Utami, penerapan Teknik *Low-Bit Encoding* menyisipkan data pada

LSB sampel-sampel audio file WAVE. Penerapan steganografi digital tidak hanya menggunakan data rahasia dalam bentuk teks saja. Dan sesuai kenyataan, setiap orang mempunyai berbagai macam bentuk data digital. Oleh karena itu, diperlukan penerapan steganografi menggunakan bebagai data rahasia, seperti file citra digital, audio digital, dan video digital.

Penerapan terhadap kedua teknik tersebut sangat diperlukan untuk mengetahui perbedaan yang terdapat pada masing-masing teknik. Dalam melakukan perbandingan diperlukan beberapa variabel sebagai faktor pembanding. Beberapa variabel yang digunakan diantaranya, yaitu ukuran data yang dapat disisipkan pada file media penampung dan perubahan ukuran file media penampung setelah disisipkan data [5].

Tujuan dari penelitian tugas akhir ini, yaitu : menghasilkan aplikasi steganografi pada file WAVE berformat PCM/*uncompressed* menggunakan Teknik *Low-Bit Encoding* dan Teknik *End Of File*. Serta menjelaskan perbedaan steganografi menggunakan Teknik *Low-Bit Encoding* dan dengan menggunakan Teknik *End Of File* pada file WAVE berformat PCM/*uncompressed* dengan beberapa faktor pembanding, antara lain : ukuran data yang dapat disisipkan pada file WAVE dan perubahan ukuran file WAVE setelah disisipkan data.

2. DASAR TEORI

1.1. Steganografi

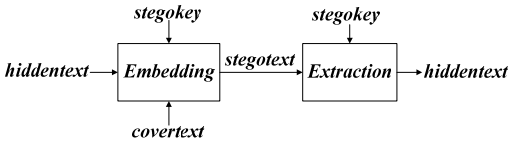
Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia sedemikian sehingga keberadaan/eksistensi pesan tidak terdeteksi oleh indera manusia [6]. Kata steganografi berasal dari Bahasa Yunani yang berarti "tulisan tersembunyi" (*covered writing*). Steganografi membutuhkan dua properti : media penampung dan pesan rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai media penampung, misalnya citra digital, audio digital, dan video digital. Pesan rahasia yang disembunyikan juga berbentuk digital, seperti file citra, audio, teks, dan video [7].

Penyembunyian pesan rahasia ke dalam media penampung mengubah kualitas media tersebut. Kriteria yang harus diperhatikan dalam penyembunyian pesan diantaranya adalah [7] :

1) *Imperceptibility*. Keberadaan pesan tidak dapat dipersepsi oleh indra manusia, baik indra pengelihatan maupun indra pendengaran.
2) *Fidelity*. Mutu media penampung tidak berubah banyak akibat penyisipan.
3) *Recovery*. Pesan yang disembunyikan harus dapat diungkap kembali.

Terdapat beberapa istilah yang berkaitan dengan steganografi [7] :

1) *Hiddentext* atau *embedded message* : pesan yang disembunyikan.
2) *Covertext* atau *cover-object* : pesan yang digunakan untuk menyembunyikan *embedded message*.
3) *Stegotext* atau *stego-object* : pesan yang sudah berisi *embedded message*.



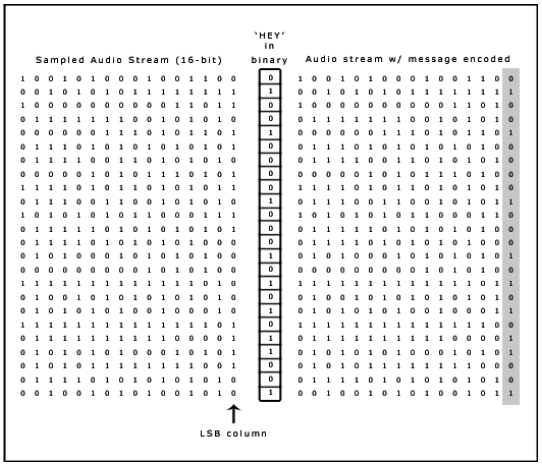
Gambar 2.1 Proses Penyisipan dan Ekstraksi Pesan

Gambar 2.1 memperlihatkan proses penyisipan dan ekstraksi pesan. Proses penyisipan pesan membutuhkan tiga input, yaitu *hiddentext*, *stegokey*, dan *covertext*. Sedangkan output yang dihasilkan dari proses penyisipan pesan berupa *stegotext*. Untuk proses ekstraksi dibutuhkan *stegotext* dan *stegokey* sebagai input. Hasil output dari proses ekstraksi adalah *hiddentext*. Penyisipan pesan ke dalam media *covertext* dinamakan *encoding* atau *embedding*, sedangkan ekstraksi pesan dari *stegotext* dinamakan *decoding* atau *extraction*. Kedua proses ini memerlukan kunci rahasia (yang dinamakan *stegokey*) agar pihak yang berhak saja yang dapat melakukan penyisipan pesan dan ekstraksi pesan.

0.0.1. Teknik *Low-Bit Encoding*

Teknik ini memodifikasi nilai yang paling kurang signifikan atau *Least Significant Bit* (LSB) dari setiap *sampling point*/sampel file media penampung [2]. Sebagai contoh 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). Bit yang memiliki signifikansi paling tinggi (MSB) adalah numerik yang memiliki nilai tertinggi (2^7 = 128), artinya jika terjadi perubahan pada bit ini akan menghasilkan perubahan yang sangat signifikan. Bit yang memiliki signifikansi paling rendah (LSB) adalah numerik yang memiliki nilai terendah (2^0 = 1), artinya jika terjadi perubahan pada bit ini akan menghasilkan perubahan yang tidak terlalu signifikan, sehingga apabila dilakukan modifikasi pada bit ini hanya menyebabkan perubahan nilai bit satu lebih tinggi atau satu lebih rendah. Misalnya pada byte 11010010, bit 1 yang pertama (digarisbawahi) adalah bit MSB dan bit 0 yang terakhir (digarisbawahi) adalah bit LSB.

Gambar 2.2 memperlihatkan penyisipan pesan pada audio digital. Teknik *Low-Bit Encoding* melakukan pengubahan nilai bit paling rendah dari sampel audio dengan nilai bit pesan yang akan disisipkan. Seperti yang terlihat pada gambar 2.2, audio digital terdiri dari sampel-sampel dengan ukuran 16 bit yang berisi informasi suara dan bit-bit pesan yang akan disisipkan pada audio digital tersebut.

