

# UNIVERSITAS GUNADARMA

## FAKULTAS TEKNOLOGI INDUSTRI



## AUDIO STEGANOGRAFI

Disusun Oleh :

Nama           Yusrian Roman Arubusman  
NPM            50403770  
Jurusan        Teknik Informatika  
Pembimbing   Dr. I Wayan Simri Wicaksana, S.Si, M.Eng

Diajukan Guna Melengkapi Sebagian Syarat  
Dalam Mencapai Gelar Sarjana Strata Satu (S1)

Jakarta  
Agustus 2007

LOGO GUNADARMA ONLY  
GOLD COLOR  
EMBOSS PRINTING

## ABSTRAK

Dengan semakin populernya media digital, perhatian pada tingkat keamanan akan menjadi semakin penting. Salah satu isu penting adalah tingkat keamanan pengiriman informasi. Hal ini dapat dilakukan dengan menggunakan enkripsi atau steganography. Steganography merupakan suatu metode untuk menyisipkan potongan sebuah informasi rahasia dalam suatu objek media lain.

Dalam steganography dikenal data hiding atau data embedding yaitu penyembunyian data yang nampak sangat familiar dengan enkripsi. Namun data hiding dalam steganography dan enkripsi sangat berbeda, dimana enkripsi melakukan data hiding dengan mengubah susunan karakter dalam suatu media yang sama. Sedangkan dalam steganography, data hiding dilakukan dengan cara mengubah atau menukar beberapa informasi yang tidak terlihat penting dalam media host pembawa pesan.

Dalam proyek akhir ini, metode yang diajukan adalah penggunaan media audio mp3 sebagai data masukan media pembawa pesan rahasia. Dengan membagi media gambar data masukan dalam frame, teknik ini diharapkan dapat menyisipkan informasi rahasia ke dalam satu frame maximum sebanyak 1 bit sehingga perubahan yang terjadi tidak terlihat mencolok.

Metode proyek akhir ini membuktikan suatu teknik penyembunyian pesan rahasia dalam media audio. Hasil file keluaran yang dihasilkan oleh proyek akhir ini mengalami perubahan yang rendah, hal ini dibuktikan melalui besar rata-rata nilai Signal-to Noise Ratio sebesar 99.5 % yang artinya bahwa hanya terjadi kerusakan audio sebesar 0.5 % dalam setiap file hasil keluaran yang dibandingkan dengan file asli yang menjadi masukannya.

Kata kunci: enkripsi, data hiding, steganography

Komisi Pembimbing

No	Nama	Kedudukan
1	DIGANTI NAMA PENGUJI 1	DIGANTI JABATAN PENGUJI 1
2	DIGANTI NAMA PENGUJI 2	DIGANTI JABATAN PENGUJI 2
3	DIGANTI NAMA PENGUJI 3	DIGANTI JABATAN PENGUJI 3

**DIGANTI NAMA PENULIS**

Disetujui oleh:

**DIGANTI NAMA PEMBIMBING**

Pembimbing

**DIGANTI NAMA PENGUJI-1**

Penguji-1

**DIGANTI NAMA PENGUJI-N**

Penguji-N

Jakarta, Agustus 2007

## **Riwayat Hidup**

Nama                      Yusrian Roman Arubusman  
Tanggal Lahir            Ende-Flores, 16 Januari 1985  
Pendidikan Formil    FTI Jurusan Teknik Informatika Universitas Gunadarma 2003

## UCAPAN TERIMA KASIH

Segala puji syukur hanya ke hadirat Allah SWT, yang telah memberikan rahmat dan hidayahNya, sehingga Proyek Akhir ini dapat terselesaikan. Sholawat serta salam senantiasa kepada Nabi Muhammad SAW, keluarga, sahabat dan para pengikutnya. Proyek Akhir ini berjudul "Audio Steganography". Proyek Akhir ini dilaksanakan sebagai salah satu persyaratan guna Melengkapi sebagian syarat dalam mencapai gelar sarjana strata satu (S1).

Disadari sepenuhnya bahwa Proyek Akhir yang telah dilaksanakan ini jauh dari sempurna, meski segala daya upaya telah dicurahkan untuk penyelesaian Proyek Akhir ini secara optimal. Oleh karena itu hasil dari Proyek Akhir ini perlu dikembangkan dan disempurnakan lagi pada masa yang akan datang.

Terimakasih diucapkan kepada seluruh pihak yang telah membantu baik secara langsung maupun tak langsung terselesaikannya Proyek Akhir. Semoga hasil dari Proyek Akhir ini dapat bermanfaat bagi semua pihak.

Jakarta, 9 Agustus 2007

Penulis

# Daftar Isi

Abstrak . . . . .	iii
Lembar Pengesahan . . . . .	iv
Riwayat Hidup . . . . .	v
Ucapan Terimakasih . . . . .	vi
Daftar Isi . . . . .	viii
Daftar Gambar . . . . .	ix
<b>1 Pendahuluan</b>	<b>1</b>
1.1 Latar Belakang Masalah . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Batasan Masalah . . . . .	2
1.4 Tujuan Penulisan . . . . .	2
1.5 Sistematika Penulisan . . . . .	3
<b>2 Tinjauan Pustaka</b>	<b>4</b>
2.1 Pengertian Steganografi . . . . .	4
2.2 Sejarah Steganografi . . . . .	6
2.3 Kegunaan Steganografi . . . . .	7
2.4 Tipe Media Steganografi . . . . .	8
2.4.1 File Sistem Komputer . . . . .	8
2.4.2 Transmisi Protokol . . . . .	8
2.4.3 Penyembunyian informasi dalam dokumen text . . . . .	9
2.4.4 Penyembunyian informasi dalam File Audio . . . . .	9
2.5 Metode Steganografi . . . . .	10
2.5.1 Metode Steganografi pada teks . . . . .	10
2.5.2 Metode Steganografi pada Gambar . . . . .	13
2.5.3 Metode Steganografi pada suara . . . . .	17

<b>3 Pendekatan dan Pengembangan Aplikasi Encoding</b>	<b>20</b>
3.1 Metode Perancangan . . . . .	20
3.1.1 Prinsip Kerja (Algoritma) LSB . . . . .	20
3.1.2 Flowchart LSB . . . . .	23
3.2 Coding . . . . .	24
3.2.1 Tools Untuk Coding . . . . .	25
3.2.2 Implementasi Proses dalam Coding . . . . .	25
3.2.2.1 Implementasi Tahap Embedding . . . . .	25
3.2.2.2 Implementasi Tahap Retrieving . . . . .	26
<b>4 Pengujian dan Analisa</b>	<b>29</b>
4.1 Pengujian . . . . .	29
4.1.1 Lingkungan Uji Coba . . . . .	29
4.1.2 Skenario Uji Coba . . . . .	30
4.1.2.1 Uji Coba . . . . .	30
4.1.3 Pelaksanaan Uji Coba . . . . .	32
4.1.3.1 Pengujian Tahap Embedding . . . . .	32
4.1.3.2 Pengujian Tahap Retrieving . . . . .	35
4.2 Analisis Hasil Pengujian . . . . .	36
4.2.1 Analisa Signal to Noise . . . . .	38
4.2.1.1 Analisa SNR dengan perubahan terhadap ukuran file pesan rahasia . . . . .	38
4.2.1.2 Analisa SNR dengan perubahan terhadap panjang na- ma file pesan rahasia yang dipakai. . . . .	39
4.2.2 Analisa Ratio Data Embedding . . . . .	39
<b>5 Penutup</b>	<b>41</b>
5.1 Ringkasan dan Kesimpulan . . . . .	41
5.2 Rencana Kedepan . . . . .	42
<b>Bibliografy</b>	<b>43</b>
<b>Index</b>	<b>44</b>

# Daftar Gambar

3.1	Diagram Metode LSB . . . . .	24
4.1	Memilih file pembawa . . . . .	32
4.2	Kotak dialog open file . . . . .	33
4.3	Path lengkap file pembawa . . . . .	33
4.4	Memilih file pesan rahasia . . . . .	34
4.5	Kotak dialog save file pesan rahasia . . . . .	34
4.6	Path file pesan rahasia . . . . .	34
4.7	Menentukan direktori dan nama output file . . . . .	34
4.8	Kotak dialog save output file . . . . .	35
4.9	Tampilan tombol embed . . . . .	35
4.10	Memilih file pembawa . . . . .	36
4.11	Kotak dialog open file . . . . .	37
4.12	Path lengkap file pembawa . . . . .	37
4.13	Tampilan tombol retrieve . . . . .	37

# Daftar Tabel

4.1	Tabel uji SNR dengan ukuran file pesan rahasia yang berbeda . . . . .	38
4.2	Tabel uji SNR dengan perubahan panjang nama file . . . . .	39

# Bab 1

## Pendahuluan

### 1.1 Latar Belakang Masalah

Keamanan suatu informasi pada jaman global ini makin menjadi sebuah kebutuhan vital dalam berbagai aspek kehidupan. Suatu informasi akan memiliki nilai lebih tinggi apabila menyangkut tentang aspek-aspek keputusan bisnis, keamanan, ataupun kepentingan umum. Dimana informasi-informasi tersebut tentunya akan banyak diminati oleh berbagai pihak yang juga memiliki kepentingan di dalamnya.

