AUDIO STEGANOGRAFI DENGAN ALGORITMA LSB UNTUK PENGAMANAN DATA DIGITAL



Disusun Oleh:

Nama : Muhammad Elan Mustakmal

NIM : 12523190

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

AUDIO STEGANOGRAFI DENGAN ALGORITMA LSB UNTUK PENGAMANAN DATA DIGITAL

TUGAS AKHIR

Disusun Oleh:

Nama : Muhammad Elan Mustakmal
NIM : 12523190

Yogyakarta, 20 Mei 2018
Pembimbing,

(Yudi Prayudi, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

AUDIO STEGANOGRAFI DENGAN ALGORITMA LSB UNTUK PENGAMANAN DATA DIGITAL

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 4 Juni 2018

Tim Penguji

Yudi Prayudi, S.Si., M.Kom. _____

Anggota 1

Syarif Hidayat, S.Kom., M.I.T. _____

Anggota 2

Chanifah Indah R., S.Kom., M.Kom. _____

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Hendrik, S.T., M.Eng.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Elan Mustakmal

NIM : 12523190

Tugas akhir dengan judul:

AUDIO STEGANOGRAFI DENGAN ALGORITMA LSB UNTUK PENGAMANAN DATA DIGITAL

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Juni 2018



(Muhammad Elan Mustakmal)

HALAMAN PERSEMBAHAN

Alhamdulillahi Robbil 'Alamin puji syukur atas segala nikmat dan karunia yang Allah SWT berikan kepada saya, keluarga, saudara-saudara, serta sahabat-sahabat.
Sholawat serta salam dipanjatkan kepada Nabi Muhammad SAW sebagai pemberi syafa'at kepada seluruh umat manusia.

Tugas akhir ini saya persembahkan kepada:

Kedua orang tua saya yang tercinta, Bapak Drs. H. Budi Prayitno dan Ibu Hj. Siti Rochmiana. yang senantiasa mendidik, menjaga, mendo'akan, menasehati, memberi dukungan lahir dan batin, serta mengusahakan segala yang terbaik bagi anak-anaknya. Semoga Allah SWT selalu memberikan kesehatan, rahmat, dan karunia-Nya kepada Bapak dan Ibu. Aamiin.

Saudara saudariku, Fatiha Putri Rahmani, Gema Wahyuning Fitri dan Arba'an Syahputra yang selalu menjadi supporter yang baik, memberikan do'a, nasehat dan dukungan semangat. Semoga tali persaudaraan kita selalu terjaga. Aamiin.

Kekasih saya yang selalu percaya penuh terhadap kemampuan saya, yang selalu memberikan dukungan kepada saya, yang selalu minta diajak makan mie ayam, yang membantu saya bangkit dikala jatuh terpuruk Dan menapakkan kaki saya di bumi
dikala sedang melambung.
Semoga Allah SWT merestui hubungan kita. Aamiin.

Teman-teman Elf dan KOSMIK UII #6 yang selalu menorehkan senyum dan tawa di sela alunan nada yang selalu dapat memberikan kenyamanan dan kegembiraan. Semoga persahabatan kita selalu terjalin. Aamiin.

Seluruh teman-teman GRAVITY 2012 yang menjadi teman seperjuangan di bangku perkuliahan.

Dan seluruh teman-teman yang tidak dapat disebutkan satu per satu.

.

HALAMAN MOTO

“Maka sesungguhnya beserta kesulitan itu ada kemudahan. Sesungguhnya beserta kesulitan itu ada kemudahan.”

(QS. Al-Insyirah : 5-6)

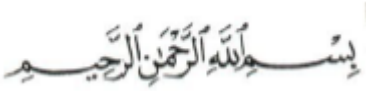
“Lebih baik maju walaupun sedikit, daripada tidak maju sama sekali.”

(Elan mustakmal)

“Orang yang berbahagia adalah mereka yang bangga menjadi dirinya sendiri tanpa mengkhawatirkan apa yang dipikirkan orang lain terhadap dirinya”

(anonim)

KATA PENGANTAR



Assalaamu'alaikum, Wa rahmatullaahi Wa barokaatuh

Alhamdulillaahi Robbil 'Aalamiin, puja dan puji syukur kehadirat Allah SWT yang selalu melimpahkan rahmat dan karunia-Nya kepada kita semua, shalawat dan salam selalu disanjungkan kepada baginda nabi besar – Nabi Muhammad SAW, yang telah membawa kita dari zaman *jahiliyyah* ke zaman yang penuh dengan ilmu ini sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Audio Steganografi Dengan Algoritma LSB Untuk Pengamanan Data Digital" dengan baik.

Laporan ini disusun sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan rangkaian pendidikan pada jenjang Strata 1 Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penyusunan tugas akhir ini, terdapat beberapa kendala dan kesulitan yang dialami oleh penulis, mulai dari fokus yang terpecah oleh urusan lain, sampai kemauan yang pasang surut. Namun pada akhirnya, *Alhamdulillaah* penulis dapat melewati itu semua dengan baik.

Tugas akhir ini dapat terselesaikan dengan baik atas dukungan, bimbingan, dan bantuan yang diberikan oleh berbagai pihak. Maka dari itu penulis mengucapkan ribuan terimakasih kepada :

1. Nandang Sutrisno, SH., LL.M., M.Hum., Ph.D., selaku Rektor Universitas Islam Indonesia.
2. Dr. Imam Djati Widodo, M.Eng, Sc., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Hendrik, S.T, M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Yudi Prayudi S.Si., M.Kom., selaku dosen pembimbing tugas akhir yang selalu membagi ilmu dan dengan sabar memberikan waktunya untuk membimbing penulis dalam menyelesaikan tugas akhir ini.
5. Bapak dan ibu dosen Jurusan Teknik Informatika yang telah membagi ilmunya kepada penulis.

6. Kedua orang tua penulis, Bapak Budi Prayitno dan Ibu Siti Rochmiana yang telah memberikan kasih sayang dan dukungan serta dorongan moril maupun materil dalam menyelesaikan tugas akhis ini.
7. Saudara saudari kandung penulis, Fatiha Putri Rahmani, Gema Wahyuning Fitri, dan Muhammad Arba'an Syahputra, terimakasih atas dukungan, semangat, dan do'anya.
8. Kekasih penulis, terimakasih atas dukungan, doa, dan kepercayaan berbalut cinta yang telah diberikan sampai detik ini.
9. Teman-teman Elf dan KOSMIK UII #6, terimakasih atas canda tawa dan semangatnya.
10. Semua pihak yang tidak mungkin dapat disebutkan satu persatu, terimakasih atas bantuan dan do'anya.

Tugas akhir ini tidak akan lepas dari kekurangan dan kesalahan dikarenakan terbatasnya kemampuan dan pengalaman penulis. Oleh karena itu kritik dan saran yan membangun sangat penulis harapkan agar kedepannya dapat menjadi lebih baik lagi. Semoga laporan ini dapat diterima dan menjadi manfaat bagi pembaca maupun penulisnya. *Aamiin yaa Robbal Aalamiin.*

Wassalaamu'alaikum Wa Rahmatullaahi Wa Barokaatuh.

Yogyakarta, 4 Juni 2018

(Muhammad Elan Mustakmal)

SARI

Kemajuan teknologi di zaman ini menjadikan komunikasi melalui media elektronik bagian besar dari kehidupan setiap orang. karena selain lebih simpel, komunikasi melalui media elektronik juga lebih cepat dan praktis. Namun kemudahan tersebut dapat menjadi sasaran kejahatan pencurian data oleh pihak-pihak yang tidak bertanggung jawab. Tujuan dari penelitian ini adalah untuk menemukan teknik menyembunyikan keberadaan pesan rahasia untuk menghindari pencurian data digital. Steganografi adalah seni berkomunikasi secara rahasia yang dimaksudkan untuk menyembunyikan keberadaan dari sebuah komunikasi yang sedang berlangsung. Audio steganografi adalah teknik menyembunyikan pesan rahasia ke dalam sinyal audio.

Dalam penelitian ini penulis akan membandingkan kualitas audio dengan menghitung dan membandingkan nilai PSNR antara *carrier audio* dengan *stego audio*. *File* yang akan digunakan adalah *file plainteks* berupa pantun berukuran 100 *bytes*. *Carrier audio* yang digunakan merupakan 5 *file* audio berformat WAV. Untuk kemudian dapat dilihat apa saja perubahan yang terjadi terhadap setiap sampel suara setelah dilakukan steganografi.