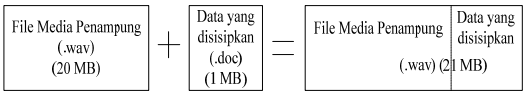


Gambar 2.2 Penyisipan LSB Audio Digital

Pesan yang disisipkan berupa kata "HEY", di mana dalam bilangan biner menjadi rentetan biner 010010000100010101011001. Karakter H (01001000), E (01000101), dan Y (01011001). Semua bit dari kata "HEY" tersebut menggantikan LSB dari tiap-tiap sampel pada media penampung, seperti terlihat pada gambar 2.2 bagan bagian kanan.

0.0.2. Teknik *End Of File*

Disebut dengan Teknik EOF karena teknik ini menyisipkan data pada akhir file media penampung. Teknik ini dapat dikatakan sebagai metode *injection*, di mana teknik ini memasukkan secara langsung data di dalam file media penampung [11]. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sesuai dengan kebutuhan. Ukuran file yang telah disisipkan data sama dengan ukuran file sebelum disisipkan data ditambah dengan ukuran data yang disisipkan ke dalam file tersebut [9]. Dalam teknik ini, data disisipkan pada akhir file dengan diberi tanda khusus sebagai pengenal *start* dari data tersebut dan pengenal akhir dari data tersebut [11]. Untuk lebih jelasnya mengenai penerapan teknik ini dapat dilihat pada gambar 2.3.



Gambar 2.3 Penyisipan Data Menggunakan Teknik *End Of File*

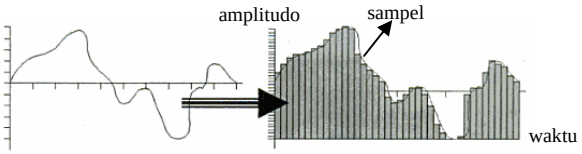
Dari gambar 2.3 terdiri atas 2 buah input, yaitu file media penampung WAVE (.wav) berukuran 20 MB dan data yang akan disisipkan berupa file dokumen (.doc) berukuran 1 MB. File dokumen disisipkan di akhir dari file WAVE di mana file dokumen tersebut melebur menjadi satu di dalam file WAVE, sehingga menghasilkan sebuah output file WAVE yang berisi file dokumen yang berukuran 21 MB (merupakan jumlah ukuran file media penampung WAVE dan ukuran data).

1.2. Audio Digital

Audio digital merupakan versi digital dari suara analog [3]. Suara adalah sesuatu yang dihasilkan oleh getaran yang berasal dari benda yang bergetar, sehingga menghasilkan gelombang di udara. Gelombang tersebut adalah gelombang analog. Untuk mengubah gelombang analog ke dalam komputer dilakukan dengan mendigitalkan gelombang analog tersebut [4].

Kualitas perekaman digital tergantung pada seberapa sampel diambil (angka *sampling* atau *sampling rate* - dihitung dalam kilohertz atau seribu sampel per detik) dan berapa banyak angka yang digunakan untuk merepresentasikan nilai dari tiap sampel (*bitdepth*, ukuran sampel, resolusi, *range* dinamis). *Sampling rate* adalah banyaknya sampel gelombang yang diambil dalam waktu satu detik. Sedangkan resolusi merupakan kuantisasi dari isi sampel berisi bit yang mewakili amplitudo.

Tiga *sampling rate* yang paling sering digunakan dalam multimedia adalah kualitas CD 44.1 kHz, 22.05 kHz, dan 11.025 kHz dengan bit resolusi 8 bit dan 16 bit. Bit resolusi 8 bit menghasilkan nilai resolusi sebesar $2^8 = 256$ untuk deskripsi resolusi atau amplitudo (level suara dalam satu waktu) atau 16 bit menghasilkan 65536 unit deskripsi resolusi [3]. Gambar 2.4 memperlihatkan sebuah hasil rekaman analog yang dikonversi menjadi audio digital.



Gambar 2.4 Gelombang suara analog menjadi digital [3]

Cara menghitung ukuran file audio digital (dalam byte) adalah sebagai berikut :

Mono : angka *sampling* * durasi *recording* dalam detik * (bit resolusi / 8) * 1

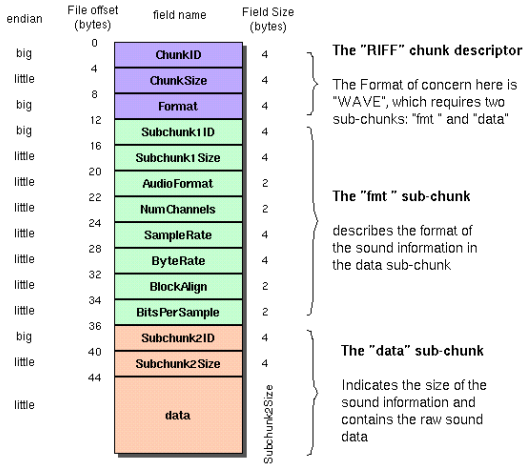
Stereo : angka *sampling* * durasi *recording* dalam detik * (bit resolusi / 8) * 2

2.2.1. WAVE

Waveform Audio File Format (WAVE, atau lebih dikenal sebagai WAV dikarenakan ekstensi filenya) merupakan file format standar untuk penyimpanan audio digital pada PC. Format ini dikembangkan oleh Microsoft dan IBM [8]. Format file WAVE sendiri merupakan sebuah sub bagian dari spesifikasi Microsoft RIFF (*Resource Interchange File Format*) yang berfungsi untuk menyimpan file-file multimedia [1].

File WAVE dapat menampung audio dalam bentuk termampatkan/terkompresi. Namun, umumnya file WAVE menyimpan audio yang tidak terkompresi. Format penyimpanan audio tidak terkompresi ini menggunakan pengkodean PCM.

PCM (*Pulse Code Modulation*) merupakan representasi digital dari suatu sinyal analog [3]. PCM menyimpan sampel dalam keadaan tidak terkompresi (mentah). Struktur file format WAVE PCM ini dapat dilihat pada gambar 2.5.



Gambar 2.5 Format File WAVE PCM [1]

RIFF mengelompokkan isi file ke dalam *chunk-chunk* terpisah [1]. File WAVE disebut juga dengan RIFF dengan *single* WAVE *chunk* yang terdiri dari dua sub-*chunk* yaitu sebuah *chunk* "fmt" yang berfungsi untuk menentukan format data audio dan sebuah *chunk* "data" yang berisi sampel data audio. Bentuk ini disebut dengan "*Canonical form*" [1], seperti terlihat pada gambar 2.5.

2.3. Studi Perbandingan

Dari hasil penelitian yang telah dilakukan Krisnawati [5] mengunakan Metode *Least Significant Bit* (LSB) dan *End Of File* (EOF) untuk menyisipkan teks ke dalam citra, terdapat beberapa faktor/hal yang diperbandingkan pada masing-masing teknik steganografi. Faktor yang dibandingkan, yaitu jumlah ukuran data/file rahasia yang dapat disisipkan pada file media penampung WAVE dan perubahan ukuran file media penampung setelah disisipkan data.