Oleh karena itu, *steganography* semakin dibutuhkan guna memberikan keamanan yang maksimal dalam proses pengiriman informasi. Teknik *steganography* umum digunakan bersamaan dengan menggunakan dua media yang berbeda, dimana salah satunya berfungsi sebagai media yang berisikan informasi dan yang lain berfungsi sebagai media pembawa informasi tersebut. Penggunaan teknologi *steganography* ini diharapkan dapat membantu upaya dalam peningkatan pengamanan pengiriman informasi dan mempermudah perlindungan atas hak cipta hasil karya media elektronik.

## 1.2 Rumusan Masalah

Dalam pelaksanaan tugas akhir penelitian ini terdapat beberapa permasalahan yang menjadi titik utama pembahasan, diantaranya adalah sebagai berikut :

- Bagaimana menyisipkan suatu pesan rahasia ke dalam sebuah file multimedia agar tidak mudah diketahui oleh yang tidak berhak, tapi mudah di buka oleh yang berhak.
- Apakah terjadi perubahan dalam file multimedia hasil keluaran baik itu kualitas file maupun besar data file dan seberapa besar perubahan itu terjadi dengan dilakukan proses decoding dan encoding dalam penyisipan pesan rahasia.

## 1.3 Batasan Masalah

Agar tidak terjadi kesalahan persepsi dan tidak meluasnya pokok bahasan, maka penulis memberikan batasan-batasan masalah sebagai berikut:

- Data yang dipakai hanyalah data audio berformat mp3.
- Objek penelitian difokuskan pada kualitas file dan besar file keluaran, tetapi kecepatan dalam pemrosesan encoding dan decoding belum menjadi pokok penelitian.

## 1.4 Tujuan Penulisan

Tujuan penulisan ini adalah membuat/modifikasi program aplikasi *steganografi* menggunakan bahasa pemrograman Java yang berfungsi untuk menyimpan atau menyembunyikan berkas-berkas penting ke dalam suatu pesan biasa sehingga tidak dapat diketa-

hui oleh orang lain. Pemanfaatan pemrograman Java agar sistem dapat berjalan pada multi platform secara independen.

## 1.5 Sistematika Penulisan

Untuk memudahkan cara mempelajari dan menganalisa, penulis menguraikan isi penulisan ke dalam 4 bab, dimana tiap-tiap bab saling berhubungan satu sama lain. Bab-bab tersebut adalah :

BAB I : Pendahuluan, bab ini merupakan pengantar untuk menjelaskan latar belakang masalah, tujuan penulisan, batasan masalah, metode penulisan dan sistematika penulisan yang digunakan penulis.

BAB II : Tinjauan pustaka, bab ini berisi gambaran umum dan teori tentang steganografi baik itu pengertian tentang steganografi, sejarah, kegunaan tipe-tipe media yang digunakan sampai dengan metode-metode yang digunakan untuk membuat suatu aplikasi steganografi.

Bab III : Pendekatan dan pengembangan aplikasi, bab ini berisi tentang perancangan aplikasi dengan menggunakan flowchart, pembuatan source code program dengan menggunakan bahasa pemrograman Java dan implementasi dari coding aplikasi yang telah dibuat.

Bab IV : Membahas uji coba dari aplikasi yang digunakan apakah sesuai dengan yang diinginkan.

Bab V : Penutup, bab ini berisi tentang kesimpulan dari pembahasan pada bab-bab sebelumnya dan saran-saran yang diberikan penulis. Selain keempat bab tersebut diatas, terdapat juga lampiran yang berisi listing program dan output program.

## Bab 2

# Tinjauan Pustaka

### 2.1 Pengertian Steganografi

*Steganografi* adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya. Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia, dimana disinilah fungsi dari teknik steganography yaitu sebagai teknik penyamaran menggunakan media lain yang berbeda sehingga informasi rahasia dalam media awal tidak terlihat secara jelas [Waheed, 2000].

*Steganografi* biasanya sering disalahkaprahkan dengan kriptografi karenanya keduanya sama-sama bertujuan untuk melindungi informasi yang berharga. Perbedaan yang mendasar antara keduanya yaitu steganografi berhubungan dengan informasi tersembunyi sehingga tampak seperti tidak ada informasi tersembunyi sama sekali. Jika seseorang mengamati objek yang menyimpan informasi tersembunyi tersebut, ia tidak akan menyangka bahwa terdapat pesan rahasia dalam objek tersebut, dan karenanya ia tidak akan berusaha memecahkan informasi (dekripsi) dari objek tersebut.

Kata *steganografi* berasal dari bahasa Yunani, yaitu dari kata *Steganiē* (tersem-

bunyi) dan Graptos (tulisan). Steganografi di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra digital, audio, atau video. Teknik *Steganography* ini telah banyak digunakan dalam strategi peperangan dan pengiriman sandi rahasia sejak jaman dahulu kala. Dalam perang Dunia II, teknik steganography umum digunakan oleh tentara Jerman dalam mengirimkan pesan rahasia dari atau menuju Jerman [Simmons., 1983].

Semakin pentingnya nilai dari sebuah informasi, maka semakin berkembang pula metode-metode yang dapat digunakan untuk melakukan penyisipan informasi yang didukung pula dengan semakin berkembangnya media elektronik. Berbagai macam media elektronik kini telah dapat digunakan untuk melakukan berbagai fungsi steganography dengan berbagai macam tujuan dan fungsi yang diharapkan oleh penggunanya. Sebagai fungsi yang umum, steganography digunakan untuk memberikan cap khusus dalam sebuah karya yang dibuat dalam format media elektronik sebagai identifikasi [Johnson, 2006].

Satu hal esensial yang menjadi kelebihan steganografi adalah kemampuannya untuk menipu persepsi manusia, manusia tidak memiliki insting untuk mencurigai adanya arsip-arsip yang memiliki informasi yang tersembunyi di dalamnya, terutama bila arsip tersebut tampak seperti arsip normal lainnya. Namun begitu terbentuk pula suatu teknik yang dikenal dengan steganalysis, yaitu suatu teknik yang digunakan untuk mendeteksi penggunaan steganografi pada suatu arsip. Seorang steganalyst tidak berusaha untuk melakukan dekripsi terhadap informasi yang tersembunyi dalam suatu arsip, yang dilakukan adalah berusaha untuk menemukannya. Terdapat beberapa cara yang dapat digunakan untuk mendeteksi steganografi seperti melakukan pengamatan terhadap suatu arsip dan membandingkannya dengan salinan arsip yang dianggap belum direkayasa, atau berusaha mendengarkan dan membandingkan perbedaannya dengan arsip lain bila arsip tersebut adalah dalam bentuk audio.

## 2.2 Sejarah Steganografi

Seperti kriptografi, penggunaan *steganografi* sebetulnya telah digunakan berabad-abad yang lalu bahkan sebelum istilah *steganografi* itu sendiri muncul. Berikut adalah contoh penggunaan *steganografi* di masa lalu:

1. Selama terjadinya Perang Dunia ke-2, tinta yang tidak tampak (*invisible ink*) telah digunakan untuk menulis informasi pada lembaran kertas sehingga saat kertas tersebut jatuh di tangan pihak lain hanya akan tampak seperti lembaran kertas kosong biasa. Cairan seperti air kencing (*urine*), susu, vinegar, dan jus buah digunakan sebagai media penulisan sebab bila salah satu elemen tersebut dipanaskan, tulisan akan menggelap dan tampak melalui mata manusia.
2. Pada sejarah Yunani kuno, masyarakatnya biasa menggunakan seorang pembawa pesan sebagai perantara pengiriman pesan. Pengirim pesan tersebut akan dicukur rambutnya, untuk kemudian dituliskan suatu pesan pada kepalanya yang sudah botak. Setelah pesan dituliskan, pembawa pesan harus menunggu hingga rambutnya tumbuh kembali sebelum dapat mengirimkan pesan kepada pihak penerima. Pihak penerima kemudian akan mencukur rambut pembawa pesan tersebut untuk melihat pesan yang tersembunyi.
3. Metode lain yang digunakan oleh masyarakat Yunani kuno adalah dengan menggunakan lilin sebagai media penyembunyi pesan mereka. Pesan dituliskan pada suatu lembaran, dan lembaran tersebut akan ditutup dengan lilin untuk menyembunyikan pesan yang telah tertulis. Pihak penerima kemudian akan menghilangkan lilin dari lembaran tersebut untuk melihat pesan yang disampaikan oleh pihak pengirim.