Kata kunci : Steganografi, Audio steganografi, Audio, LSB

GLOSARIUM

Cover object	file yang akan digunakan sebagai media steganografi
Stego object	file yang mengandung pesan rahasia hasil steganografi
Stego key	kunci untuk membuka pesan rahasia dari stego object
Bit	digit dalam baris biner
Byte	satuan penyimpanan data komputer
Lossless	tidak berkurang
Lossy	ada yang berkurang
Flowchart	diagram alir
Embedding	menyembunyikan
Extracting	mengekstrak
Secret message	pesan rahasia
Hidden message	pesan tersembunyi
Options	pilihan-pilihan
Pushbutton	tombol tekan
Laptop	komputer jinjing
Mouse	piranti masukan untuk menggerakan kursor pada komputer
Keyboard	piranti masukan untuk mengetik
Pop up	muncul tiba-tiba
Source code	sumber kode
File	berkas

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING **Error! Bookmark not defined.**
HALAMAN PENGESAHAN DOSEN PENGUJI ii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 3
1.6 Metodologi Penelitian 3
1.7 Sistematika Penelitian 4
BAB II LANDASAN TEORI 5
2.1 Penelitian Terdahulu 5
2.2 Audio 6
2.2.1 WAV (WAVeform audio format) 7
2.3 Steganografi 8
2.3.1 Sejarah Steganografi 9
2.3.2 Skema Steganografi 9
2.3.3 Metode Steganografi 10
2.4 Audio Steganografi 11
2.5 Algoritma LSB (*Least Significant Bit*) 13
2.6 Data 14
2.6.1 Jenis-jenis Data 14

2.7 Data Digital 15
2.8 Matlab 16
2.9 *Peak Signal to Noise Ratio* (PSNR) 17
BAB III METODOLOGI DAN PERANCANGAN 18
3.1 Metode Penelitian 18
3.2 Analisis Kebutuhan 19
3.2.1 Analisis Kebutuhan Masukan 19
3.2.2 Analisis Kebutuhan Keluaran 20
3.3 Diagram Alir 20
3.3.1 *Flowchart* 21
3.3.2 *Data Flow Diagram* (DFD) 23
3.4 Perancangan Desain Antarmuka 25
3.4.1 Desain antarmuka proses *embedding* 26
3.4.2 Desain antarmuka proses *extracting* 27
3.4.3 Desain antarmuka proses hitung PSNR 28
3.5 Rancangan Pengujian 29
3.5.1 Rancangan pengujian proses *embedding* 29
3.5.2 Rancangan pengujian proses *extracting* 29
3.5.3 Rancangan pengujian proses hitung PSNR 30
BAB IV IMPLEMENTASI HASIL DAN ANALISIS 31
4.1 Implementasi 31
4.1.1 Lingkungan implementasi 31
4.1.2 Implementasi antarmuka 32
4.1.3 *Source Code* untuk Membangun Implementasi 42
4.2 Uji Kinerja Aplikasi 48
4.2.1 Uji proses *embedding* dan *extracting* 48
4.2.2 Uji perbandingan *carrier audio* dengan *stego audio* 49
4.2.3 Uji PSNR 50
4.3 Analisis Hasil Pengujian 50
BAB V KESIMPULAN DAN SARAN 52
5.1 Kesimpulan 52
5.2 Saran 52
DAFTAR PUSTAKA 53

DAFTAR TABEL

Tabel 3.1 Daftar *file* audio yang akan dijadikan *carrier audio*.....29
Tabel 4.1 Hasil uji proses *embedding*.....48
Tabel 4.2 Hasil uji proses *extracting*.....48
Tabel 4.3 Perbandingan *carrier audio* dengan *stego audio*.....49
Tabel 4.4 Hasil uji PSNR.....50

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi rambat gelombang bunyi 7
Gambar 2.2 Struktur file WAV (Craig, 2008) 8
Gambar 2.3 Skenario aplikasi steganografi 10
Gambar 2.4 Grafik Spread spectrum 11
Gambar 2.5 Skema algoritma LSB 13
Gambar 3.1 *Flowchart embedding* 21
Gambar 3.2 *Flowchart extracting* 22
Gambar 3.3 *Flowchart* hitung PSNR 23
Gambar 3.4 DFD Level 0 24
Gambar 3.5 DFD Level 1 25
Gambar 3.6 Rancangan antarmuka proses *embedding* 26
Gambar 3.7 Rancangan antarmuka proses *extracting* 27
Gambar 3.8 Rancangan antarmuka proses hitung PSNR 28
Gambar 4.1 Tampilan awal aplikasi 32
Gambar 4.2 Tampilan awal proses *embedding* 33
Gambar 4.3 Tampilan proses memilih file *carrier audio* 34
Gambar 4.4 Tampilan periodogram pada proses *embedding* 34
Gambar 4.5 Tampilan *pop up message* 35
Gambar 4.6 Tampilan proses *embedding* 35
Gambar 4.7 Tampilan proses pemberian nama untuk hasil *embedding* 36
Gambar 4.8 Tampilan *pop up message* setelah proses *embedding* berhasil dilakukan 36
Gambar 4.9 Tampilan awal prose *extracting* 37
Gambar 4.10 Tampilan proses pemilihan *stego audio* 38
Gambar 4.11 Tampilan periodogram pada proses *extracting* 39
Gambar 4.12 Tampilan proses *extracting* 39
Gambar 4.13 Tampilan awal proses hitung PSNR 40
Gambar 4.14 Tampilan proses pemilihan *carrier audio* 41
Gambar 4.15 Tampilan proses pemilihan *stego audio* 41
Gambar 4.16 Tampilan proses hitung PSNR 42
Gambar 4.17 *Source code* tombol *select audio* 42
Gambar 4.18 *Source code* tombol *execute* 43
Gambar 4.19 *Source code* tombol *execute* 44

Gambar 4.20 *Source code* tombol *execute*.....45
Gambar 4.21 *Source code* tombol *select audio* 45
Gambar 4.22 *Source code* tombol *execute*.....46
Gambar 4.23 *Source code* tombol *execute*.....46
Gambar 4.24 *Source code* tombol *select audio* 47

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komunikasi yang begitu pesat berbanding lurus dengan tindak kejahatan siber yang juga turut berkembang. Sebagaimana laporan yang dirilis *Javelin Strategy & Research* mengungkap, pada 2013 ada 13,1 juta korban yang datanya dicuri secara digital. Angka itu berarti terjadi kenaikan hingga 28% dibanding periode sebelumnya. Sementara yang jadi korban, jumlahnya naik lebih dari 500 ribu akun. Angka tersebut juga menunjukkan adanya kenaikan kasus yang luarbiasa dibanding angka tahun 2010. Saat itu, 'hanya' 10,2 juta saja, akun yang dibobol.

Maraknya kasus pencurian data digital ini mulai menyedot perhatian publik termasuk para pegiat ilmu forensika digital yang tergerak untuk menemukan cara untuk mengamankan data digital dengan efisien dan cepat dari pihak-pihak yang tidak bertanggung jawab, salah satunya dengan menggunakan teknik steganografi.

Steganografi merupakan seni dan ilmu untuk menyembunyikan komunikasi rahasia yang sedang terjadi. Dengan steganografi kita dapat menyusupkan data rahasia ke dalam suatu objek yang tidak mencurigakan dan mengirimnya tanpa ada pihak lain yang mengetahui keberadaan pesan rahasia tersebut. Kata steganografi berasal dari bahasa yunani; *stegos* yang berarti tersembunyi, dan *graphia* yang berarti tulisan sehingga artinya secara harfiah kurang lebih adalah menulis tulisan yang tersembunyi.

Dalam skenario steganografi, data rahasia disembunyikan di dalam objek lain yang disebut sebagai *cover object* untuk membentuk *stego object* yang kemudian *stego object* ini dapat disimpan atau dikirim dengan aman ke penerima yang dikehendaki dan kemudian penerima tersebut dapat membuka pesan rahasia di dalam *stego object* dengan *stego key* yang dibuat oleh pengirim. Dengan menggunakan teknik yang berbeda, kita dapat mengirim data rahasia dalam bentuk gambar, suara, bahkan video dengan menyusupkannya ke dalam sebuah objek lainnya dan membentuk *stego object.*

Audio steganografi adalah salah satu skema steganografi yang menggunakan *file* audio sebagai *cover object* yang disebut sebagai *carrier audio* untuk membawa data rahasia yang dikirimkan dengan memodifikasi sinyal audio sedemikian rupa. Audio steganografi dilakukan untuk memanfaatkan kelemahan pada sistem pendengaran mausia. Banyak metode yang

ditemukan untuk melakukan audio steganografi, salah satunya dengan mengunakan algoritma LSB (*Least Significant Bit)*.

Algoritma LSB adalah metode yang paling populer dalam audio steganografi. Algoritma ini bekerja dengan cara memodifikasi *bit-bit* terakhir dalam beberapa *byte file* audio untuk menyembunyikan urutan *byte* yang mengandung data rahasia. LSB merupakan teknik yang paling simpel dan efektif jika dibandingkan dengan metode lainnya dan tidak menyebabkan perubahan kualitas yang signifikan terhadap audio *file* yang menjadi *cover object* tersebut.

Berdasarkan hal tersebut maka penulis mencoba melakukan suatu penelitian untuk menguji seberapa efektif audio steganografi dengan menggunakan algoritma LSB untuk meminimalisir terjadinya pencurian informasi digital oleh pihak yang tidak bertanggung jawab.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat disimpulkan bahwa terdapat masalah berupa butuhnya pemanfaatan *file* audio sebagai media untuk pengamanan data digital dengan waktu eksekusi yang singkat. Maka rumusan masalah yang dapat dibuat adalah bagaimana mengimplementasikan algoritma LSB ke dalam audio steganografi untuk mengamankan data digital, dan apa saja perubahan yang terjadi terhadap *file* audio setelah dilakukan proses steganografi.