3. ANALISIS DAN PERANCANGAN

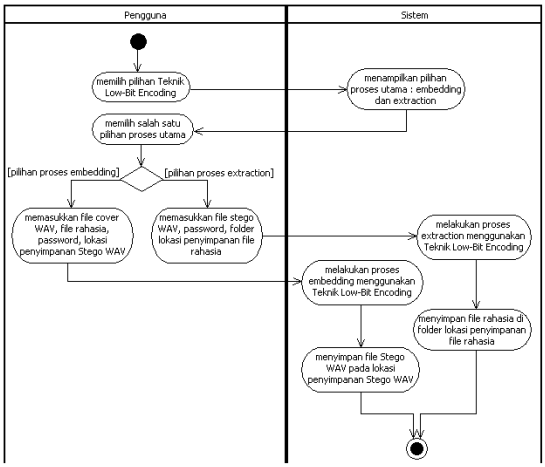
Sistem yang dikembangkan dalam tugas akhir ini berfungsi untuk mengamankan sebuah data dari ancaman orang lain yang tidak mempunyai otoritas terhadap data tersebut dengan cara menyembuyikan data tersebut ke dalam sebuah file WAV. Tidak hanya menyembunyikannya, sistem ini juga mengungkap kembali data tersebut dari dalam file WAV.

Sistem ini menggunakan dua teknik steganografi, yaitu Teknik *Low-Bit Encoding* dan Teknik *End Of File*. Masing-masing teknik terdiri atas dua proses utama, yaitu proses *embedding* (penyembunyian data ke dalam file WAV) dan proses *extraction* (mengungkap kembali data dari file WAV).

1) Teknik *Low-Bit Encoding*

Alur proses sistem untuk Teknik *Low-Bit Encoding* digambarkan dengan sebuah *activity diagram* dan ditunjukkan pada gambar 3.1. Sistem dimulai dengan pengguna memilih pilihan teknik yaitu Teknik *Low-Bit Encoding*, selanjutnya sistem menampilkan dua pilihan proses yang harus dipilih. Jika pengguna memilih pilihan proses *embedding*, maka terdapat beberapa input yang harus dimasukkan, diantaranya file WAV sebagai media penampung (file *cover* WAV), file rahasia sebagai data yang disisipkan ke dalam file *cover* WAV berupa sebuah file, *password* sebagai parameter pengaman, dan lokasi penyimpanan file *cover* WAV yang telah disisipkan data (Stego WAV). Selanjutnya sistem melakukan proses *embedding* pada file *cover* WAV menggunakan Teknik *Low-Bit Encoding*. Dari hasil proses *embedding* dihasilkan sebuah file Stego WAV yang disimpan pada lokasi penyimpanan yang telah ditentukan sebelumnya.

Sedangkan jika pengguna memilih proses *extraction*, pengguna perlu memasukkan file Stego WAV, *password* (harus sama dengan *password* saat proses *embedding*), dan *folder* lokasi peyimpanan file rahasia. Selanjutnya sistem melakukan proses *extraction* pada file Stego WAV menggunakan Teknik *Low-Bit Encoding*. Hasil *output* berupa file rahasia yang diekstrak dari file Stego WAV yang disimpan pada *folder* lokasi peyimpanan yang telah ditentukan sebelumnya.



Gambar 3.1 *Activity Diagram* Sistem Untuk Teknik *Low-Bit Encoding*

a) Proses *Embedding* Teknik *Low-Bit Encoding*

Proses *embedding* menggunakan Teknik *Low-Bit Encoding* dilakukan dengan langkah-langkah sebagai berikut :

i) Input file *cover* WAV. Baca keseluruhan sampel audio file *cover* WAV.

ii) Input *password* dan file rahasia. Dibaca keseluruhan isi MD5*Password* dan isi file rahasia. Masukkan informasi tambahan, yaitu : *mark* berupa kata "!S@1C" dialokasikan sebesar 5 byte, nama file rahasia (32 byte), ekstensi file rahasia (4 byte), dan panjang file rahasia (4 byte). Isi MD5*Password* yang dimaksud adalah *password* yang telah mengalami fungsi *hash* MD5, sehingga seberapun panjang *password* yang diinput panjangnya tetap 32 byte setelah mengalami fungsi *hash* MD5. Semua data di atas dibaca dalam bentuk byte.

iii) Dilakukan proses penyisipan data no ii) ke LSB tiap-tiap sampel audio file *cover* WAV dengan urutan penyisipan : *mark*, MD5*Password*, nama, ekstensi, panjang file rahasia, dan file rahasia tersebut. Di dalam proses ini dilakukan beberapa kali *looping* untuk dilakukan pengambilan bit-bit data no ii) yang kemudian disisipkan ke dalam sampel-sampel audio file *cover* WAV. Penyisipan dilakukan dengan mengganti LSB setiap sampel audio file *cover* WAV dengan bit-bit data no ii).

iv) Setelah semua bit-bit data telah disisipkan, semua sampel audio disimpan kembali dalam bentuk file WAV yang baru (Stego WAV).

b) Proses *Extraction* Teknik *Low-Bit Encoding*

Proses *extraction* menggunakan Teknik *Low-Bit Encoding* dilakukan dengan langkah-langkah sebagai berikut :

i) Input File Stego WAV. Baca keseluruhan sampel audio file Stego WAV.

ii) Input *password*. Dibaca keseluruhan isi MD5*Password* dalam bentuk *string*.

iii) Dilakukan proses ekstraksi data dengan mengambil LSB tiap-tiap sampel audio File Stego WAV. Pada tahap ini data yang diekstraksi hanya *mark*, MD5*Password*, nama, ekstensi, dan panjang file rahasia. Tahap ini dilakukan beberapa *looping* untuk diambil LSB tiap-tiap sampel audio yang hanya mengandung data *mark*, MD5*Password*, nama, ekstensi, dan panjang file rahasia. Bit-bit hasil ekstraksi tersebut disusun dalam bentuk byte kembali dan diubah menjadi deratan sebuah *string*. Khusus panjang file rahasia diubah dalam bentuk *integer*.