## 2.3 Kegunaan Steganografi

Seperti perangkat keamanan lainnya, *steganografi* dapat digunakan untuk berbagai macam alasan, beberapa diantaranya untuk alasan yang baik, namun dapat juga untuk alasan yang tidak baik. Untuk tujuan legitimasi dapat digunakan pengamanan seperti citra dengan watermarking dengan alasan untuk perlindungan *copyright*. *Digital watermark* (yang juga dikenal dengan *fingerprinting*, yang dikhususkan untuk hal-hal menyangkut *copyright*) sangat mirip dengan *steganografi* karena menggunakan metode penyembunyian dalam arsip, yang muncul sebagai bagian asli dari arsip tersebut dan tidak mudah dideteksi oleh kebanyakan orang.

*Steganografi* juga dapat digunakan sebagai cara untuk membuat pengganti suatu nilai *hash* satu arah (yaitu pengguna mengambil suatu masukan panjang variabel dan membuat sebuah keluaran panjang statis dengan tipe string untuk melakukan verifikasi bahwa tidak ada perubahan yang dibuat pada variabel masukan yang asli). Selain itu juga, *steganografi* dapat digunakan sebagai tag-notes untuk citra online. Terakhir, *steganografi* juga dapat digunakan untuk melakukan perawatan atas kerahasiaan informasi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotasi, pencuri, atau dari pihak yang tidak berwenang.

Sayangnya, *steganografi* juga dapat digunakan untuk alasan yang ilegal. Sebagai contoh, jika seseorang telah mencuri data, mereka dapat menyembunyikan arsip curian tersebut ke dalam arsip lain dan mengirimkannya keluar tanpa menimbulkan kecurigaan siapapun karena tampak seperti email atau arsip normal. Selain itu, seseorang dengan hobi menyimpan pornografi, atau lebih parah lagi, menyimpannya dalam hard disk, mereka dapat menyembunyikan hobi buruk mereka tersebut melalui *steganografi*. Begitu pula dengan masalah terorisme, *steganografi* dapat digunakan oleh para teroris untuk menyamarkan komunikasi mereka dari pihak luar.

## 2.4 Tipe Media Steganografi

Dalam steganography, ada beberapa tipe media yang dapat digunakan untuk menyisipkan pesan rahasia. Tipe - tipe media ini dapat berfungsi sebagai media pembawa pesan rahasia, yang disebut dengan host message. Terdapat beberapa media dalam data digital yang dapat digunakan sebagai media steganography, diantaranya adalah File Sistem Komputer, Transmisi Protokol, Dokumen Text dan File Audio. Secara detail 4 tipe media steganography tersebut dapat dijelaskan dalam sub bab di bawah ini [Bender, 1998].

### 2.4.1 File Sistem Komputer

Sebagaimana penyimpanan data secara normal, sebuah file sistem komputer juga dapat digunakan untuk menyembunyikan informasi diantara file yang tidak terlihat penting. Sebagai contoh sebuah hard drive ketika menampakkan partisi dalam komputer pe- makai dapat berisi partisi tersembunyi yang dapat membawa informasi tersembunyi didalamnya. Sebagai contoh *sfspatch* adalah sebuah potongan kernel, yang dapat berfungsi untuk memasukkan modul pendukung file *steganography* dalam sistem Linux. *Sfspatch* menggunakan enkripsi secara bersamaan dengan tehnik *steganography* untuk menyembunyikan informasi rahasia di dalam disk sehingga tidak akan terlihat oleh pemakai awam [Westfeld, 1999].

### 2.4.2 Transmisi Protokol

Transmission Control Protocol (TCP) dan Internet Protocol (IP) adalah sebagian dari protokol yang dapat digunakan untuk menyembunyikan informasi didalam bagian header tertentu. Beberapa bagian dari TCP/IP akan diubah atau dipotong melalui mekanisme paket filter atau melalui fragment-fragment yang dikumpulkan kembali. Bagai-

manapun, terdapat beberapa bagian yang tidak dapat diubah. Bagian - bagian tersebut meliputi : Identification field, Sequence Number field, dan Acknowledge Sequence Number field.

### **2.4.3 Penyembunyian informasi dalam dokumen text**

Menurut Bender et al [Bender, 1996] softcopy text merupakan salah satu tempat yang paling menarik untuk melakukan penyembunyian data. Karena kurangnya informasi redundan di dalam data text. Bender et al mendiskusikan tiga cara berbeda untuk melakukan penyembunyian di dalam data text. Metode - metode tersebut adalah : Metode Spasi Terbuka, *Syntactic*, dan Metode *Semantic*.

### **2.4.4 Penyembunyian informasi dalam File Audio**

Bender et al mengungkapkan penyembunyian informasi dalam sinyal audio sebagai tantangan khusus. Hal ini disebabkan karena fakta bahwa *Human Auditory System* (HAS) berjalan melebihi jangkauan dinamis yang luas. Tetapi sebagaimana yang didiskusikan dalam paper masih terdapat beberapa kemungkinan untuk mengungkap sebagian "holes" yang tersedia. Terlebih dahulu untuk menyembunyikan informasi dalam data audio tidak hanya cukup untuk mengingat dari sensitivitas HAS, namun juga fakta bahwa sinyal audio berjalan diantara encoding dan decoding. Sinyal audio tersebut juga dapat berjalan melalui sebuah media penyimpan atau ditransmisikan melalui suatu media. Ketika data audio direpresentasikan secara digital, metode kuantisasi sample dan penilaian sampling temporal akan menjadi faktor penting. Beberapa tehnik dipresentasikan oleh Bender et al untuk menyembunyikan informasi melalui data audio meliputi *low-bit encoding* dan *Phase coding*.

Dalam *low-bit encoding* informasi disimpan dengan mengubah *Least Significant*

*Bit* (LSB) dari masing-masing sampling point menggunakan sebuah kode string biner. Hasil ini dalam jumlah informasi yang besar dapat diencode dalam sebuah single data audio. Sebagai contoh apabila channel tanpa suara idealnya berkapasitas 1 Kbps maka nilai bit yang diberikan sebesar 8 Kbps untuk 8Khz sequence sample. Sementara sebagai cara termudah untuk menyembunyikan informasi dalam data audio, skema low-bit encoding dapat dihancurkan dengan noise channel dan re-sampling.

Phase coding telah terbukti sebagai teknik coding yang paling efektif dalam kasus sinyal ke rasio noise. Dalam metode ini phase dari sinyal audio asli akan diubah dengan referensi phase dari informasi yang akan disembunyikan. Bender et al menemukan bahwa sebuah kapasitas channel sekitar 8 bps dapat dicapai dengan mengalokasikan 128 slot frekuensi per bit dengan background noise yang minimum. Bender et al mendiskusikan metode untuk memperbaiki sinyal audio encoding dibawah channel komunikasi yang berbeda. Bagaimanapun, fokus penelitian ini terletak pada penyembunyian informasi dalam data grafik dan pembaca yang tertarik terhadap kasus ini dapat melihat dalam "Techiques for Data Hiding" oleh Bender et al. [Hyperlink](#) kepada dokumen tersebut tersedia pada akhir dari paper ini.

## **2.5 Metode Steganografi**

Terdapat banyak metode yang digunakan dalam melakukan penyembunyian data ke dalam data lainnya. Berikut adalah penjelasan mengenai beberapa metode yang banyak digunakan dalam steganografi.

### **2.5.1 Metode Steganografi pada teks**

1. Metode Spasi Terbuka Terdapat beberapa cara untuk memanfaatkan spasi terbuka dalam data text guna menyembunyikan informasi. Metode ini dapat berhasil ka-

rena buku bacaan pada umumnya menambahkan satu spasi tambahan pada akhir baris atau diantara dua kata sehingga tidak terbaca aneh. Bagaimanapun, metode spasi terbuka hanya dapat digunakan dengan memakai ASCII (*American Standard Character Interchange*) format. Bender et al memberikan tiga metode untuk mengungkap white space dalam proses penyembunyian. Spasi terbuka antar kalimat akan menghasilkan nilai "0" apabila hanya terdapat sebuah spasi yang ditambahkan diantara kalimat tersebut. Dengan menambahkan dua spasi akan menghasilkan nilai "1". Metode ini dapat berhasil, tetapi membutuhkan data dalam jumlah besar untuk menyembunyikan sebuah informasi kecil. Dan juga terdapat banyak software word-processing yang akan secara otomatis membetulkan spasi antara kalimat, sehingga metode ini seringkali gagal. Metode spasi end-of-line (EOL) mengutarakan white space pada akhir dari masing-masing baris. Data disembunyikan menggunakan jumlah spasi yang telah ditentukan sebelumnya dari akhir untuk masing-masing kalimat. Sebagai contoh dua spasi akan menyembunyikan satu bit, empat spasi akan menyembunyikan dua bit dan delapan spasi akan menghasilkan tiga bit dan seterusnya. Teknik ini lebih baik dibandingkan metode spasi terbuka antar kalimat, karena dengan meningkatkan jumlah spasi akan dapat menyembunyikan lebih banyak data. Salah satu kekurangan dari teknik ini adalah dapat hilangnya informasi tersembunyi jika hard copy data yang diberikan

Pada akhirnya, pemerataan kanan dari text dapat digunakan pula untuk menyembunyikan informasi rahasia pada data text. Penghitungan dan pengontrolan spasi diantara kata dapat menyembunyikan informasi dalam data text yang terlihat tidak penting. Sebuah spasi antara kata akan menghasilkan nilai "0" dan dua buah spasi akan menghasilkan nilai "1". Bagaimanapun, pendekatan ini akan mempersulit untuk mengeluarkan informasi penting dari media data text tersebut karena

akan semakin tidak mungkin untuk membedakan sebuah spasi biasa dengan spasi yang berfungsi untuk penyembunyian data. Untuk mewujudkan hal ini, Bender et al menggunakan Manchester coding untuk mengelompokkan bit-bit. Sehingga "01" diinterpretasikan sebagai "1" dan "10" diinterpretasikan sebagai "0". Dimana "00" dan "11" akan dianggap sebagai null bit string.