1.3 Batasan Masalah

Adapun beberapa batasan masalah di dalam pembuatan aplikasi audio steganografi adalah sebagai berikut :

a. Algoritma steganografi yang digunakan dalam penelitian ini adalah LSB.
b. Media steganografi adalah 5 *file* audio berformat wav berukuran 1,24 MB. Dengan pesan rahasia berupa pantun berukuran 100 *bytes*.
c. Penelitian berfokus pada proses penymbunyian pesan rahasia, ekstraksi pesan rahasia, dan perhitungan PSNR.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui apa itu audio steganografi, mengimplementasikan penggunaan algoritma LSB untuk melakukan audio steganografi dan melihat perbedaan antara *carrier audio* dengan *stego audio*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi penulis dan pembaca. Antara lain:

a. Memperkaya literatur mengenai penerapan ilmu pengetahuan yang didapatkan penulis selama menempuh pendidikan di bangku kuliah.
b. Menambah wawasan ilmu dan pemahaman bagi penulis dan pembaca tentang pengimplementasian algoritma LSB dalam audio steganografi.
c. Dapat meminimalisir kasus bocornya data dan informasi yang bersifat rahasia.
d. Menjadi acuan untuk penelitian-penelitian selanjutnya tentang audio steganografi dengan algoritma LSB.

1.6 Metodologi Penelitian

Untuk melakukan penelitian ini, ada beberapa tahapan yang dilakukan agar dapat memperoleh hasil akhir yang diharapkan. Tahapan-tahapan penelitian tersebut di antaranya :

Studi literatur

Tahap awal untuk mempelajari teori dan metode yang sesuai dan berkaitan dengan penelitian ini.

Analisis kebutuhan

Tahap awal untuk mencari dan mengumpulkan bahan-bahan yang sekiranya diperlukan untuk penelitian ini.

Perancangan perangkat lunak

Tahap untuk merancang aplikasi yang akan dibuat mulai dari desain antarmuka

Implementasi

Implementasi dilakukan dengan membuat aplikasi dari rancangan yang telah dibuat

Pengujian Sistem

Pengujian sistem dilakukan setelah aplikasi telah selesai dibangun. Pengujian dilakukan dengan cara memasukkan data yang sudah ditentukan kemudian mengevaluasi performa dan kinerja aplikasi tersebut.

1.7 Sistematika Penelitian

Laporan penelitian ini dibuat dengan urutan sebagai berikut:

BAB I PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penelitian yang berhubungan dengan penelitian ini.

BAB II LANDASAN TEORI

Bab ini akan memaparkan beberapa tinjauan pustaka yang dikutip berdasarkan pada penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang metodologi yang digunakan untuk menyelesaikan penelitian ini. Selain itu pada bab ini berisikan analisis terhadap kebutuhan perangkat lunak, pembuatan dan perancangan aplikasi audio steganografi.

BAB IV IMPLEMENTASI HASIL DAN PEMBAHASAN

Merupakan bagian yang berisikan hasil implementasi dan proses pengujian dari perancangan dan pembuatan aplikasi audio steeganografi.

BAB V KESIMPULAN DAN SARAN

Bab terakhir yang memuat kesimpulan-kesimpulan yang didapatkan dari seluruh tahapan yang dilakukan dari pengujian sistem yang disesuaikan dengan konsep awal penelitian. Berisikan saran yang perlu diperhatikan berdasarkan pada hasil pengujian yang telah dilakukan pada penelitian ini terkait dengan kekurangan-kekurangan sistem agar dapat dikembangkan lebih lanjut

BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian mengenai audio steganografi dengan algoritma LSB telah banyak dibahas dan dikaji dalam penelitian-penelitian terdahulu dengan tujuan untuk mengamankan data dari tindak kejahatan siber. Salah satu penelitian yang membahas audio steganografi dengan algoritma LSB adalah jurnal yang ditulis oleh Vipka Firdaus dengan judul "Studi Dan Implementasi Steganografi Pada File Audio dengan Teknik *Spread Spectrum*" (Firdaus, 2009). Penelitian tersebut membahas tentang penyisipan pesan rahasia ke dalam file WAV dengan menggunakan algoritma *Spread Spectrum* yang diimplementasikan dengan bahasa Microsoft Visual Studio C#.NET 2012. Dalam penelitian tersebut dapat diketahui bahwa transmisi sinyal audio dilakukan dengan metode *spread spectrum* yang berkerja dengan cara menyebar sinyal audio melalui *bandwidth* yang lebih besar daripada *bandwidth* sinyal audio yang menampung informasi dengan memanfaatkan modulasi dari kode *pseudo noise.* Cara tersebut dimaksudkan untuk mencegah terjadinya pencegatan informasi ataupun gangguan lainnya. Pada penelitian tersebut, aplikasi yang dibangun berhasil melakukan proses steganografi terhadap 4 sampel audio wav dengan nilai PSNR rata-rata sebesar 40 dB. Kelemahan yang terlihat pada penelitian ini adalah ukuran stego file yang masih besar sehingga membutuhkan waktu pemrosesan yang lumayan lama dan kualitas audio yang tidak begitu bagus dengan nilai PSNR rata-rata sebesar 40 dB mengingat ukuran file yang besar dan pesan rahasia yang kecil.

Penelitian serupa juga telah dilakukan oleh Maryam dengan judul jurnal "Audio Steganografi dengan Metode *Least Significant Bit* Terhadap Pesan Terenkripsi dengan Algoritma *Twofish"*(MARYAM, 2013). Penelitian tersebut membahas tentang proses pembuatan aplikasi untuk menyisipkan file teks yang terenkripsi ke dalam file audio mp3. Proses enkripsi dilakukan terhadap pesan rahasia dengan menggunakan algoritma twofish yang bekerja dengan kunci simetris dan memiliki blok chiper sepanjang 128 bit dan key sepanjang 256 bit. Ukuran kunci yang besar dimaksudkan untuk meminimalisir adanya *weak-key*. Setelah proses enkripsi selesai dilakukan, proses dilanjutkan dengan menyisipkan pesan rahasisa terenkripsi ke dalam sinyal audio dengan menggunakan metode LSB. Dalam penelitian tersebut dapat dilihat bahwa proses enkripsi dan steganografi berhasil dilakukan terhadap seluruh sampel audio dan ukuran pesan rahasia yang dapat disisipkan bergantung pada

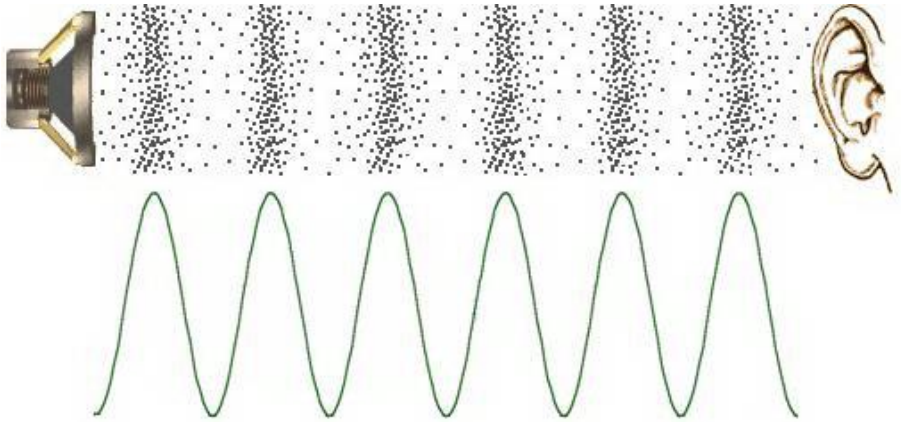
banyaknya *byte* homogen dalam suatu file audio yang berarti semakin besar ukuran file audio maka kapasitas pesan rahasia semakin banyak. Kelemahan yang terdapat pada penelitian tersebut adalah waktu eksekusi yang relatif lama dan kualitas audio yang kurang baik dengan nilai PSNR rata-rata sebesar 40 dB.

Jika dilihat dari rangkuman kedua penelitian di atas, maka dapat disimpulkan bahwa penelitian ini memiliki beberapa keunikan tersendiri. Aplikasi steganografi yang dibangun oleh Vipka Firdaus menggunakan metode *spread spectrum* yang berbeda dengan penelitian ini yang dibangun dengan metode LSB. Hal tersebut dikarenakan metode *spread spectrum* membutuhkan pembuatan *pseudo code* sehingga memperlambat proses eksekusi, berbeda dengan LSB yang langsung menyisipkan pesan rahasia sehingga memiliki waktu eksekusi yang relatif lebih cepat. Sementara aplikasi audio steganografi yang dibangun oleh Maryam menggunakan pesan rahasia terenkripsi untuk kemudian dilakukan proses steganografi dengan algoritma LSB yang berbeda dengan penelitian ini yang tidak menerapkan proses enkripsi terhadap pesan rahasia. Hal tersebut dikarenakan proses enkripsi dan dekripsi membutuhkan proses dan barisan kode yang relatif lebih panjang sehingga akan memakan waktu eksekusi yang lebih lama jika dibandingkan dengan penelitian ini yang tidak menggunakan proses enkripsi dan dekripsi sehingga memiliki baris kode yang relatif lebih pendek sehingga mempercepat proses eksekusi.

2.2 Audio

Menurut seorang pakar multimedia bernama Lu pada tahun 1999, audio adalah salah satu elemen dari multimedia yang tercipta karena adanya perubahan tekanan udara yang menjangkau gendang telinga manusia. Sedangkan menurut Andleigh, jika frekuensi tekanan udara berada pada 20 sampai 20.000 Hz, maka telinga manusia akan mengidentifikasinya sebagai suara. Frekuensi adalah banyaknya gelombang suara yang dihasilkan dalam satuan waktu, suatu gelombang dikatakan sebagai periode apabila gelombang tersebut memiliki pola yang sama dan berulang pada interval tertentu. Menurut (E-jurnal, 2016), suara dibedakan menjadi 2 :

a. Suara Periodik : Adalah suara yang terjadi secara terus-menerus/kontinyu dan getaran yang dihasilkan sama. Contoh : instrumen musik dan nyanyian burung.
b. Suara Non-periodik : Adalah suara yang tidak terjadi secara berlanjut dan memiliki bentuk gelombang kotak-kotak. Contoh : suara batuk dan percikan ombak.