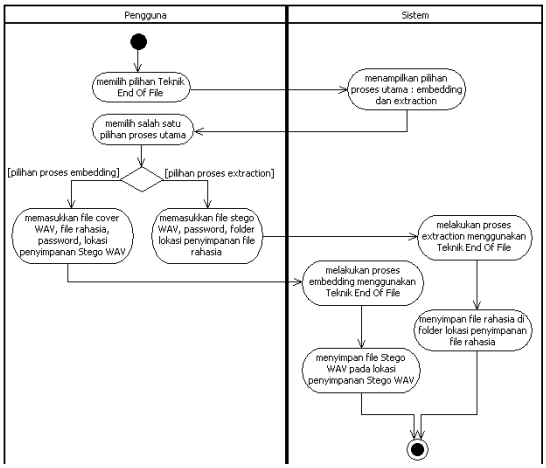
iv) *String mark* hasil ekstraksi dicocokkan. Jika *mark* sama dengan *string* "!S@1C" maka akan dilanjutkan ke tahap pencocokan *MD5Password*, yaitu tahap no. v). Namun, jika *mark* tidak sama maka ditampilkan pesan bahwa tidak terdapat file rahasia pada file Stego WAV tersebut dan proses berhenti sampai di sini.

v) *String* MD5*Password* hasil ekstraksi dicocokkan dengan *string* MD5*Password* input no. ii). Jika MD5*Password* hasil ekstraksi sama dengan MD5*Password* input maka proses berlanjut ke tahap no. vi). Jika MD5*Password* input tidak sama dengan MD5*Password* hasil ekstraksi maka ditampilkan pesan bahwa *password* input salah dan proses berhenti sampai di sini.

vi) Dilakukan proses ekstraksi file rahasia. Pada tahap ini sampel-sampel audio yang tidak diproses pada tahap no. iii) di-*looping* sebanyak ukuran file rahasia yang diperoleh dari tahap no. iii) untuk diambil LSB-nya. Bit-bit file rahasia hasil ekstraksi tersebut disusun dalam bentuk byte dan disimpan ke sebuah file baru dengan nama dan ekstensi yang diperoleh dari tahap no. iii).

2) Teknik *End Of File*

Alur proses sistem untuk Teknik *End Of File* digambarkan dengan sebuah *activity diagram* dan ditunjukkan pada gambar 3.2. Sistem dimulai dengan pengguna memilih pilihan teknik yaitu Teknik *End Of File*, selanjutnya sistem menampilkan dua pilihan proses yang harus dipilih. Jika pengguna memilih pilihan proses *embedding*, maka terdapat beberapa *input* yang harus dimasukkan, diantaranya file WAV sebagai media penampung (file *cover* WAV), file rahasia sebagai data yang disisipkan ke dalam file *cover* WAV berupa sebuah file, *password* sebagai parameter pengaman, dan lokasi penyimpanan file *cover* WAV yang telah disisipkan data (Stego WAV). Selanjutnya sistem melakukan proses *embedding* pada file *cover* WAV menggunakan Teknik *End Of File*. Dari hasil proses *embedding* dihasilkan sebuah file Stego WAV yang disimpan pada lokasi penyimpanan yang telah ditentukan sebelumnya.



Gambar 3.2 *Activity Diagram* Sistem Untuk Teknik *End Of File*

Sedangkan jika pengguna memilih proses *extraction*, pengguna perlu memasukkan file Stego WAV, *password* (harus sama dengan *password* saat proses *embedding*), dan *folder* lokasi peyimpanan file rahasia. Selanjutnya sistem melakukan proses *extraction* pada file Stego WAV menggunakan Teknik *End Of File*. Hasil *output* berupa file rahasia yang diekstrak dari file Stego WAV yang disimpan

pada *folder* lokasi peyimpanan yang telah ditentukan sebelumnya.

a) Proses *Embedding* Teknik *End Of File*

Untuk proses *embedding* menggunakan Teknik *End Of File* dilakukan dengan langkah-langkah sebagai berikut :

i) Input file *cover* WAV. Baca seluruh isi file *cover* WAV dalam bentuk byte.
ii) Input *password* dan file rahasia. Baca seluruh isi MD5*Password* (sebesar 32 byte) dan isi file rahasia. Masukkan informasi tambahan, yaitu : *mark* berupa kata "!S@1C" (dialokasikan sebesar 5 byte), nama file rahasia (dialokasikan sebesar 32 byte), ekstensi file rahasia (dialokasikan sebesar 4 byte), dan panjang file rahasia (dilalokasikan sebesar 4 byte). Semua data di atas dibaca dalam bentuk byte.
iii) Tulis byte file *cover* WAV ke sebuah file WAV yang baru (Stego WAV). Kemudian dilanjutkan dengan menulis byte-byte data dari tahap no. ii) setelahnya ke file Stego WAV dengan urutan penulisan : *mark*, MD5*Password*, nama, ekstensi, dan panjang file rahasia.

b) Proses *Extraction* Teknik *End Of File*

Proses *extraction* menggunakan Teknik *End Of File* dilakukan dengan langkah-langkah sebagai berikut :

i) Input file Stego WAV. Baca seluruh isi file Stego WAV dalam bentuk byte.
ii) Input *password*. Dibaca keseluruhan isi MD5*Password* dalam bentuk *string*.
iii) Dilakukan proses ekstraksi informasi tambahan dengan mengambil byte-byte terletak akhir Stego WAV. Byte-byte hasil ekstraksi tersebut disusun dan diubah menjadi deratan sebuah *string*. Khusus panjang file rahasia diubah dalam bentuk *integer*.
iv) *String mark* hasil ekstraksi dicocokkan. Jika *mark* sama dengan *string* "!S@1C" maka akan dilanjutkan ke tahap pencocokan MD5*Password*, yaitu tahap no. v). Namun, jika *mark* tidak sama maka ditampilkan pesan bahwa tidak terdapat file rahasia pada file Stego WAV tersebut dan proses berhenti sampai di sini.
v) *String* MD5*Password* hasil ekstraksi dicocokkan dengan *string* MD5*Password* input no. ii). Jika MD5*Password* hasil ekstraksi sama dengan MD5*Password* input maka proses berlanjut ke tahap no. vi). Jika MD5*Password* input tidak sama dengan MD5*Password* hasil ekstraksi maka ditampilkan pesan bahwa *password* input salah dan proses berhenti sampai di sini.
vi) Dilakukan proses ekstraksi file rahasia dengan mengambil sejumlah byte-byte sebelum informasi tambahan sesuai panjang file rahasia yang diperoleh dari tahap no. iii). Byte-byte file rahasia hasil ekstraksi tersebut disimpan ke sebuah file baru dengan nama dan ekstensi yang diperoleh dari tahap no. iii).