2. Metode Syntactic Metode Syntactic sebagaimana yang telah di sarankan oleh Bender et al, mengutarakan penggunaan puntuasi dan struktur text untuk menyembunyikan informasi tanpa secara signifikan mengubah arti dari pesan pembawa. Sebagai contoh terdapat dua frase "bread, butter, and milk" dan "bread, butter and milk" secara gramatikal benar tetapi berbeda dalam penggunaan koma. Salah satu dapat digunakan secara alternatif dalam pesan text guna menginterpretasikan nilai "1" apabila salah satu metode dipakai dan nilai "0" untuk metode lain yang dipakai
3. Metode Semantic Metode Semantic menggunakan dua sinonim sebagai nilai primer atau sekunder. Nilai tersebut akan diterjemahkan kedalam biner "1" atau "0". Bender et al menggunakan sebuah contoh dimana kata "big" berfungsi sebagai primer dan "large" berfungsi sebagai sekunder. Oleh karena itu, dalam menguraikan isi sebuah pesan akan menterjemahkan atas penggunaan primer sebagai "1" dan sekunder sebagai "0". Bender et al menyebutkan masalah yang dapat muncul dengan penggunaan

metode ini adalah ketika sinonim tidak dapat digantikan karena dapat mengubah arti dari struktur kalimat. Sebagai contoh dalam memanggil seseorang dalam bahasa Inggris dengan "cool" mempunyai arti berbeda dibandingkan dengan memanggilnya "chilly".

## 2.5.2 Metode Steganografi pada Gambar

Sudah banyak metode yang digunakan untuk menyembunyikan pesan di dalam sebuah image tanpa mengubah tampilan image, sehingga pesan yang disembunyikan tidak akan terlihat. Berikut akan dibahas beberapa metode umum yang digunakan pada image steganography.

### 1. Penyisipan Least Significant Bit

Cara paling umum untuk menyembunyikan pesan adalah dengan memanfaatkan Least-Significant Bit (LSB). Walaupun banyak kekurangan pada metode ini, tetapi kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang. Metode ini membutuhkan syarat, yaitu jika dilakukan kompresi pada stego, harus digunakan format lossless compression, karena metode ini menggunakan bit-bit pada setiap pikses pada image. Jika digunakan format lossy compression, pesan rahasia yang disembunyikan dapat hilang. Jika digunakan image 24 bit color sebagai cover, sebuah bit dari masing-masing komponen Red, Green, dan Blue, dapat digunakan sehingga 3 bit dapat disimpan pada setiap piksel. Sebuah image 800 x 600 piksel dapat digunakan untuk menyembunyikan 1.440.000 bit (180.000 bytes) data rahasia. Misalnya, di bawah ini terdapat 3 piksel dari image 24 bit color :

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

jika diinginkan untuk menyembunyikan karakter A (10000001b) dihasilkan :

(00100111 11101000 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

dapat dilihat bahwa hanya 3 bit saja yang perlu diubah untuk menyembunyikan karakter A ini. Perubahan pada LSB ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan image 8 bit color sebagai cover, hanya 1 bit saja dari setiap piksel warna yang dapat dimodifikasi sehingga pemilihan image harus dilakukan dengan sangat hati-hati, karena perubahan LSB dapat menyebabkan terjadinya perubahan warna yang ditampilkan pada citra. Akan lebih baik jika image berupa image grayscale karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak LSB dari masing-masing piksel pada stego secara berurutan dan menuliskannya ke output file yang akan berisi pesan tersebut. Kekurangan dari metode modifikasi LSB ini adalah bahwa metode ini membutuhkan "tempat penyimpanan" yang relatif besar. Kekurangan lain adalah bahwa stego yang dihasilkan tidak dapat dikompres dengan format lossy compression.

## 2. **Masking dan Filtering**

Teknik masking dan filtering ini biasanya dibatasi pada image 24 bit color atau image grayscale. Metode ini mirip dengan watermark, dimana suatu image diberi tanda (marking) untuk menyembunyikan pesan rahasia. Hal ini dapat dilakukan, misalnya dengan memodifikasi luminance beberapa bagian dari image. Walaupun metode ini akan mengubah tampilan dari image, dimungkinkan untuk melakukannya dengan cara tertentu sehingga mata manusia tidak melihat perbedaannya. Karena metode ini menggunakan aspek image yang memang terlihat langsung, metode ini akan lebih "robust" terhadap kompresi (terutama lossy compression), cropping, dan beberapa image processing lain, bila dibandingkan

dengan metode modifikasi LSB.

### 3. Transformation

Metode yang lebih kompleks untuk menyembunyikan pesan pada image ini dilakukan dengan memanfaatkan Discrete Cosine Transformation (DCT) dan Wavelet Compression. DCT digunakan, terutama pada kompresi JPEG, untuk mentransformasikan blok 8x8 piksel yang berurutan dari image menjadi 64 koefisien DCT. Setiap koefisien DCT  $F(u,v)$  dari blok 8x8 piksel image  $f(x,y)$  dihitung sebagai berikut:

$$F(u, v) = \frac{1}{4}C(u)C(v) \left[ \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

di mana  $C(x) = \frac{1}{\sqrt{2}}$  saat  $x$  sama dengan 0 dan  $C(x) = 1$  saat  $x$  sama dengan 1. Setelah koefisien-koefisien diperoleh, dilakukan proses kuantisasi sebagai berikut :

$$F^2(u, v) = \left\lfloor \frac{F(u, v)}{Q(u, v)} \right\rfloor$$

dengan  $Q(u,v)$  adalah 64-elemen dari tabel kuantisasi. Sebagai contoh, berikut merupakan algoritma sederhana untuk menyembunyikan pesan di dalam image JPEG :

Walaupun image yang dikompresi dengan lossy compression akan menimbulkan kecurigaan karena perubahan LSB akan terlihat jelas, pada metode ini hal ini tidak akan terjadi karena metode ini terjadi di domain frekuensi di dalam image, bukan pada domain spasial, sehingga tidak akan ada perubahan yang ter-

---

**Example List Program 2.1** Algoritma Image JPEG
 

---

- 1: WHILE (masih ada data untuk di-embed) do
  - 2: ambil koefisien DCT selanjutnya dari cover image (DCT)
  - 3: IF koefisien < nilai treshold then
  - 4: ambil bit selanjutnya dari pesan
  - 5: ganti bit koefisien DCT dengan bit pesan tersebut
  - 6: END IF
  - 7: masukkan DCT ke stego (invers DCT)
  - 8: END WHILE
- 

lihat pada cover image. Wavelet Compression adalah salah satu cara kompresi data yang cocok digunakan untuk kompresi image, audio, dan video. Tujuannya adalah untuk menyimpan data dalam "ruang" yang sekecil mungkin dalam sebuah file, karenanya hilangnya informasi tertentu memang sudah diharapkan akan terjadi, kompresi ini merupakan contoh lossy compression. Sama seperti DCT, wavelet compression juga berbasis pada domain frekuensi. Keuntungannya, wavelet compression lebih baik dalam merepresentasikan daerah transien, contohnya image bintang pada langit malam. Artinya, elemen dari data yang transien akan direpresentasikan dalam jumlah informasi yang lebih kecil daripada yang terjadi pada transformasi lain, seperti pada DCT. Kerugiannya, wavelet compression kurang baik digunakan pada data yang bersifat periodik dan smooth. Metode yang dilakukan pada wavelet compression akan dijelaskan sebagai berikut. Pertama-tama, dilakukan wavelet transform yang akan menghasilkan koefisien sesuai dengan jumlah piksel pada image sebagai berikut

$$[W\psi f](a, b) = \frac{1}{\sqrt{|a|}} \sum_{-\infty}^{\infty} \psi \left( \frac{x-b}{a} \right) f(x) dx$$

Koefisien wavelet  $c_{jk}$  diperoleh dengan

$$c_{jk} = [W\psi f](2^{-j}, k2^{-j})$$

dimana  $a = 2^{-j}$  disebut binary dilation atau dyadic dilation, dan  $b = k2^{-j}$  disebut binary position atau dyadic position. Setelah koefien wavelet diperoleh, koefisien ini dapat dikompresi dengan mudah karena informasi terkonsentrasi secara statistik pada beberapa koefisien tertentu saja. Prinsip ini disebut dengan transform coding. Setelah itu, koefisien-koefisien tadi dikuantisasi, baru kemudian di-encode dengan entropy encoding dan/atau run length encoding. Berikut merupakan algoritma sederhana untuk menyembunyikan pesan di dalam image dengan menggunakan wavelet compression:

---

**Example List Program 2.2** Algoritma Image JPEG Wavelet Compression

---

**Input:** pesan, cover image

**Output:** Stego

- 1: WHILE (masih ada data untuk di-embed) do
  - 2: ambil koefisien wavelet selanjutnya dari cover image (wavelet transform)
  - 3: IF koefisien < nilai treshold then
  - 4: ambil bit selanjutnya dari pesan
  - 5: ganti koefisien wavelet dengan bit pesan tersebut dan kompresi
  - 6: END IF
  - 7: masukkan DCT ke stego (invers wavelet transform)
  - 8: END WHILE
- 

Proses ekstraksi pesan dengan menggunakan metode transformasi ini dilakukan dengan melakukan transformasi pada stego untuk memperoleh koefisien transformasi image. Pilih koefisien yang nilainya lebih kecil dari nilai treshold. Ekstrak bit data yang sesuai dengan koefisien ini dan tulis ke output file yang akan berisi pesan tersebut.