Gambar 2.1 Ilustrasi rambat gelombang bunyi

Sumber: E-jurnal (2016)

Data audio digital pada sistem komputer disimpan dalam suatu format *file* audio. Data tersebut dapat disimpan tanpa dikompresi maupun dikompresi untuk mengurangi ukuran *file* tersebut. Ini dapat menjadi sebuah *raw bitstream*, tetapi biasanya merupakan sebuah *container format* atau format data audio dengan lapisan penyimpanan yang telah ditentukan (MARYAM, 2013). Menurut (Haris, 2010), ada tiga kelompok utama format *file* audio :

a. Format audio yang tidak terkompresi, seperti WAV, AIFF, AU atau raw header-less PCM.
b. Format audio dengan kompresi *lossless*, seperti FLAC, MP3, TTA, Monkey's Audio (berekstensi APE), MPEG -4 DST, dan lainnya.
c. Format audio dengan kompresi *lossy*, seperti MP#, Vorbis, Musepack, AAC, dan ATRAC.

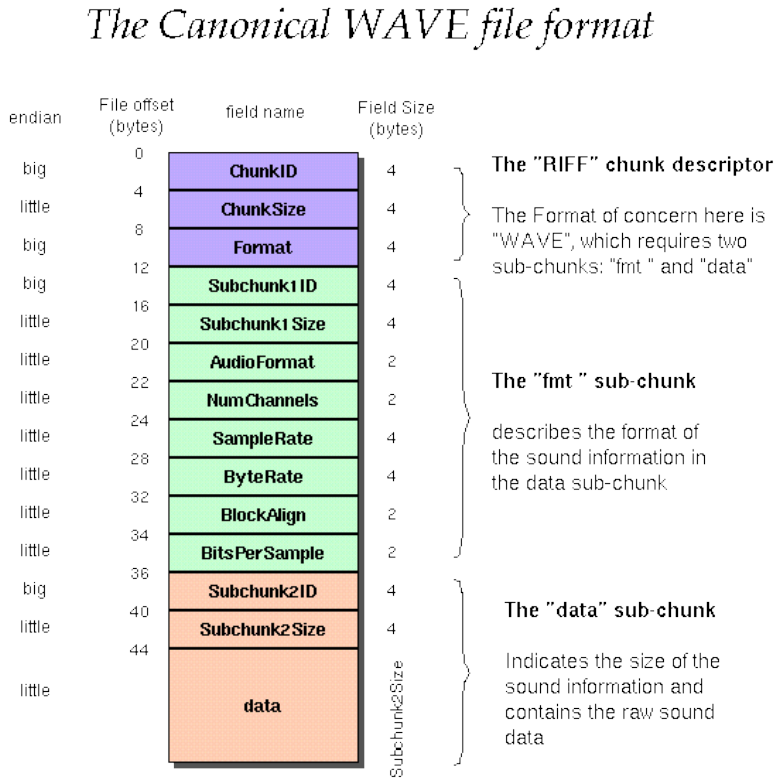
2.2.1 WAV (WAVeform audio format)

Adalah format *file* audio yang dibuat oleh Microsoft dan pertama kali dirilis pada tahun 1991. WAV merupakan format audio standar untuk PC mulai dari suara yang terdapat pada sistem dan permainan, sampai kualitas standar audio CD. *File* WAV biasanya tidak tekompresi, namun dapat juga menyimpan *file* yang sudah terkompresi.

File WAV menggunakan *sample rate* 44,1KHz yang berarti audio dengan format WAV memainkan 44.100 *sample* setiap detiknya dan setiap *sample* berukuran 2 *byte* untuk *mono*, dan 4 *byte* untuk *stereo* Sehingga perhitungan untuk memainkan sebuah lagu WAV *stereo* berdurasi 4 menit adalah 2 x 2 byte x 44.100 kali/detik x 4 x 60 detik = 42.336.000 byte (42MB). Sebagai perbandingan, perhitungan untuk memainkan sebuah *file* audio dengan

format mp3 dengan *bit rate* 128 kbps berdurasi 4 menit adalah 128 kbps x 4 x 60 detik = 30720 kilobit (4,25MB) (Salikin, 2008).

Dari kedua perhitungan di atas, dapat disimpulkan bahwa untuk memainkan *file* audio berdurasi 4 menit dengan format WAV membutuhkan kapasitas penyimpanan yang jauh lebih besar dibandingkan *file* audio berdurasi 4 menit dengan format mp3. Hal tersebut disebabkan oleh kerapatan *sample rate* format WAV yang jauh lebih rapat dibandingkan format mp3. Itulah mengapa file WAV biasanya memiliki ukuran yang relatif lebih besar dan lebih mumpuni untuk dijadikan media steganografi jika dibandingkan dengan format audio .mp3.



Gambar 2.2 Struktur file WAV

Susmber: Craig (2008)

2.3 Steganografi

Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia (Wikipedia, 2018). Steganografi merupakan seni penyembunian pesan ke dalam pesan lainnya sedemikian rupa sehingga orang lain tidak dapat menyadari ada sesuatu di dalam pesan tersebut (Purba,

Situmorang, & Arisandi, 2012). Kata steganografi berasal dari bahasa yunani; *stegos* yang berarti tersembunyi, dan *graphia* yang berarti tulisan sehingga artinya secara harfiah kurang lebih adalah menulis tulisan yang terselubung.

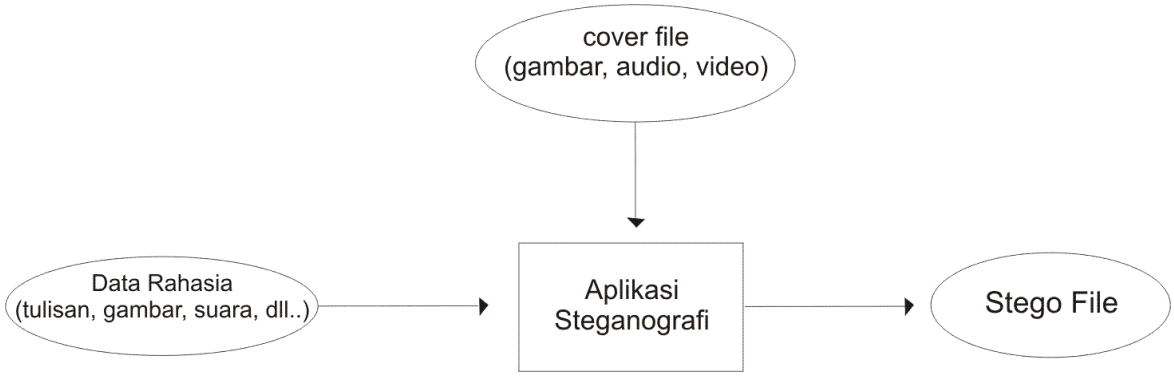
Tidak seperti kriptografi yang merubah susunan bentuk suatu pesan setelah dilakukan operasi enkripsi, steganografi hampir tidak merubah bentuk suatu pesan setelah disisipkan pesan rahasia. Konsep *Security Through Obscurity* ini membuat pihak ketiga yang tidak berhak atas informasi rahasia tersebut tidak menyadari adanya pesan rahasia di balik pesan tersebut.

2.3.1 Sejarah Steganografi

Sejarah mencatat bahwa seni steganografi sudah digunakan berabad-abad lalu oleh yunani kuno. Steganografi pertama kali dipraktekkan oleh seorang raja kejam bernama Histaeus yang dipenjara karena kekejamannya oleh raja Darius pada abad ke-5 sebelum masehi. Kala itu Histaeus yang berada di dalam penjara ingin mengirim pesan kepada anaknya, Aristogras, yang berada di Militus. Untuk mengelabui penjaga dan menjaga kerahasiaan pesan tersebut, Histaeus pun mentato pesan di kulit kepala seorang budak. Lalu ketika rambut sudah mulai tumbuh, Histaeus mengutus budak tersebut ke Militus untuk menyampaikan pesan rahasia kepada Aristogras.

2.3.2 Skema Steganografi

Dalam skenario steganografi, data rahasia disisipkan ke dalam objek lain yang disebut sebagai *cover object* menggunakan aplikasi steganografi untuk membentuk *stego object* yang kemudian *stego object* ini dapat disimpan atau dikirim dengan aman ke penerima yang dikehendaki. Dengan menggunakan teknik yang berbeda, kita dapat mengirim data rahasia dalam bentuk gambar, suara, bahkan video dengan menyusupkannya ke dalam sebuah objek lainnya dan membentuk *stego object* sebagaimana yang digambarkan pada Gambar 2.3



Gambar 2.3 Skenario aplikasi steganografi

Steganografi dapat dilakukan dengan menggunakan berbagai jenis media sebagai cover object. Dalam steganografi digital sendiri, format media yang biasa digunakan antara lain adalah *file* gambar, audio, dan video.