4. IMPLEMENTASI, PENGUJIAN, DAN PERBANDINGAN

4.1. Implementasi

Spesifikasi perangkat keras yang digunakan untuk membangun sistem Steganografi pada File Audio WAVE Menggunakan Teknik *Low-Bit Encoding* dengan Teknik *End Of File* adalah :

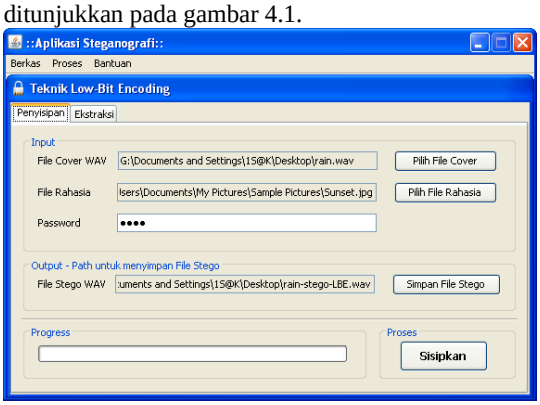
1) CPU : Intel(R) Core(TM)2 Duo CPU T5870 @ 2.00 GHz
2) RAM : 2040 MB

Sedangkan perangkat lunak yang digunakan untuk membangun sistem ini adalah sebagai berikut:

1) Sistem Operasi : Microsoft Windows XP Professional SP3
2) Bahasa Pemrograman : Java SE Versi JDK 1.7.0
3) Alat Bantu Pemrograman : Netbeans IDE 7.1

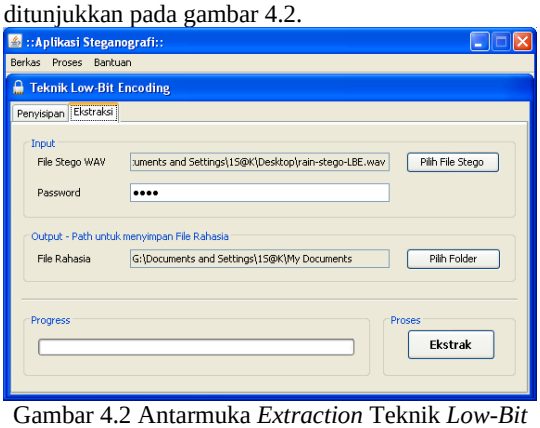
1) Teknik *Low-Bit Encoding*

Implementasi antarmuka fungsi *embedding* menggunakan Teknik *Low-Bit Encoding* ditunjukkan pada gambar 4.1.



Gambar 4.1 Antarmuka *Embedding* Teknik *Low-Bit Encoding*

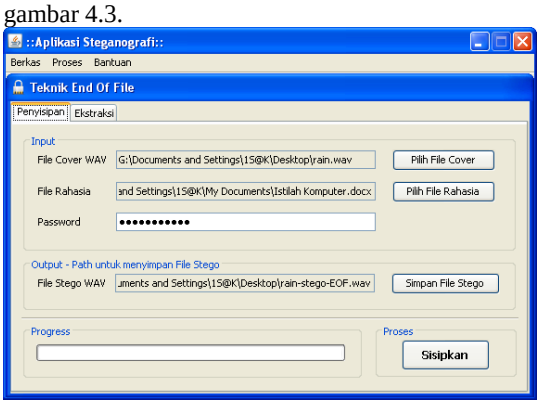
Implementasi antarmuka fungsi *extraction* menggunakan Teknik *Low-Bit Encoding* ditunjukkan pada gambar 4.2.



Gambar 4.2 Antarmuka *Extraction* Teknik *Low-Bit Encoding*

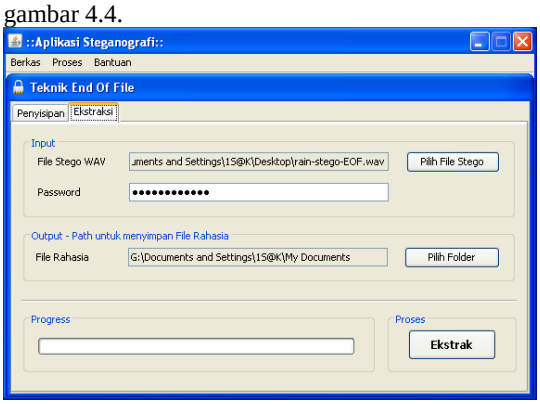
2) Teknik *End Of File*

Implementasi antarmuka fungsi *embedding* menggunakan Teknik *End Of File* ditunjukkan pada gambar 4.3.



Gambar 4.3 Antarmuka *Embedding* Teknik *End Of File*

Implementasi antarmuka fungsi *extraction* menggunakan Teknik *End Of File* ditunjukkan pada gambar 4.4.



Gambar 4.4 Antarmuka *Extraction* Teknik *End Of File*

4.2. Pengujian

Pengujian dilakukan terhadap kemampuan aplikasi dalam menyisipkan berbagai macam file rahasia, seperti file citra, suara, teks, dan video ke dalam file *cover* WAV (*embedding*). Serta untuk mengetahui kemampuan aplikasi dalam mengekstraksi file rahasia dari file Stego WAV (*extraction*). Pengujian dilakukan dengan menyisipkan file rahasia dengan berbagai ukuran dan berbagai tipe file ke dalam sebuah file *cover* WAV. Kemudian mengekstraksi file rahasia tersebut dari file Stego WAV.

File *cover* WAV yang digunakan bernama Mephisto Waltz Excerpt - 16bit 44_1kHz.wav dengan *sampling rate* 44.100Hz, resolusi 16 bit, ber-*channel stereo*, durasi 1 menit 5 detik, ukuran 10,9 MB (11.518.656 byte), dan jumlah sampel audio sebanyak 5.759.306 yang di-*download* dari http://www.lessloss.com/steinway-sons-grand-piano-recording-p-202.html. File rahasia yang digunakan dapat dilihat pada tabel 4.1. Dan *password* yang digunakan sama untuk proses *embedding* dan *extraction*.

Tabel 4.1 File Rahasia yang Digunakan Untuk Pengujian

Kriteria File	Tipe File	Nama File	Ukuran File (byte)
File Teks	.txt	license	11.141
	.docx	Pengenalan MySQL	24.521
	.pdf	rutebusway.com	86.750
File Citra	.bmp	kiloa	577.554
	.jpg	Water lilies	83.794
	.gif	crane	42.744
File Audio	.wav	Windows Logon Sound	157.412
	.mp3	on_new_direct_message	49.285
	.mid	onestop	40.075
File Video	.avi	delete	224.256
	.mp4	sample_mpeg4	245.779
	.wmv	title_trans_notes	709.220
Lainnya	.exe	hjsplit	350.720
	.zip	All_Default_File_Extentions	117.477
	.rar	surat beasiswa(format mipa)	150.249

1) Teknik *Low-Bit Encoding*

Dari hasil *embedding* menggunakan Teknik *Low-Bit Encoding* didapatkan bahwa aplikasi dapat menyisipkan semua file rahasia tabel 4.1 dengan berbagai nama file, tipe file, dan ukuran file. File Stego WAV yang dihasilkan mempunyai ukuran, *sampling rate*, resolusi, *channel*, dan durasi yang tidak berubah atau sama dengan file *cover* WAV, yaitu ukuran 10,9 MB (11.518.656 byte), *sampling rate* 44.100Hz, resolusi 16 bit, *channel stereo*, durasi 1 menit 5 detik. File Stego WAV hasil pengujian tetap dapat dibuka dan dijalankan di semua media player audio. Dan ketika dimainkan tidak terasa perubahan kualitas audio dari file Stego WAV jika dibandingkan dengan file *cover* WAV, walaupun sampel-sampel audio file Stego WAV telah mengalami perubahan.