### 2.5.3 Metode Steganografi pada suara

Cara untuk mengaplikasikan steganografi pada file audio terdiri dari beberapa cara yang lazim digunakan dan prinsip kerja atau algoritma yang digunakan sama seperti pada metode steganografi pada gambar. Berikut adalah beberapa teknik yang digunakan:

### 1. Low Bit coding

Cara ini lazim digunakan dalam teknik digital steganografi yaitu mengganti LSB input setiap samplingnya dengan data yang dikodekan. Dengan metode ini keuntungan yang didapatkan adalah ukuran pesan yang disisipkan relative besar, namun berdampak pada hasil audio yang berkualitas kurang dengan banyaknya noise.

### 2. Phase coding

Metode kedua yang digunakan ini adalah merekayasa fasa dari sinyal masukan. Teori yang digunakan adalah dengan mensubstitusi awal fasa dari tiap awal segment dengan fasa yang telah dibuat sedemikian rupa dan merepresentasikan pesan yang disembunyikan. Fasa dari tiap awal segment ini dibuat sedemikian rupa sehingga setiap segmen masih memiliki hubungan yang berujung pada kualitas suara yang tetap terjaga. Teknik ini menghasilkan keluaran yang jauh lebih baik daripada metode pertama namun dikompensasikan dengan kerumitan dalam realisasinya.

### 3. Spread Spectrum

Metode yang ketiga adalah penyebaran spektrum. Dengan metode ini pesan dikodekan dan disebar ke setiap spectrum frekuensi yang memungkinkan. Maka dari itu akan sangat sulit bagi yang akan mencoba memecahkannya kecuali ia memiliki akses terhadap data tersebut atau dapat merekonstruksi sinyal random yang digunakan untuk menyebarkan pesan pada range frekuensi.

### 4. Echo Hiding

Metode terakhir yang sering digunakan adalah menyembunyikan pesan melalui teknik echo. Teknik menyamarkan pesan ke dalam sinyal yang membentuk echo. Kemudian pesan disembunyikan dengan bervariasi tiga parameter da-

lam echo yaitu besar amplitude awal, tingkat penurunan atenuasi, dan offset. Dengan adanya offset dari echo dan sinyal asli maka echo akan tercampur dengan sinyal aslinya, karena sistem pendengaran manusia yang tidak memisahkan antara echo dan sinyal asli.

Keempat metode di atas memiliki kesamaan yaitu menggunakan kelemahan dari sistem pendengaran manusia. Maka dari itu teknik steganografi dalam MP3 juga akan menggunakan kelemahan ini untuk menyembunyikan pesan.

## **Bab 3**

# **Pendekatan dan Pengembangan Aplikasi Encoding**

### **3.1 Metode Perancangan**

Pada bab ini dibahas tentang Metode perancangan secara lengkap dengan prinsip kerjanya dan flowchart aplikasi proyek akhir yang dibuat.

#### **3.1.1 Prinsip Kerja (Algoritma) LSB**

Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen suatu data dengan bit-bit data rahasia. Salah satu metode penyembunyian data yang sederhana adalah LSB.

Perhatikan contoh sebuah susunan bit pada sebuah byte:

11010010

MSB LSB

LSB = Least Significant Bit

MSB = Most Significant Bit

Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan byte tersebut menyatakan warna keabuan tertentu, maka perubahan satu bit LSB tidak mengubah warna keabuan tersebut secara berarti. Lagipula, mata manusia tidak dapat membedakan perubahan yang kecil.

Misalkan segmen data sebelum perubahan:

0 0 1 1 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 1 1 1

Segmen data setelah '0 1 1 1' disembunyikan:

0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 0 1 1 1 1

Untuk memperkuat teknik penyembunyian data, bit-bit data rahasia tidak digunakan mengganti byte-byte yang berurutan, namun dipilih susunan byte secara acak. Misalnya jika terdapat 50 byte dan 6 bit data yang akan disembunyikan, maka byte yang diganti bit LSB-nya dipilih secara acak, misalkan byte nomor 36, 5, 21, 10, 18, 49.

Bilangan acak dibangkitkan dengan *pseudo-random-number-generator* (PRNG) kriptografi. PRNG kriptografi sebenarnya adalah algoritma kriptografi yang digunakan untuk enkripsi. PRNG dibangun dengan algoritma DES (Data Encryption Standard), algoritma hash MD5, dan mode kriptografi CFB (Cipher-Feedback Mode). Tujuan dari enkripsi adalah menghasilkan sekumpulan bilangan acak yang sama untuk setiap kunci enkripsi yang sama. Bilangan acak dihasilkan dengan cara memilih bit-bit dari sebuah blok data hasil enkripsi

Pengkode LSB watermark biasanya memilih sebuah subset dari seluruh host suara yang mungkin dengan menggunakan sebuah kunci rahasia. Operasi substitusi pada LSB dilakukan di subset tersebut.

Proses ekstraksi dilakukan dengan membaca bit-bit yang diterima dari aliran bit suara yang diterima. Alat penerjemah memerlukan semua bagian dari data suara yang digunakan selama proses penempelan pesan rahasia.

Metode pengkodean LSB standar dengan mudah mengganti bit pada suara asli pada lapisan ke- $i$  dengan bit dari aliran bit data rahasia.

Algoritma LSB yang digunakan harus melakukan penempelan bit yang menimbulkan distorsi yang minimal pada suara yang ditempel.

Salah satu algoritma yang bisa digunakan adalah sebagai berikut:

---

**Example List Program 3.1** Algoritma LSB

---

```

1: IF (host sample a = 0
2: IF bit 0 is to be embedded
3: IF (ai-1)=0 THEN (ai-1ai-2a0)=111
4: IF (ai-1)=1 THEN (ai-1ai-2a0)=000
5: AND IF (ai+1)=0 THEN ai+1=1
6: ELSE
7: IF ai+2=0 THEN ai+2=1
8: ELSE IF a15=0 THEN a15=1
9: ELSE IF bit 1 is to be embedded
10: IF ai-1=1 THEN ai-1ai-2a0=000
11: IF ai-1=0 THEN ai-1ai-2a0=111
12: AND
13: IF ai+1=0 THEN ai+1=1
14: ELSE IF ai+2=0 THEN ai+2=1
15: ELSE IF a15=0 THEN a15=1
16: IF host sample a<0
17: IF bit 0 is to be embedded
18: IF ai-1=0 THEN ai-1ai-2a0=11...1
19: IF ai-1=1 THEN ai-1ai-2a0=000
20: AND
21: IF ai+1=1 THEN ai+1=0
22: ELSE IF ai+2=1 THEN ai+2=0
23: ELSE IF a15=1 THEN a15=0
24: ELSE IF bit 1 is to be embedded
25: IF ai-1=1 THEN ai-1ai-2a0=000
26: IF ai-1=0 THEN ai-1ai-2a0=11...1 AND ifai+1=1 THEN ai+1=0
27: ELSE IF ai+2=1 THEN ai+2=0
28: ELSE IF a15=1 THEN a15=0

```

---

### Ukuran Data Yang Disembunyikan

Ukuran data yang akan disembunyikan bergantung pada ukuran data penampung. Pada citra 8-bit yang berukuran 256 256 pixel terdapat 65536 pixel, setiap pixel berukuran 1 byte. Setelah diubah menjadi citra 24-bit, ukuran data bitmap menjadi  $65536 \times 3 = 196608$  byte. Karena setiap byte hanya bisa menyembunyikan satu bit di LSB-nya, maka ukuran data yang akan disembunyikan di dalam citra maksimum  $196608/8 = 24576$  byte. Ukuran data ini harus dikurangi dengan panjang nama berkas, karena penyembunyian data rahasia tidak hanya menyembunyikan isi data tersebut, tetapi juga nama berkasnya.