2.3.3 Metode Steganografi

Dalam skema steganografi, proses penyembunyian pesan rahasia dilakukan dengan cara mengidentifikasi *cover object* dan memanfaatkan bit redundant yang dapat dimodifikasi dengan metode yang tepat untuk meminimalisir resiko kerusakan pada *cover object* tersebut. Menurut (Srikandi, 2015), berikut adalah beberapa metode yang paling umum digunakan dalam steganografi :

a. Algoritma LSB

   LSB (*Least Significant Bit*) adalah metode steganografi yang paling populer karena dianggap paling mudah dan efektif untuk digunakan. Cara kerja LSB adalah dengan mengganti *bit* paling tidak signifikan dalam beberapa *byte* dari *cover object* untuk menyembunyikan urutan *byte* yang mengandung informasi rahasia.

b. *Algorithms Compression*

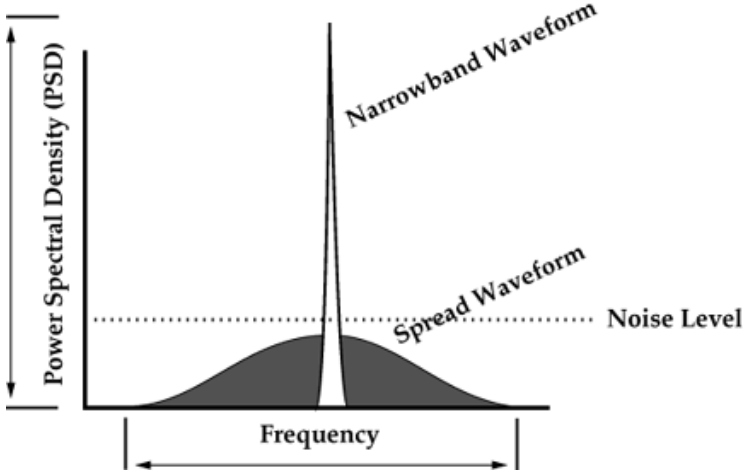
   Algoritma *compression* adalah metode steganografi dengan menyembunyikan data dalam fungsi matematika. Dua fungsi tersebut adalah *Discrete Cosine Transformation* (DCT) dan *Wavelet Transformation*. Fungsi DCT dan *Wavelet* adalah mentransformasikan data dari satu domain ke domain yang lain. Fungsi DCT yaitu mentransformasi data dari *spatial domain* ke *frequency domain.*

c. *Redundant Pattern Encoding* (RPE)

RPE adalah menggambar pesan kecil pada kebanyakan gambar. Keuntungan dari metode ini adalah dapat bertahan dari kegagalan. Kerugiannya adalah tidak dapat menggambar pesan yang lebih besar.

d. *Spread Spectrum*

Spread Spectrum bekerja dengan cara menyebar pesan rahasia di sepanjang frekuensi spektrum dari *cover object*. Tidak seperti LSB, *Spread Spectrum* menyebar pesan rahasianya secara acak. Namun metode *Spread Spectrum* memiliki kelemahan yaitu memiliki waktu eksekusi yang relatif lama dan cenderung mengakibatkan *noise* yang besar terhadap *cover object*.



Gambar 2.4 Grafik Spread spectrum

Sumber: Firdaus (2009)

2.4 Audio Steganografi

Audio steganografi merupakan salah satu skema steganografi yang menggunakan audio file sebagai *cover object* yang disebut dengan *carrier audio* untuk membawa pesan rahasia yang dikirimkan dengan memodifikasi sinyal audio sedemikian rupa. Audio steganografi merupakan teknik yang memanfaatkan keterbatasan yang terdapat pada indra pendengaran manusia (*human auditory system*) bahwa indra pendengaran manusia tidak dapat membedakan suara pelan dan keras yang mana suara keras dapan menenggelamkan suara yang lebih pelan. Dalam steganografi, format *file* dalam bentuk audio memiliki kelebihan dibandingkan fomat citra maupun video. Berkas audio biasanya berukuran relatif lebih besar dibandingkan dengan format citra, sehingga dapat menampung pesan rahasia dalam jumlah yang lebih besar pula. Adapun format video memang berukuran relatif sangat besar, tetapi ukurannya yang sangat

besar tersebut mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

Menurut penelitian yang dilakukan oleh (Saragih, 2006), terdapat beberapa performa yang harus diperhatikan dalam merancang suatu skema audio steganografi, yaitu:

a. Kualitas Audio

Hal terpenting dalam audio steganografi adalah proses penyisipan data tidak boleh sampai mengubah kualitas suara dari sinyal audio yang dijadikan sebagai *carrier audio* sehingga pendengar tidak dapat mendeteksi data rahasia yang disisipkan ke dalam sinyal audio tersebut. Dalam menentukan kualitas audio yang telah disisipi pesan rahasia adalah SNR (*signal to noise ratio*).

b. Bit Rate

Bit rate berguna untuk menghitung banyaknya pesan rahasia yang dapat disisipkan ke dalam sinyal *carrier audio* per satuan waktu (*second*). Penyisipan *serial number* dan pengidentifikasian hak cipta membutuhkan sedikit data yang disisipkan pada sinyal *carrier audio*.

c. Keamanan

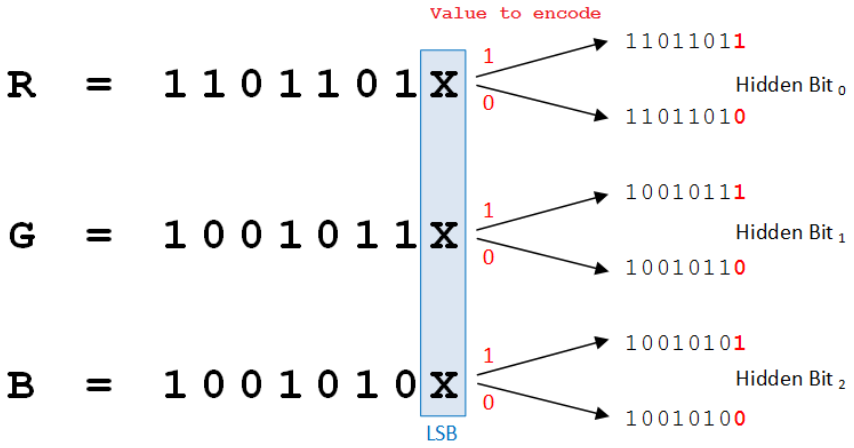
Keamanan dalam proses penyisipan data pada setiap aplikasi adalah hal mutlak dan bertujuan untuk mencegah pendeteksian ataupun penghapusan data oleh pihak lain terhadap data rahasia yang telah disisipkan. Setiap aplikasi membutuhkan keamanan yang berbeda tergantung pada seberapa penting data yang perlu diamankan dan seberapa sistem yang harus diamankan. Skema penyisipan data akan aman apabila diketahui algoritma yang akan digunakan. Pihak lain selain pengirim dan penerima tidak akan dapat mengambil data walaupun pihak tersebut mengetahui bahwa terdapat pesan rahasia di dalam sinyal audio tersebut.

d. Kesulitan Perhitungan

Kesulitan perhitungan yang dilihat berdasarkan pada proses yang dibutuhkan untuk menyisipkan data ke dalam sinyal audio dan pengambilan data tersebut dari sinyal audio. Hal terebut merupakan hal yang paling esensial dan kritis apabila aplikasi yang digunakan beroperasi secara online baik dalam penyisipan dan pengambilan data. Kesulitan algoritma juga merupakan hal yang penting dan mempengaruhi pemilihan struktur implementasi atau arsitektur pemrosesan sinyal digital.

2.5 Algoritma LSB (*Least Significant Bit*)

Tujuan utama dalam steganografi adalah untuk mengirim pesan rahasia dengan bantuan *cover object*. Untuk mencapai tujuan tesebut, pesan rahasia harus disembunyikan ke dalam *cover object* untuk membentuk sebuah *stego object* yang kemudian dapat dikirim ke tujuan yang diinginkan. Algoritma LSB (*Least Significant Bit)* adalah metode yang paling populer dan paling banyak digunakan dalam melakukan audio steganografi. Menurut (MARYAM, 2013) Algoritma LSB bekerja dengan cara memodifikasi nilai yang paling kurang signifikan dari jumlah *bit* dalam 1 *byte file carrier*. *Bit* yang memiliki signifikansi paling tinggi adalah numerik yang memiliki nilai tertinggi (contoh, $2^7 = 128$), artinya bila terjadi perubahan pada *bit* ini maka akan menghasilkan perubahan yang sangat signifikan. *Bit* yang memiliki signifikansi paling rendah adalah numerik yang memiliki nilai paling rendah (contoh, $2^0 = 1$), artinya bila terjadi perubahan pada bit ini maka akan menghasilkan perubahan yang tidak terlalu signifikan.



Gambar 2.5 Skema algoritma LSB

Sumber: Julianto (2016)

Sebagai contoh, akan dilakukan proses penyembunyian karakter ‘H’ (ASCII 72) pada *file carrier* yang berukuran 8 *byte*. LSB dari *file carrier* ditandai dengan garis bawah. Berkas *carrier* berukuran 8 *byte* dalam biner :

‘10010101 00001101 11001001 10010110
00001111 11001011 10011111 00010000’

Karakter ‘H’ dalam biner dengan ukuran 1 *byte* :

‘01001000’

Kedelapan *bit* ini kemudian dimasukkan ke dalam LSB dari tiap-tiap *byte* pada *file carrier* menjadi seperti berikut :

‘1001010**0** 0000110**1** 1100100**0** 1001011**0**
00001111 1100101**0** 1001111**0** 0001000**0**’

Pada contoh di atas, perubahan hanya terjadi pada *bit-bit* paling terakhir dari *file carrier* (ditandai dengan karakter berwarna merah). Berdasarkan contoh di atas, diperoleh teori bahwa kemungkinan terjadinya perubahan *bit* adalah sekitar 50%, karena peluang perubahannya adalah antara 1 atau 0 dan dengan mengubah LSB maka ukuran *file* pembawa tidak akan berubah sehingga akan sulit untuk terdeteksi (MARYAM, 2013).