Dari hasil *extraction* menggunakan Teknik *Low-Bit Encoding* didapatkan bahwa aplikasi dapat mengekstraksi semua file rahasia dari semua file Stego WAV yang ada. File rahasia hasil ekstraksi dapat dikembalikan seperti semula dengan nama file, tipe file, dan ukuran file yang sama dengan file rahasia pada tabel 4.1. Selain itu, file rahasia hasil ekstraksi dapat dibuka dan dibaca secara normal.

2)* Teknik *End Of File

Dari hasil *embedding* menggunakan Teknik *End Of File* didapatkan bahwa aplikasi dapat menyisipkan semua file rahasia tabel 4.1 dengan

berbagai nama file, tipe file, dan ukuran file. File Stego WAV yang dihasilkan mempunyai *sampling rate*, resolusi, *channel*, dan durasi yang tidak berubah atau sama dengan file *cover* WAV, yaitu *sampling rate* 44.100Hz, resolusi 16 bit, *channel stereo*, durasi 1 menit 5 detik. Namun, ukuran dari file Stego WAV mengalami perubahan dengan rincian sebagai berikut :

Ukuran File Stego WAV = (Ukuran File *cover* WAV) + jumlah byte informasi tambahan + jumlah byte MD5*Password* .....(4.1)

Dengan byte informasi tambahan berupa : *mark* (sebesar 5 byte), nama file rahasia (32 byte), ekstensi file rahasia (4 byte), dan panjang file rahasia (4 byte). Sedangkan untuk jumlah byte MD5*Password* sebanyak 32 byte. File Stego WAV hasil pengujian tetap dapat dibuka dan dijalankan di semua media player audio. Untuk kualitas audio dari file Stego WAV tidak mengalami perubahan karena Teknik *End Of File* tidak mengubah data/sampel audio dari file *cover* WAV.

Dari hasil *extraction* menggunakan Teknik *End Of File* didapatkan bahwa aplikasi dapat mengekstraksi semua file rahasia dari semua file Stego WAV yang ada. File rahasia hasil ekstraksi dapat dikembalikan seperti semula dengan nama file, tipe file, dan ukuran file yang sama dengan file rahasia pada tabel 4.1. Selain itu, file rahasia hasil ekstraksi dapat dibuka dan dibaca secara normal.

4.3. Perbandingan

Untuk membandingkan kedua teknik tersebut dilakukan proses pengujian yang sama. Dengan kata lain, segala macam bentuk pengujian yang dilakukan pada Teknik *Low-Bit Encoding* dilakukan juga pada Teknik *End Of File*. Sesuai subbab 2.3, terdapat dua faktor pembanding yang digunakan, yaitu ukuran file rahasia yang dapat disisipkan pada file WAV dan perubahan ukuran file WAV setelah disisipkan file rahasia. Selain itu, digunakan dua tambahan faktor pembanding lagi, yaitu tingkat perubahan kualitas audio file WAV setelah disisipkan file rahasia dan tingkat pendeteksian terdapatnya file rahasia di dalam sebuah file WAV.

Pelaksanaan perbandingan dilakukan berdasarkan faktor pembanding dengan rincian sebagai berikut :

1) Ukuran file rahasia yang dapat disisipkan pada file WAV

   Dilakukan dengan menyisipkan file rahasia dengan berbagai ukuran pada sebuah file *cover* WAV. File *cover* WAV yang digunakan sama, yaitu file WAV “Mephisto Waltz Excerpt - 16bit 44_1kHz”. File rahasia yang digunakan dapat dilihat pada tabel 4.2.

Tabel 4.2 Rincian File Rahasia Pada Pelaksanaan Perbandingan

No	Nama File	Tipe File	Ukuran File (byte)
1.	DX_800_all	.bmp	1.440.056
2.	Kalender Akademik Tahun 2012-2013	.pdf	4.098.523
3.	11.Innocent	.mp3	12.140.435
4.	Hack 2.15	.avi	33.105.294
5.	AdbeRdr1001_en_US	.exe	48.536.984
6.	ESET+Smart+Security+5.0.95+x86	.rar	63.736.842
7.	SamsungKiesSetup	.exe	77.557.544
8.	Flash8-en	.exe	113.060.248
9.	ProPsWW	.cab	149.278.843
10.	avg_free_x86_all_2011_1388a3717	.exe	173.411.176
11.	Office2003 Portable	.exe	190.403.869
12.	Nero-8.3.13.0_all_update	.exe	235.583.248
13.	CorelDRAWGraphics SuiteX4Installer_EN	.exe	312.477.872
14.	vmware 7	.rar	521.489.384
15.	John Carter (2012)	.mkv	941.515.307

Dari hasil *embedding* yang dilakukan terhadap kedua teknik diperoleh bahwa penyisipan file rahasia tabel 4.2 menggunakan Teknik *Low-Bit Encoding* mengalami kegagalan. Sembilan penyisipan pertama mengalami kegagalan karena ukuran file rahasia yang disisipkan melebihi batas ukuran file rahasia yang seharusnya disisipkan (yaitu 719.836 byte). Ukuran tersebut diperoleh dari perhitungan berikut ini :

(Jumlah sampel audio / 8) - jumlah byte informasi tambahan - jumlah byte MD5*Password*..... (4.2)

Dengan byte informasi tambahan berupa : *mark* (sebesar 5 byte), nama file rahasia (32 byte), ekstensi file rahasia (4 byte), dan panjang file rahasia (4 byte). Sedangkan untuk jumlah byte MD5*Password* sebanyak 32 byte. Untuk penyisipan file rahasia dari no.10 sampai dengan no.15 mengalami kegagalan selain karena ukuran file rahasia telah melebihi ukuran 719.836 byte dan juga dikarenakan kehabisan memori (tidak tersedianya sisa ruang memori untuk mengalokasikan file rahasia tersebut). Di mana file rahasia tersebut terlalu besar untuk dialokasikan di sisa ruang memori yang tersedia.