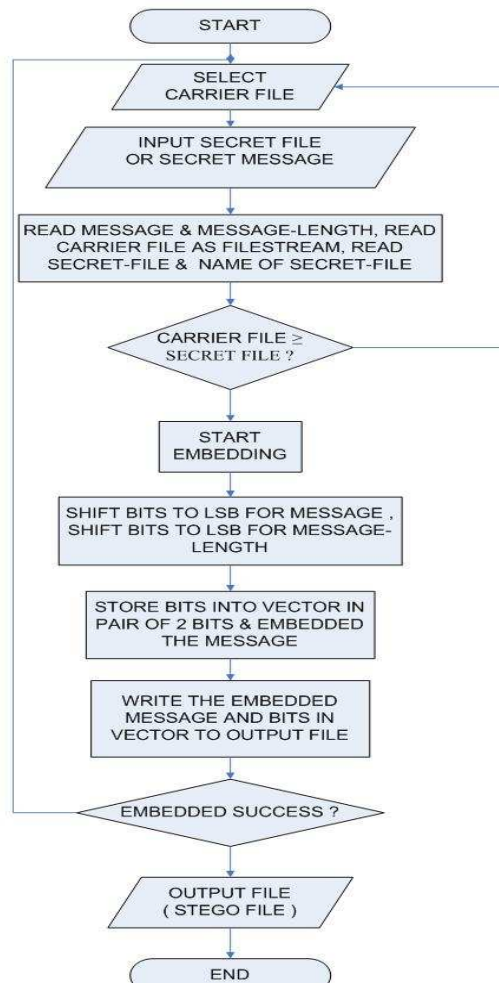
Semakin besar data disembunyikan di dalam citra, semakin besar pula kemungkinan data tersebut rusak akibat manipulasi pada citra penampung.

### Teknik Pengungkapan Data

Data yang disembunyikan di dalam citra dapat dibaca kembali dengan cara pengungkapan (reveal atau extraction). Posisi byte yang menyimpan bit data dapat diketahui dari bilangan acak yang dibangkitkan oleh PRNG. Karena algoritma kriptografi yang digunakan menggunakan kunci pada proses enkripsi, maka kunci yang sama digunakan untuk membangkitkan bilangan acak. Bilangan acak yang dihasilkan sama dengan bilangan acak yang dipakai pada waktu penyembunyian data. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali.

### 3.1.2 Flowchart LSB

Flowchart yang ditampilkan di bawah ini merupakan flowchart dari metode LSB standar untuk menyisipkan suatu data rahasia ke dalam data lain.



Gambar 3.1: Diagram Metode LSB

### 3.2 Coding

Pada bagian ini, ditampilkan terdapat beberapa pembahasan mengenai tool-tool yang digunakan untuk membuat dan menjalankan kode sumber dari perangkat lunak yang akan dibuat dan implementasi proses yang utama dalam coding tersebut.

### 3.2.1 Tools Untuk Coding

Ada beberapa tools atau perangkat lunak yang dibutuhkan untuk membuat dan menjalankan kode sumber sehingga didapat sebuah perangkat lunak yang mampu melakukan proses steganografi. Tools tersebut adalah sebagai berikut:

- Text Editor

Tool ini digunakan oleh penulis untuk memudahkan dalam penulisan kode sumber dari perangkat lunak yang akan dibuat. Tool text editor yang penulis gunakan adalah PSPad Editor yaitu program editor yang bersifat free untuk mengubah isi suatu file teks.

- Java J2SDK

Perangkat lunak yang satu ini sangat dibutuhkan dalam pengkompilasian kode sumber Java hingga menjalankan kode sumber yang telah menjadi native code tersebut agar dapat menampilkan hasilnya berupa sebuah perangkat lunak.

### 3.2.2 Implementasi Proses dalam Coding

Sesuai dengan algoritma dan flowchart dari metode LSB di atas, dibuat tahap-tahap implementasi proses yang dibutuhkan sehingga tercapai tujuan yang diharapkan. Tahap ini dibagi dalam dua bagian yaitu tahap penyisipan informasi yang disebut tahap embedding dan tahap pengambilan informasi yang disebut tahap retrieving.

#### 3.2.2.1 Implementasi Tahap Embedding

Untuk membentuk tahap embedding diperlukan dua fungsi utama sehingga data yang diperoleh dapat disisipkan dalam file pembawa. Dua fungsi tersebut di antaranya adalah :

- Fungsi `encodeMessage` Digunakan untuk membaca karakter pesan rahasia dalam sebuah kotak teks sehingga karakter dari pesan tersebut nantinya diubah menjadi bit-bit yang nantinya dapat di-embed ke dalam file pembawa.

---

**Example List Program 3.2** Coding Encode Message
 

---

```

1: //Digunakan untuk membaca ukuran panjang karakter dalam pesan
2: for(i=14; i>=0; i-=2)
3: temp= messageSize;
4: temp>>=i;
5: by= (byte) temp;
6: by&= 0x03;
7: byt= in.readByte();
8: byt&= 0xFC;
9: bytl= by;
10: out.writeByte(byt);
11: for(i=0; i<messageSize; i++)
12: byt= (byte) message.charAt(i);
13: byt&= 0x7F;
14: for(j=6; j>=0; j-=2)
15: by= byt;
16: by>>= j;
17: by&= 0x03;
18: byb= in.readByte(); //tulis ke output file
19: byb&= 0xFC;
20: bybl= by;
21: out.writeByte(byb);

```

---

- Fungsi `encodeFile` Digunakan untuk membaca file pesan rahasia sehingga file tersebut nantinya diubah menjadi bit-bit yang nantinya dapat di-embed ke dalam file pembawa.

### 3.2.2.2 Implementasi Tahap Retrieving

Untuk membentuk tahap un hiding diperlukan fungsi yang berguna untuk mengambil kembali data yang telah disisipkan pada proses embedding di atas dalam keadaan utuh.

- Fungsi `decodeMessage` Fungsi ini berguna untuk mengambil isi pesan yang disisipkan ke dalam sebuah file pembawa dan ditampilkan ke dalam kotak teks.

---

**Example List Program 3.3** Coding Encode File

---

```
1: //SISIPKAN UKURAN NAMA FILE
2: for(i=6; i>=0; i-=2)
3: temp= messageSize;
4: temp>>=i;
5: by= (byte) temp;
6: by&= 0x03;
7: byt= in.readByte();
8: byt&= 0xFC;
9: byt|= by;
10: out.writeByte(byt);
11: //SISIPKAN PESAN DALAM FILE
12: for(i=0; i<messageSize; i++)
13: byt= (byte) message.charAt(i);
14: byt&= 0x7F;
15: for(j=6; j>=0; j-=2)
16: by= byt;
17: by>>= j;
18: by&= 0x03;
19: byb= in.readByte(); //tulis ke output file
20: byb&= 0xFC;
21: byb|= by;
22: out.writeByte(byb);
```

---

---

**Example List Program 3.4** Coding Decode Message

---

```
1: //BACA UKURAN PESAN
2: for(i=14; i>=0; i-=2)
3: by= in.readByte();
4: temp= (short) by;
5: temp&= 0x0003;
6: temp<<= i;
7: messageSize= temp;
8: //BACA ISI PESAN YANG DISISIPKAN
9: for(i=0; i<messageSize; i++)
10: by= 0;
11: for(j=6; j>=0; j-=2)
12: byt= in.readByte();
13: byt&= 0x03;
14: byt<<= j;
15: by|= byt;
16: mesg[i]= (char) (((char) by)& 0x007F);
```

---

- decodeFile

Selanjutnya fungsi ini berguna untuk mengambil file beserta isi file pesan rahasia yang disisipkan ke dalam sebuah file pembawa dan ditulis ke dalam harddisk.

---

**Example List Program 3.5** Coding Decode File

---

```
1: //BACA UKURAN FILE PESAN RAHASIA
2: for(i=6; i>=0; i-=2)
3: by= in.readByte();
4: by&= 0x03;
5: temp= (short) by;
6: temp<<= i;
7: messageSize= temp;
8: //BACA ISI FILE PESAN RAHASIA
9: fileName= new char[messageSize];
10: for(i=0; i<messageSize; i++)
11: by= 0;
12: for(j=6; j>=0; j-=2)
13: byt= in.readByte();
14: byt&= 0x03;
15: byt<<= j;
16: byl= byt;
17: by&= 0x7F;
18: fileName[i]= (char) by;
```

---

# **Bab 4**

## **Pengujian dan Analisa**

### **4.1 Pengujian**

Pengujian yang dilakukan pada perangkat lunak ini ditujukan untuk mengetahui tingkat keberhasilan suatu proyek akhir dalam mencapai hasil yang diinginkan.