2.6 Data

Data adalah catatan atas kumpulan fakta yang merupakan bentuk jamak dari *datum*. *Datum* berasal dari bahasa latin yang berarti “sesuatu yang diberikan”. Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima apa adanya (Wikipedia, 2008). Sedangkan menurut (Imam, 2015), Data adalah fakta mentah atau rincian peristiwa yang belum diolah, yang terkadang tidak dapat diterima oleh akal pikiran dari si penerima data tersebut, maka dari itu harus diolah terlebih dahulu menjadi informasi agar dapat diterima dengan baik oleh penerima. Data dapat ditemukan dalam bentuk angka, karakter, simbol, gambar, suara, dan sebagainya. Informasi adalah sekumpulan data yang sudah diolah sehingga dapat diterima oleh akal pikiran penerima yang kemudian dapat digunakan sebagai pertimbangan dalam pengambilan keputusan.

Dalam dunia keilmuan, data merupakan sekumpulan fakta. Data-data kemudian diolah sehingga menjadi suatu informasi yang dapat dimengerti oleh orang lain yang tidak mengalaminya sendiri secara langsung, hal tersebut dinamakan deskripsi. Pemilahan banyak data berdasarkan persamaan dan perbedaan yang terdapat pada data tersebut dinamakan klasifikasi.

2.6.1 Jenis-jenis Data

Menurut (Sayogo, 2015), data dapat dikasifikasikan berdasarkan sifat, sumber, cara memperoleh, waktu pengumpulan, dan metode transmisinya.

a. Jenis data menurut sifatnya :

1. Data Kualitatif : merupakan data yang tidak berbentuk angka, contoh: kuesioner pertanyaan mengenai sistem yang sudah ada, pelayanan suatu jasa, gaya kepemimpinan, dan lainnya.
2. Data Kuantitatif : merupakan data yang berbentuk angka, contoh: harga bahan pokok, jumlah pendapatan, dan sebagainya.

b. Jenis data berdasarkan sumbernya :
   1. Data Internal : merupakan data yang menggambarkan keadaan suatu organisasi dan berasal dari dalam organisasi tersebut. Contoh: jumlah perusahaan dalam suatu perusahaan, gaji karyawan, dan sebagainya.
   2. Data Eksternal : merupakan data yang mungkin mempengaruhi kinerja suatu organisasi dan berasal dari luar organisasi tersebut.

c. Jenis data berdasarkan cara memperolehnya :
   1. Data Primer : merupakan data yang dikumpulkan oleh seseorang atau suatu organisasi terhadap objek penelitian langsung tanpa perantara.
   2. Data Sekunder : merupakan data yang dikumpulkan oleh seseorang atau suatu organisasi berdasarkan penelitian-penelitian terdahulu terhadap objek yang diteliti (tidak langsung).

d. Jenis data berdasarkan waktu pengumpulannya :
   1. Data *Cross Section* : merupakan data yang dikumpulkan dalam satu waktu tertentu untuk meneliti aktifitas pada waktu tersebut.
   2. Data Berkala : merupakan data yang dikumpulkan dalam jangka waktu tertentu secara berkala untuk melihat perkembangan aktifitas pada jangka waktu tersebut.

e. Jenis data berdasarkan metode transmisinya :
   1. Data Analog : merupakan gelombang elektronik yang bervariasi dan secara terus menerus ditransmisikan melalui beragam media bergantung pada frekuensinya.
   2. Data Digital : merupakan data yang ditransmisikan dalam bentuk bit-bit biner sebagai pengganti gelombang.

2.7 Data Digital

Menurut (Dendy, 2014), Data digital yang biasa disebut sebagai *baseband*, merupakan data yang ditransmisikan dalam bentuk *bit-bit* biner sebagai pengganti gelombang. Sistem biner merupakan sistem dengan dua kondisi (on-off, ya-tidak, 1-0), jadi bila terdapat tegangan maka akan dikondisikan sebagai 1. dan sebaliknya, jika tidak terdapat tegangan maka

dikondisikan sebagai 0. Sedangkan menurut (Tanty, 2010) menyatakan bahwa data digital adalah data dalam bentuk pulsa yang dapat mengalami perubahan yang tiba-tiba dan mempunyai besaran 1 dan 0. Sinyal data digital yang mempunyai dua keadaan disebut sebagai *bit*. Sebuah *bit* dapat berupa nol (0) atau satu (1). Jika probabilitas nilai untuk sebuah *bit* adalah 2 (2,1), maka probabilitas nilai untuk 2 bit adalah 4 (2,2) berupa 00, 01, 10, dan 11, dan sebagainya. Secara umum, jumlah probabilitas nilai yang terbentuk oleh kombinasi n *bit* adalah sebesar 2n buah.

Data digital adalah metode transmisi yang dapat dibilang rentan terhadap *noise* interferensi elektrik. Semakin jauh data tersebut ditransmisikan secara digital, maka semakin jauh data tersebut ditransmisikan, akan semakin besar interferensi elektrik yang diterima oleh data tersebut. Oleh karena itu transmisi sinyal digital hanya dapat mencapai jarak jangkau yang relatif dekat, yang biasanya dikenal sebagai sinyal diskret.

Namun begitu, transmisi data digital memiliki beberapa kelebihan yang tidak dimiliki oleh transmisi analog, yaitu :

a. Mampu mengirimkan informasi dengan kecepatan yang sangat tinggi sehingga lebih hemat waktu dan lebih efisien.
b. Penggunaan yang berulang-ulang terhadap suatu data tidak akan berdampak pada kualitas data tersebut.
c. Informasi dapat dengan mudah diakses, diproses, dan dimodifikasi ke dalam berbagai bentuk bahkan dapat dikonversikan menjadi analog.
d. Dapat memproses data dalam jumlah yang sangat besar (*big data*) dan dapat mengirimkannya secara interaktif.

2.8 Matlab

Dewasa ini, bahasa pemrograman yang merupakan bahasa yang digunakan manusia untuk berkomunikasi dengan komputer dibuat lebih praktis dan mutakhir untuk mempermudah pengaplikasian perintah manusia terhadap komputer. Perkembangan bahasa Pascal contohnya, yang terus memunculkan varian baru hingga pada akhirnya menjadi Delphi, begitu pula Basic dengan Visual Basic-nya dan C dengan C++ Builder-nya. Pada akhirnya semua bahasa pemrograman akan semakin memanjakan penggunanya dengan penambahan fungsi-fungsi baru yang sangat mudah digunakan bahkan oleh pengguna tingkat pemula sekalipun (Gunaidi, 2006).

Matlab (*Matrix Laboratory*) merupakan salah satu bahasa pemrograman yang dikembangkan oleh MathWorks. Matlab tidak hanya berfungsi sebagai bahasa pemrograman, tetapi sekaligus sebagai alat visualisasi yang berhubungan langsung dengan ilmu matematika. Oleh karena itu, Matlab semakin banyak digunakan oleh para programmer yang menghendaki kepraktisan dalam pembuatan program (Paulus & Nataliani, 2007). Sedangkan (Abidia Away, 2006) menyatakan bahwa Matlab adalah bahasa pemrograman level tinggi (semakin tinggi level bahasa pemrograman semakin mudah cara menggunakannya) yang dikhususkan untuk komputasi teknis.

2.9 *Peak Signal to Noise Ratio* (PSNR)

PSNR adalah sebuah cara untuk mengetahui perubahan yang terjadi antara *original file* dan *stego file* dengan cara menghitung perubahan *bit* yang terjadi di antara keduanya. Sebelum perhitungan PSNR dilakukan, disyaratkan untuk menemukan nilai MSE (*Mean Square Error*). Dengan rumus yang dapat dilihat pada persamaan (2.1) (Bhalshankar & Gulve, 2015):

$$MSE\ (x, y) = \frac{1}{N} \sum_{1=1}^{N} (xi - yi)^2 \tag{2.1}$$

Setelah nilai MSE berhasil didapatkan, dilanjutkan dengan pencarian nilai PSNR dengan cara menghitung nilai maksimal sinyal dibagi dengan nilai MSE. Dengan rumus yang dapat dilihat pada persamaan (2.2) (Bhalshankar & Gulve, 2015):

$$PSNR = 10\log_{10}\left(\frac{255}{MSE}\right) \tag{2.2}$$

Dari perhitungan di atas akan didapatkan nilai PSNR yang kemudian akan digunakan sebagai bahan analisis. Satuan yang digunakan dalam nilai PSNR adalah *decibel* (dB). Nilai PSNR yang baik adalah di atas 30 dB karena jika nilai PSNR berada di bawah 30 dB, maka dapat dipastikan kedua audio tersebut memiliki kesamaan yang sedikit.