Sedangkan pada Teknik *End Of File*, diperoleh bahwa aplikasi dapat menyisipkan file rahasia dari no.1 sampai dengan no.9 dan sisanya mengalami kegagalan. Kegagalan tersebut diakibatkan sistem tidak dapat mengalokasikan file rahasia dari no.10 sampai dengan no.15 ke dalam memori atau sistem mengalami kehabisan memori. Di mana file rahasia tersebut terlalu besar untuk dialokasikan di sisa ruang memori yang tersedia. Sehingga, ukuran file rahasia yang dapat disisipkan menggunakan Teknik

End Of File bergantung pada tersedianya sisa ruang memori untuk mengalokasikan file rahasia

2) Perubahan ukuran file WAV setelah disisipkan file rahasia

   Dilakukan dengan mengurangi ukuran file Stego WAV dengan ukuran file *cover* WAV-nya File *cover* WAV yang digunakan yaitu file WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz". Pada pelaksanaan perbandingan ini dilakukan perhitungan pengurangan ukuran file Stego WAV pada hasil *embedding* Teknik *Low-Bit Encoding* dan hasil *embedding* Teknik *End Of File* menggunakan file rahasia tabel 4.1 dengan file *cover* WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz".

   Dari hasil perbandingan diperoleh bahwa pada Teknik *Low-Bit Encoding* ukuran file WAV setelah disisipkan file rahasia tidak berubah. Sedangkan, pada Teknik *End Of File* dihasilkan bahwa ukuran file WAV setelah disisipkan file rahasia bertambah sesuai rumus 4.1.

3) Perubahan kualitas audio file WAV setelah disisipkan file rahasia

   Dilakukan pengukuran secara kualitatif, yaitu dengan cara mendengarkan audio file Stego WAV yang kemudian dibandingkan dengan kualitas audio file *cover* WAV-nya. File *cover* WAV yang digunakan yaitu file WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz". Kualitas file Stego WAV dari hasil penyisipan file rahasia tabel 4.1 dibandingkan dengan file cover WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz".

   Dari hasil pendengaran, yaitu kualitas audio file-file Stego WAV pada Teknik *Low-Bit Encoding* tidak berbeda dengan file *cover* WAV-nya. Sama halnya dengan file-file Stego WAV pada Teknik *End Of File*. Namun hasil tersebut berbeda ketika digunakan file *cover* WAV dengan resolusi 8 bit. Sebagai contoh digunakan file *cover* WAV baru bernama Mephisto Waltz Excerpt - 8bit 44_1kHz.wav. File WAV ini berasal dari file *cover* WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz.wav" yang diturunkan resolusinya menjadi 8 bit menggunakan perangkat lunak Cool Edit Pro 2.1. Dilakukan proses *embedding* menggunakan file rahasia "delete.avi" tabel 4.1. Dari hasil perbandingan *embedding* diperoleh bahwa file Stego WAV dari hasil *embedding* menggunakan Teknik *Low-Bit Encoding* mengalami sedikit *noise*/derau di 20 detik awal ketika file Stego WAV dimainkan. Sedangkan, file Stego WAV dari hasil *embedding* menggunakan Teknik *End Of File* tidak mengalami perubahan kualitas audio.

4) Tingkat pendeteksian file rahasia di dalam sebuah file WAV

   Dilakukan dengan memperhatikan tingkat perubahan apa saja yang terjadi pada file Stego WAV yang dibandingkan dengan file *cover* WAV-nya. File *cover* WAV yang digunakan yaitu file WAV "Mephisto Waltz Excerpt - 16bit 44_1kHz". Dari hasil pengamatan file Stego WAV hasil penyisipan file rahasia tabel 4.1 diperoleh dua faktor yang dapat digunakan sebagai pembanding pendeteksian file rahasia dalam file WAV, yaitu kualitas audio dan ukuran file Stego WAV.

   Dari hasil perbandingan terhadap tingkat pendeteksian file rahasia di dalam sebuah file WAV diperoleh bahwa pada Teknik *Low-Bit Encoding* kemungkinan untuk file Stego WAV dapat dideteksi file rahasianya kecil, dikarenakan kualitas audio tetap terjaga dan ukuran file WAV setelah disisipkan file rahasia tidak berubah.

   Untuk Teknik *End Of File*, kemungkinan file Stego WAV terdeteksi file rahasianya besar, dikarenakan ukuran file Stego WAV yang berubah setelah disisipkan file rahasia walaupun kualitas audio tidak berubah/sama dengan file *cover* WAV-nya. Bagi seseorang yang paham/mengenal sangat dalam mengenai strukur file WAV, orang tersebut akan curiga ketika mendapatkan file WAV namun ukurannya tidak sesuai dengan perhitungan ukuran file audio digital yang dijelaskan pada subbab 2.2.

Dari analisis hasil perbandingan yang telah dipaparkan, dapat dibuat sebuah tabel hasil perbandingan kedua teknik tersebut yang ditunjukkan pada tabel 4.3.

Tabel 4.3 Hasil Perbandingan Teknik *Low-Bit Encoding* dan Teknik *End Of File*

Perbedaan	**Teknik *Low-Bit Encoding***	**Teknik *End Of File***
Ukuran file rahasia yang dapat disisipkan	Terbatas (jumlah sampel audio file WAV yang telah dibagi delapan kemudian dikurangi jumlah byte informasi tambahan dan byte MD5*Password*)	Terbatas (Ukuran file dapat lebih besar dari Teknik *Low-Bit Encoding* dan sesuai keinginan pengguna selama alokasi memori untuk file rahasia masih tersedia)
Ukuran file WAVsetelah disisipkan file rahasia	Tidak berubah	Berubah (Bertambah dengan ukuran file rahasia yang disisipkan, jumlah byte informasi tambahan dan MD5*Password*)

Perbedaan	**Teknik *Low-Bit Encoding***	**Teknik *End Of File***
Perubahan kualitas audio file WAV setelah disisipkan file rahasia	Untuk resolusi 8 bit mengalami sedikit penurunan kualitas audio. Sedangkan untuk resolusi 16 bit ke atas tidak mengalami perubahan kualitas audio	Tidak Berubah
Tingkat pendeteksian file rahasia di dalam sebuah file WAV	Kecil (karena kualitas audio tetap terjaga dan ukuran file WAVsetelah disisipkan file rahasia tidak berubah)	Besar (karena ukuran file WAV bertambah setelah disisipkan file rahasia)

Pada tabel 4.3 ditunjukkan bahwa masing-masing teknik mempunyai kelebihan dan kekurangan. Di mana pada Teknik *Low-Bit Encoding* mempunyai beberapa kelebihan, yaitu ukuran file WAV setelah disisipkan file rahasia tidak berubah, kualitas audio file WAV setelah disisipkan file rahasia tetap terjaga, dan tingkat pendeteksian file rahasia di dalam file WAV kecil. Namun, kekurangan yang dimiliki Teknik *Low-Bit Encoding* yaitu ukuran file rahasia yang dapat disisipkan kecil (terbatas pada jumlah sampel audio file WAV).