#### **4.1.1 Lingkungan Uji Coba**

Lingkungan uji coba sistem informasi ini dilakukan dalam lingkungan dengan karakteristik spesifikasi sebagai berikut:

- Intel Pentium 4A, 2.26 GHz
- DDR SDRAM 512MB
- HDD Seagate 200GB, 7200RPM, P-ATA 100
- VGA NVIDIA GeForce 7600 GT 256 MB
- LG DVDROM 16x
- Realtek RTL8139810x Fast Family Ethernet NIC

- ATI Chipset R410

### 4.1.2 Skenario Uji Coba

Sebelum melakukan uji coba, tujuan dan alur uji coba dirancang terlebih dahulu melalui pembuatan skenario uji coba. Sehingga pelaksanaan uji coba dapat menjawab berbagai macam batasan masalah dan tujuan dari sistem informasi proyek akhir yang dibuat. Tahap-tahap dalam skenario uji coba tersebut adalah sebagai berikut :

#### 4.1.2.1 Uji Coba

Uji coba diharapkan dapat memberikan jawaban atas kebenaran dari berbagai macam batasan, teori, ataupun analisa yang hendak dilakukan oleh peneliti. Diantaranya adalah sebagai berikut :

1. Keberhasilan penyisipan teks dalam media mp3.

Untuk menilai keberhasilan perangkat lunak terhadap penyisipan teks yang dilakukan maka dibuatlah beberapa pengujian untuk menyisipkan sebuah pesan rahasia ke dalam file mp3. Pengujian ini menggunakan beberapa mp3 sebagai carrier dan file teks yang disisipkan pada file mp3 tersebut. Kemudian diperhatikan terhadap hasil keluaran mp3 dan hasil pengambilan data yang terjadi. Sehingga prosentase dari keberhasilan penyisipan yang dibandingkan dengan jumlah percobaan dapat diketahui.

2. Analisa ratio data embedding terhadap hasil keluaran.

Dalam menguji analisa ratio data embedding terhadap file keluaran yang dihasilkan maka digunakanlah suatu rumus standard dalam mencari ratio data embe-

ding terhadap suatu file keluaran, yaitu

$$ratiodataembedding = \left( \frac{jumlahbityangdisisipkan}{jumlahahtotalbitfilepembawa} \right) bit$$

Dimana jumlah bit yang disipkan merupakan nilai dari jumlah data bit maksimal yang dapat dsisipkan dalam suatu file oleh perangkat lunak yang dibuat. Sedangkan jumlah total bit file pembawa merupakan jumlah maksimal bit yang terdapat dalam file yang digunakan. Satuan dari ratio data embedding ini adalah bit.

3. Signal-to Noise Ratio terhadap file mp3 awal yang dibandingkan dengan file mp3 hasil keluaran.

Analisa dengan menggunakan Signal-to Noise Ratio adalah penting untuk mencari tahu seberapa persen error dan kerusakan yang terjadi dalam suatu file hasil keluaran yang dibandingkan terhadap file masukan. Secara matematis, pencarian nilai signal-to noise ratio ini dapat dirumuskan sebagai berikut :

$$SNR = \left( \frac{jumlahTotalBitFileMp3 - jumlahBitTerubah}{jumlahTotalBitFileMp3} \right) * 100(\%)$$

Dimana jumlah total bit file mp3 adalah nilai total dari jumlah bit yang terdapat dalam suatu file mp3 masukan. Sedangkan jumlah bit terubah merupakan total dari jumlah bit yang berubah nilainya dari nilai bit file mp3 awal. Satuan dari nilai signal-to noise ratio ini adalah %(persen), yang menyatakan seberapa besar kesesuaian nilai file mp3 hasil keluaran terhadap file mp3 masukan.

### 4.1.3 Pelaksanaan Uji Coba

Pelaksanaan uji coba dilakukan guna mencari hasil dan menjawab berbagai macam teori dan analisa yang hendak dibuktikan oleh penyusun. Secara garis besar, pelaksanaan uji coba dibedakan dalam dua tahap yaitu tahap embedding dan tahap retrieving.

#### 4.1.3.1 Pengujian Tahap Embedding

Dalam tahap ini, pengujian hanya dilakukan dalam lingkup proses penyisipan data. Beberapa masukan diperlukan untuk memulai proses penyisipan data. Masukan yang diperlukan diantaranya adalah sebagai berikut :

1. Pengambilan Data Carier

Pengambilan data audio yang akan berfungsi sebagai pembawa pesan rahasia dapat dilakukan dengan menggunakan tombol Browse dalam bagian Input file. Dengan menekan tombol ini, proses akan menampilkan kotak untuk memilih file yang akan digunakan sebagai file carrier/ file pembawa. Kemudian file tersebut akan dibaca oleh fungsi inputstream untuk membaca offset, panjang nama file dan isi file tersebut dan kemudian nilainya diubah menjadi bit.



Gambar 4.1: Memilih file pembawa

Ketika menekan tombol browse untuk memilih file pembawa, akan muncul kotak dialog **open** sehingga memudahkan untuk memilih file pembawa yang diinginkan.

Setelah memilih file pada kotak dialog di atas, pada input file akan ditampilkan path lengkap yang merujuk pada file pembawa tersebut.



Gambar 4.2: Kotak dialog open file



Gambar 4.3: Path lengkap file pembawa

## 2. Pengambilan Data Pesan Rahasia

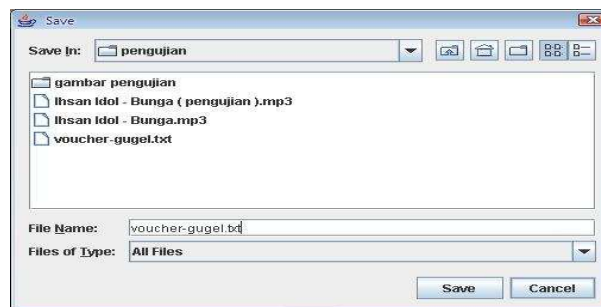
Pengambilan data pesan rahasia sama halnya dengan pengambilan data pembawa, yaitu dengan menekan tombol browse pada bagian data file. Pada saat menekan tombol ini, proses akan memanggil fungsi `showSaveDialog` yang kemudian akan dilanjutkan dengan memanggil fungsi-fungsi lain yang terdapat pada class `steganograph` untuk membaca panjang nama file dan juga isi dari file tersebut.

Pada saat menekan tombol browse tersebut untuk memilih file pesan rahasia,



Gambar 4.4: Memilih file pesan rahasia

akan muncul kotak dialog **save** dan string path dari file pesan rahasia tersebut.



Gambar 4.5: Kotak dialog save file pesan rahasia



Gambar 4.6: Path file pesan rahasia

### 3. Hasil Keluaran Tahap Embedding

Untuk mendapatkan hasil keluaran tahap embedding, sebelumnya dengan menentukan path/direktori dan nama file yang akan digunakan sebagai file keluaran tersebut dengan menekan tombol browse pada output file.



Gambar 4.7: Menentukan direktori dan nama output file

Setelah menekan tombol browse untuk menentukan direktori dan nama file yang

akan digunakan sebagai file keluaran, akan muncul kotak dialog save file sehingga memudahkan untuk memasukkan direktori dan nama file yang diinginkan.



Gambar 4.8: Kotak dialog save output file

Setelah file pembawa, file pesan rahasia dan file keluaran telah ditentukan, maka untuk melakukan proses penyisipan pesan rahasia ialah dengan memilih tombol **Embed** pada tampilan aplikasi. Tombol tersebut akan mengaktifkan class steganograph yang akan melakukan proses penyisipan pesan rahasia tersebut dengan menggunakan metode LSB.



Gambar 4.9: Tampilan tombol embed

#### 4.1.3.2 Pengujian Tahap Retrieving

Dalam tahap ini, pengujian hanya dilakukan dalam lingkup proses pengembalian data. Beberapa masukan diperlukan untuk memulai proses pengambilan data. Masukan yang diperlukan diantaranya adalah sebagai berikut :

### 1. Pengambilan Data Carier

Pengambilan data pembawa pesan rahasia dapat dilakukan dengan menggunakan tombol Browse dalam bagian Input File. Dengan menekan tombol ini, proses akan membaca data pembawa yang ditunjuk sesuai dengan nama path yang ditunjuk dalam kotak edit box dan menampilkan nama path secara lengkap pada bagian input file tersebut.



Gambar 4.10: Memilih file pembawa

Setelah memilih file pembawa pesan rahasia dengan menekan tombol Browse, maka akan ditampilkan kotak dialog **open file** dan juga string path alamat file pembawa pesan rahasia dalam kotak edit box Path, seperti yang ditunjukkan pada gambar di bawah ini.

### 2. Hasil Keluaran Tahap Retrieving

Untuk mendapatkan hasil keluaran tahap retrieving, ialah dengan memilih tombol **Retrieve**. Setelah memilih tombol **Retrieve** maka proses akan mengambil file pesan rahasia yang disisipkan dalam file pembawa pesan tersebut dan menyimpan file tersebut ke dalam harddisk komputer pada direktori yang sama dengan direktori asal aplikasi dijalankan.