BAB III
METODOLOGI DAN PERANCANGAN

3.1 Metode Penelitian

Langkah-langkah penelitian yang dibutuhkan dalam pembuatan penelitian audio steganografi dengan algoritma LSB adalah:

a. Studi literatur

Metode yang digunakan dalam pengumpulan data untuk menyelesaikan penelitian ini adalah dengan metode Kepustakaan yang merupakan pengumpulan data dengan mengkaji literatur yang berkaitan dengan penelitian yang sedang dikerjakan. Data yang digunakan dalam mengerjakan penelitian ini berupa teks yang didapat dari buku, *e-book*, jurnal dalam bentuk *hardcopy* maupun *softcopy*, dan makalah tentang penelitian terkait. Metode kepustakaan merupakan metode yang sangat umum digunakan dalam penelitian karena metode ini merupakan metode yang sangat efektif mengingat peneliti cukup mengumpulkan dan menganalisa data dari penelitian-penelitian yang sudah dilakukan sebelumnya. Selain itu pengumpulan data dengan metode ini dapat menjadi rantai ilmu yang tidak ada habisnya dan menjadikan ilmu tersebut sebagai ladang amal bagi siapapun yang menggunakan dan mengamalkannya.

b. Analisis kebutuhan

Tahap awal untuk mencari dan mengumpulkan data-data yang sekiranya diperlukan untuk penelitian ini. Data utama yang dibutuhkan dalam penelitian ini adalah mengumpulkan file audio untuk dijadikan sebagai *carrier audio*. *File* audio dapat didapatkan dengan merekam langsung sebuah suara atau melalui internet dan *softfile*. File audio yang dibutuhkan adalah file audio berformat WAV.

c. Perancangan perangkat lunak

Ini adalah tahap yang dilakukan untuk mendukung implementasi perangkat lunak. Langkah awal yang harus dilakukan adalah merancang sistem yang akan diterapkan termasuk kebutuhan masukan, pengeluaran, dan proses. Setelah perancangan sistem selesai kemudian dilakukan perancangan terhadap alur program dengan membuat *flowchart*, kemudian merancang desain antarmuka.

d. Impementasi perangkat lunak

Setelah tahap perancangan selesai dilakukan, maka dilanjutkan dengan tahap implementasi perangkat lunak. Tahap ini adalah tahap dimana dilakukannya pengkodean untuk

membuat sistem dan desain yang sudah dirancang pada tahap perancangan perangkat lunak. Pada penelitian ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman Matlab.

e. Analisis dan pengujian sistem

Setelah melalui tahap implementasi dan program sudah selesai dibuat, maka dilanjutkan dengan analisis dan pengujian terhadap program yang telah dibuat. Pada tahap ini akan terlihat hasil dan kekurangan dari program yang telah dibuat. Apabila terdapat kesalahan dalam pengujian, maka dilakukan perbaikan pada tahap perancangan. Setelah dilakukan perbaikan kemudian program akan diuji kembali berulang kali sampai hasil yang diinginkan tercapai. Hasil yang didapatkan kemudian akan dilakukan proses analisis.

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan proses penentuan kebutuhan atau kondisi yang harus dipenuhi dalam merancang dan membangun aplikasi audio steganografi dengan algoritma LSB.

3.2.1 Analisis Kebutuhan Masukan

Kebutuhan masukan adalah kebutuhan atas data-data yang akan diolah sehingga menghasilkan suatu informasi yang dapat diambil dan disimpulkan. Kebutuhan masukan untuk peneltian ini adalah:

a. Proses *embedding*

Hal yang dibutuhkan dalam proses *embedding* adalah *file carrier audio* dan pesan rahasia. Untuk *carrier audio*, dalam penelitian ini akan digunakan 5 *file* audio berformat WAV yang memiliki karakteristik suara yang berbeda satu sama lainnya. *File* audio pertama merupakan suara tilawah Al-Qur'an surat Al-Fatihah yang merupakan surat pembuka dalam kitab suci Al-Qur'an. *File* audio kedua merupakan potongan dari pidato Bung Tomo. *File* audio ketiga merupakan potongan dari sebuah lagu bergenre *soul-funk* berjudul In the stone. *File* audio keempat merupakan potongan musik klasik berjudul Nocturne karya Chopin. *File* audio kelima adalah potongan lagu *rock* berjudul Sweet Child O'Mine. Masing-masing *file* audio di atas berformat WAV berukuran 1,24Mb. Sedangkan untuk pesan rahasia yang akan disisipkan adalah teks berupa barisan angka berukuran 100 *bytes*.

b. Proses *extracting*

Kebutuhan masukan pada proses *extracting* adalah *stego audio* yang kemudian diproses dengan algoritma pada program untuk mendapatkan data rahasia di dalamnya. *Stego audio* adalah *file* audio yang di dalamnya terdapat pesan tersembunyi.

c. Proses hitung PSNR

Kebutuhan masukan pada proses hitung PSNR adalah *file carrier audio* dan *stego audio.*

3.2.2 Analisis Kebutuhan Keluaran

Data-data yang dihasilkan dalam kebutuhan keluaran proses di atas adalah :

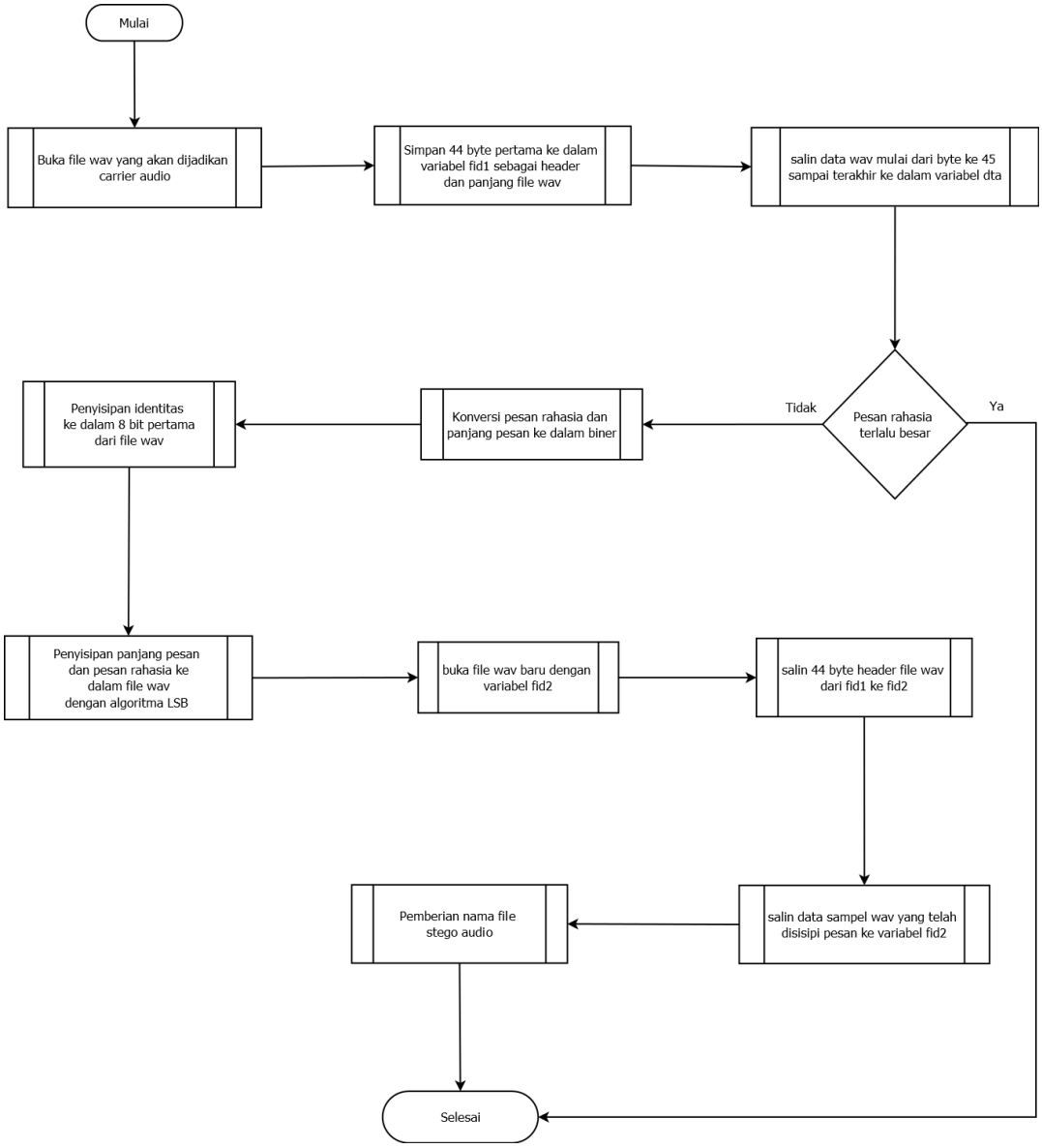
a. *Stego audio* atau *file* audio yang di dalamnya terdapat pesan rahasia setelah melalui proses penyisipan pesan.
b. Pesan rahasia yang didapatkan dari proses ekstraksi pesan.
c. Nilai PSNR yang didapat dari proses hitung PSNR.

3.3 Diagram Alir

Diagram alir atau yang biasa disebut *flowchart* adalah sebuah alur program yang terdiri dari berbagai simbol yang masing-masing simbol merepresentasikan aksi dalam suatu proses untuk menyelesaikan permasalahan dari awal program sampai pada akhir program.