Sedangkan pada Teknik *End Of File* mempunyai beberapa kelebihan, yaitu ukuran file rahasia untuk disisipkan dapat disesuaikan dengan keinginan pengguna dan kualitas audio file WAV setelah disisipkan file rahasia tidak mengalami perubahan. Namun, kekurangan yang dimiliki Teknik *End Of File*, yaitu ukuran file WAV setelah disisipkan file rahasia berubah (yaitu bertambah dengan ukuran file rahasia) dan tingkat pendeteksian file rahasia di dalam file WAV besar.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian tugas akhir ini adalah :

1) Telah dihasilkan aplikasi steganografi pada file WAV menggunakan Teknik *Low-Bit Encoding* dan Teknik *End Of File*.
2) Terdapat beberapa kegagalan penyisipan dan ekstraksi file rahasia dikarenakan terjadinya *error* kehabisan memori. Hal tersebut dikarenakan tidak tersedianya sisa ruang memori untuk mengalokasikan file rahasia atau file WAV. Di mana file rahasia atau file WAV tersebut terlalu besar untuk dialokasikan di sisa ruang memori yang tersedia.
3) Berdasarkan perbandingan yang dilakukan terhadap kedua teknik, ukuran file rahasia yang dapat ditampung file WAV menggunakan Tekn*ik Low-Bit Encoding* yaitu jumlah sampel audio file WAV yang telah dibagi delapan kemudian dikurangi jumlah byte informasi tambahan dan byte M*D5Password*. Sedangkan ukuran file rahasia yang dapat ditampung file WAV menggunakan Tekn*ik End Of File* yaitu selama alokasi memori yang digunakan aplikasi saat dijalankan masih tersedia atau tidak melebihi alokasi maksimum *default java heap space* (yaitu 259.522.560 byte).
4) Berdasarkan perbandingan yang dilakukan terhadap kedua teknik, ukuran file WAV setelah disisipkan file rahasia menggunakan Teknik *Low-Bit Encoding* tidak mengalami perubahan. Namun pada Teknik *End Of File*, ukuran file WAV bertambah dengan ukuran file rahasia yang disisipkan, informasi tambahan, dan *MD5Password.*

5.2. Saran

Saran-saran dari penulis untuk pengembangan lebih lanjut penelitian ini adalah sebagai berikut :

1) Untuk menghindari terjadinya *error* kehabisan memori dalam mengalokasikan objek dapat dilakukan dengan menambahan ukuran *java heap* dengan menggunakan JVM *command line*, antara lain : -Xms atau -Xmx.
2) Atau untuk menghindari hal yang sama, dalam implementasi penyisipan dan pengekstraksian file rahasia harus diproses bagian per bagian.
3) Untuk menambah tingkat keamanan, penerapan steganografi ini dapat dipadukan dengan penerapan kriptografi. Sebagai contoh : file rahasia dienkripsi terlebih dahulu sebelum disisipkan pada file WAV.

DAFTAR PUSTAKA

[1] Bechtel Mitch, 1999-2007, "*Wave File Format*", diakses dari http://www.sonicspot.com/guide/wavefiles.html, pada tanggal 26 Maret 2012, pukul 10.24 WIB.

[2] Bender, et all, 1996, "*Techniques for data hiding*", IBM SYSTEMS JOURNAL VOL 35 NOS 3&4, diakses dari http://www.almaden.ibm.com/cs/people/dgruhl/313.pdf, pada tanggal 14 Desember 2011, pukul 23.31 WIB.

[3] Binanto Iwan, 2010, "*Multimedia Digital - Dasar Teori dan Pengembangannya*", Andi, Yogyakarta.

[4] Daryanto Tri, 2005, "*Sistem Multimedia dan Aplikasinya*", Graha Ilmu, Yogyakarta.

[5] Krisnawati, 2008, "*Metode Least Significant Bit (LSB) Dan End Of File (EOF) Untuk Menyisipkan Teks Ke Dalam Citra Grayscale*", Seminar Nasional Informatika 2008 (semnasIF 2008) ISSN: 1979-2328, STIMIK AMIKOM, Yogyakarta, diakses dari http://repository.upnyk.ac.id/64/1/5 Metode Least Significant Bit (LSB) dan End of File (EOF) Untuk Menyisipkan Pesan Teks Ke Dalam .pdf, pada tanggal 1 Maret 2012, pukul 16.40 WIB.

[6] Munir Rinaldi, 2004, "*Diktat Kuliah IF5054 Kriptografi : Steganografi dan Watermarking*", Teknik Informatika ITB, Bandung, diakses dari http://informatika.stei.itb.ac.id/~rinaldi.munir/ Kriptografi/Steganografi dan Watermarking.pdf, pada tanggal 17 September 2011, pukul 13.14 WIB.

[7] Munir Rinaldi, 2006, "*Kriptografi*", Informatika, Bandung.

[8] Rachmat Antonius, Roswanto Alphone, 2005/2006, "*IM 2023 Multimedia : Suara dan Audio*", Teknik Informatika Universitas Kristen Duta Wacana, diakses dari http://lecturer.ukdw.ac.id/anton/download/multimedia3.pdf, pada tanggal 8 Desember 2011, pukul 01.44 WIB.

[9] Sukrisno, 2007, "*Implementasi Steganografi Teknik EOF*", Seminar Nasional Teknologi 2007 (SNT 2007) ISSN : 1978 – 9777, STIMIK AMIKOM, Yogyakarta, diakses dari http://p3m.amikom.ac.id/p3m/33 - IMPLEMENTASI STEGANOGRAFI TEKNIK EOF.pdf pada tanggal 27 Juli 2011, pukul 00.01 WIB.

[10] Utami Ema, 2009, "*Pendekatan Metode Least Bit Modification Untuk Merancang Aplikasi Steganography pada File Audio Digital Tidak Terkompresi*", JURNAL DASI ISSN: 1411-3201 Vol. 10 No. 1 Maret 2009, STIMIK AMIKOM, Yogyakarta, diakses dari http://p3m.amikom.ac.id/p3m/dasi/2010/DASI Maret2009/4 - STMIK AMIKOM YOGYAKARTA - PENDEKATAN METODE LEAST BIT MODIFICATION UNTUK MERANCANG APLIKASI STEGANOGRAPHY PADA FILE AUDIO DIGITAL TIDAK TERKOMPRESI.pdf, pada tanggal 17 September 2011, pukul 13.15 WIB.

[11] Weiss Max, "*Principles of Steganography*", diakses dari http://www.math.ucsd.edu/~crypto/Projects/MaxWeiss/steganography.pdf, pada tanggal 1 Maret 2012, pukul 20.56 WIB.