## 4.2 Analisis Hasil Pengujian

Pada bagian ini dibahas mengenai analisa terhadap hasil keluaran yang dihasilkan oleh perangkat lunak proyek akhir yang dibuat. Informasi yang digunakan untuk analisa yaitu jumlah besar bit yang dipakai dan bit yang diubah. Analisa hasil keluaran yang



Gambar 4.11: Kotak dialog open file



Gambar 4.12: Path lengkap file pembawa



Gambar 4.13: Tampilan tombol retrieve

dilakukan dibagi menjadi dua yaitu uji hasil keluaran dengan signal-to noise ratio yang dipakai dan ratio data embedding dalam sebuah file audio terhadap pesan rahasia yang disisipkan.

### 4.2.1 Analisa Signal to Noise

Analisa dengan menggunakan signal-to noise ratio (SNR) terhadap sebuah file hasil keluaran yang di dalamnya terdapat file pesan rahasia yang dibandingkan dengan file pembawa yang asli dilakukan untuk mengetahui seberapa besar terjadinya perubahan dan prosentase error dari file pembawa awal terhadap file hasil keluaran. Dari analisa signal-to noise ratio ini didapatkan prosentase error yang terjadi dalam file hasil keluaran.

Dalam pengujian analisa terhadap SNR, digunakan sebuah file pembawa dan beberapa file pesan rahasia . Pengujian pertama akan dilakukan dengan sebuah file pembawa dan sebuah file pesan rahasia sebanyak empat kali dengan perubahan ukuran pada file pesan rahasia. Sedangkan pada pengujian kedua menggunakan sebuah file pembawa dan sebuah file pesan rahasia yang memiliki nama file dengan jumlah karakter beda.

#### 4.2.1.1 Analisa SNR dengan perubahan terhadap ukuran file pesan rahasia

Uji analisa ini menggunakan sebuah file pembawa, sebuah file pesan rahasia dan empat macam masukan pesan rahasia yang berbeda sehingga dapat diketahui validasi hasil yang tercapai. Tabel 4.1 menunjukkan nilai dari SNR yang diperoleh.

Tabel 4.1: Tabel uji SNR dengan ukuran file pesan rahasia yang berbeda

Carrier	Size (bit)	Secret File	Size (bit)	Output	SNR	Retrieved
Bunga.mp3	26525696	Pesan.txt	138.168	Bunga-uji1.mp3	99.99947	Pesan.txt
Bunga.mp3	26525696	Pesan.txt	147.4	Bunga-uji2.mp3	99.99944	Pesan.txt
Bunga.mp3	26525696	Pesan.txt	152.824	Bunga-uji3.mp3	99.99942	Pesan.txt
Bunga.mp3	26525696	Pesan.txt	161.28	Bunga-uji4.mp3	99.99939	Pesan.txt

Dengan memperhatikan hasil yang telah didapat dari tabel diatas, keberhasilan perangkat lunak dalam menyisipkan dan mengembalikan data adalah sebesar :

$$= \frac{(\text{Jumlah percobaan yang dilakukan} - \text{jumlah percobaan gagal})}{\text{Jumlah percobaan yang dilakukan}} * 100\%$$

$$= \frac{4 - 0}{4} * 100 = \frac{4}{4} * 100 = 100(\%).$$

Yang artinya bahwa dari semua pengujian yang telah penulis lakukan, tingkat keberhasilannya adalah 100 %.

#### 4.2.1.2 Analisa SNR dengan perubahan terhadap panjang nama file pesan rahasia yang dipakai.

Uji analisa ini menggunakan sebuah file pesan rahasia dan setiap masing-masing file pembawa diuji dengan 3 buah file pesan rahasia dengan panjang nama file yang berbeda.

Tabel 4.2: Tabel uji SNR dengan perubahan panjang nama file

Carrier	Size (bit)	Secret File	Size (bit)	Output	SNR	Retrieved
Bunga.mp3	26525696	Pesan-satu.txt	139.239	Bunga-uji1.mp3	99.99945	Pesan-satu.txt
Bunga.mp3	26525696	Pesan-kedua.txt	148.2	Bunga-uji2.mp3	99.99942	Pesan-kedua.txt
Bunga.mp3	26525696	Pesan-ketiga.txt	154.351	Bunga-uji3.mp3	99.99939	Pesan-ketiga.txt

#### 4.2.2 Analisa Ratio Data Embedding

Analisa dengan menggunakan ratio data embedding terhadap sebuah file hasil keluaran merupakan perbandingan antara jumlah total bit yang terdapat dalam sebuah file hasil keluaran dengan jumlah total bit yang disisipkan dalam file tersebut.

Dengan mengetahui jumlah total dari sebuah file hasil keluaran pada tabel 4.1 ten-

tang analisa uji SNR terhadap perubahan ukuran file pesan rahasia di atas yaitu sebesar 26525696 bit. Sedangkan jumlah total isi dari file pesan rahasia dengan ukuran maksimalnya adalah sebesar 161.28 bit. Maka dengan menggunakan variabel-variabel tersebut, nilai rasio data hiding adalah sebagai berikut :

$$RatioDataEmbedding = \frac{161.28}{26525696} = \frac{1}{164469.84} bit$$

Dimana arti dari hasil diatas yaitu dalam 164469.84 bit yang disediakan oleh file audio pembawa pesan rahasia tersebut memiliki kemungkinan untuk disipkan satu bit informasi didalamnya.

# Bab 5

## Penutup

### 5.1 Ringkasan dan Kesimpulan

Dari hasil pengujian sistem yang dilakukan pada bab sebelumnya, maka dapat disimpulkan beberapa hal antara lain:

1. Algoritma dan flowchart yang telah dibuat dalam langkah awal pengerjaan proyek akhir dapat berjalan dengan baik sehingga informasi dapat disisipkan dan diambil kembali isinya dari suatu media audio. Hal ini dibuktikan melalui hasil uji analisa pada tabel 4.1 yang menggambarkan nilai data masukan dan data keluaran yang dihasilkan, dengan tingkat keberhasilan sebesar 100 %.
2. File keluaran yang dihasilkan mengalami noise yang rendah dikarenakan suatu blok hanya diubah maksimal sebanyak 1 bit. Namun noise yang terjadi akan semakin jelas apabila pesan yang disisipkan lebih besar, hal ini dibuktikan melalui penurunan nilai SNR terhadap ukuran file pesan rahasia yang semakin besar.
3. Berdasarkan hasil uji analisa pada tabel 4.1 dapat dinyatakan bahwa file masukan dan file hasil keluaran memiliki jumlah bit yang sama persis, dimana artinya

penyisipan pesan tidak mempengaruhi besaran data masukan maupun keluaran.

4. Kualitas dari file audio sangat ditentukan pada ukuran file pesan rahasia yang dipakai dan ukuran file pembawa. Semakin besar ukuran file pesan rahasia, maka semakin besar pula noise yang ditimbulkan. Hal ini dibuktikan melalui hasil uji analisa pada tabel 4.1 yang menggambarkan bahwa terjadinya penurunan nilai SNR terhadap file keluaran dengan semakin besarnya ukuran file pesan rahasia yang disisipkan.

## **5.2 Rencana Kedepan**

Perangkat lunak ini dibuat untuk melakukan penyisipan data dalam data, dan penulis hanya melakukan uji coba dengan file media audio. Oleh karena itu harapan kami agar aplikasi ini dapat dikembangkan lebih lanjut dengan menggunakan bahasa pemrograman Java untuk memberikan kedinamisan dalam jenis file media yang dipakai maupun pesan rahasia yang dimasukkan dan pengembangan aplikasi sehingga dapat menggunakan berbagai jenis file media lainnya.

# Bibliografi

[Bender, 1998] Bender, W. Gruhl, D. M. N. L. (August 1998). *A.: Techniques for Data Hiding*. PhD thesis.

[Bender, 1996] Bender, D. Gruhl, N. A. L. (Februari 1996). Techniques for data hiding. *IBM System*, 35(3-4).

[Johnson, 2006] Johnson, N. F. (2006). <http://www.jjtc.com/ihws98/jjgmu.html>.

[Simmons., 1983] Simmons., G. (1983). The prisoner's problem and the subliminal channel. In *Crypto'83*:halaman 51–67.

[Waheed, 2000] Waheed, Q. (2000). *Steganography and Steganalysis*. PhD thesis.

[Westfeld, 1999] Westfeld, A. (1999). The steganographic algorithm, f5. <http://wwwrn.inf.tu-dresden.de/ÿwestfeld/f5.html>.

# Indeks

<b>A</b>		Sytactic ..... 12
Algoritma		
Kriptografi.....21		
LSB.....20		
<b>D</b>		
decodeMessage ..... 26		
<b>E</b>		
Embedding ..... 25		
encodeFile.....26		
encodeMessage ..... 26		
<b>F</b>		
Flowchart LSB.....23		
<b>J</b>		
Java J2SDK.....25		
<b>M</b>		
Metode		
Semantic ..... 12		
Spasi Terbuka ..... 10		
Steganografi Gambar ..... 13		
Steganografi Suara..... 17		
	<b>R</b>	
	Retrieving ..... 26	
	<b>S</b>	
	Steganografi ..... 4	
	Kegunaan ..... 7	
	Metode ..... 10	
	Pengertian ..... 4	
	Sejarah ..... 6	
	Tipe Media ..... 8	