3.3.1 ***Flowchart***

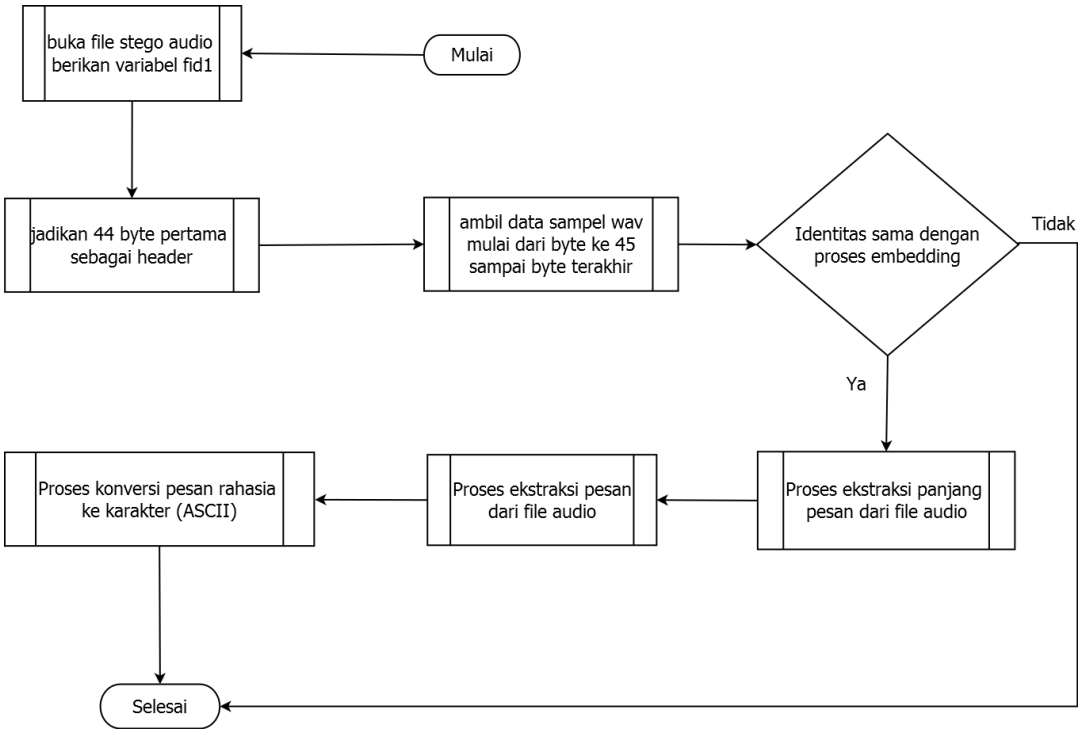
3.3.1.1 *Flowchart embedding*



Gambar 3.1 *Flowchart embedding*

Gambar 3.1 di atas merupakan diagram yang menggambarkan alur proses yang terjadi saat proses *embedding* dilakukan. Dari gambar di atas juga dapat diketahui bahwa untuk melakukian proses *embedding* diperlukan 2 input berupa *file* audio untuk dijadikan *carrier audio* dan pesan rahasia yang akan disisipkan ke dalam *file* audio dan beberapa proses untuk menyelesaikan proses *embedding* tersebut.

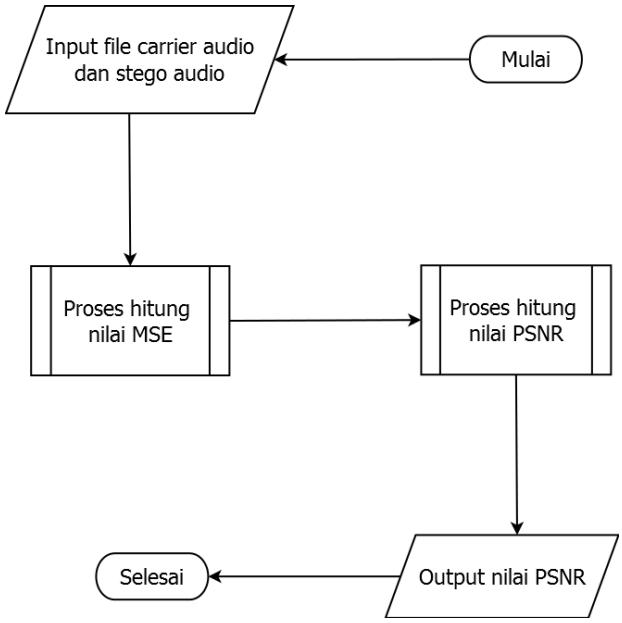
3.3.1.2 *Flowchart extracting*



Gambar 3.2 *Flowchart extracting*

Gambar 3.2 menampilkan sebuah diagram alir yang menggambarkan alur proses yang terjadi pada saat proses *extracting* dilakukan. Dari gambar di atas juga dapat diketahui bahwa untuk melakukan proses *extracting* diperlukan satu masukan berupa *stego audio* yang akan diproses.

3.3.1.3 *Flowchart* PSNR



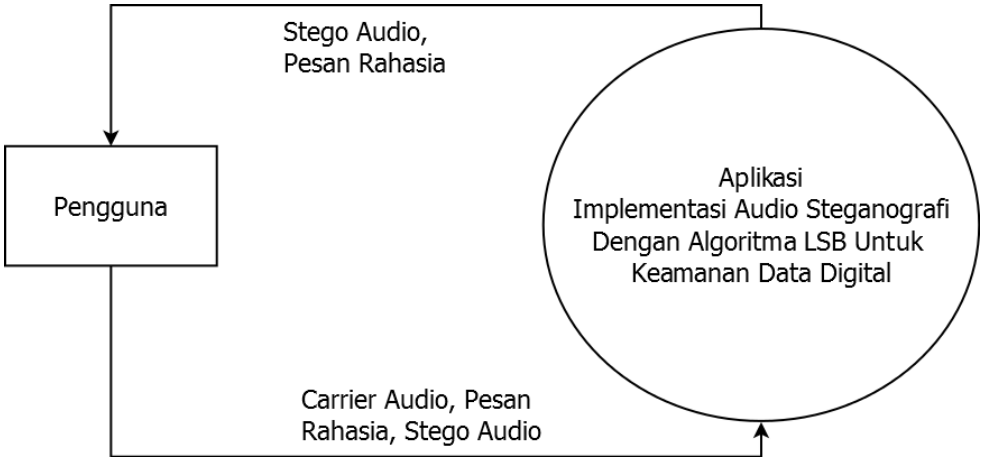
Gambar 3.3 *Flowchart* hitung PSNR

Gambar 3.3 menampilkan sebuah diagram alir yang menggambarkan alur proses yang terjadi pada saat proses hitung PSNR dilakukan. Dari gambar di atas juga dapat diketahui bahwa untuk melakukan proses hitung PSNR diperlukan masukan berupa *file carrier audio* dan *stego audio* yang akan diproses.

3.3.2 *Data Flow Diagram* (DFD)

DFD merupakan suatu diagram yang menggambarkan proses kerja dari suatu sistem. DFD bertujuan untuk mengetahui alur kerja sistem dengan menunjukkan data yang dibutuhkan sebagai masukan sehingga akan menghasilkan keluaran yang diinginkan.

3.3.2.1 DFD Level 0

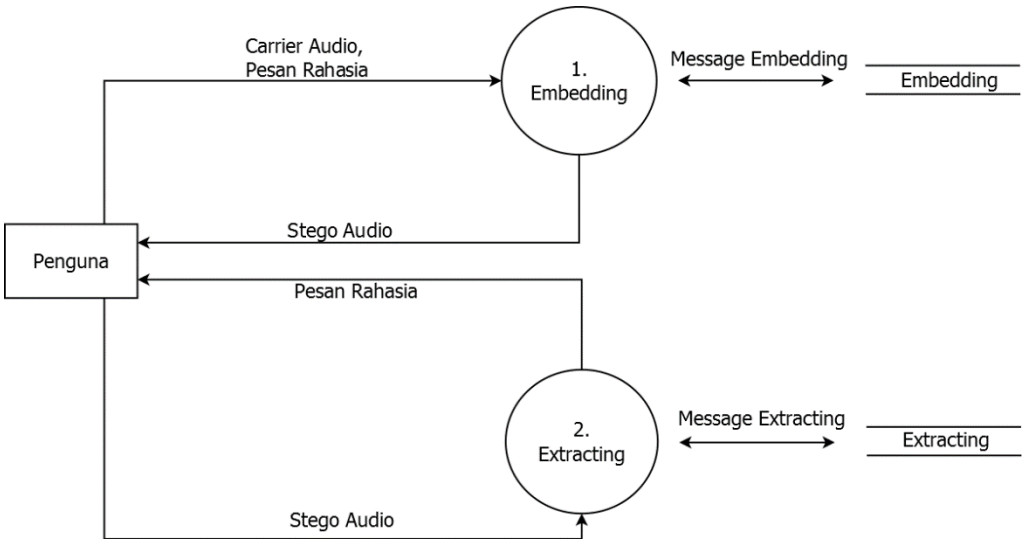


Gambar 3.4 DFD Level 0

Gambar 3.4 menunjukkan diagram DFD level 0 yang menjelaskan tentang interaksi dua arah antara pengguna dengan sistem. Pada diagram di atas dijelaskan masukan-masukan yang dibutuhkan oleh sistem untuk dapat menghasilkan keluaran yang diinginkan oleh pengguna.

Adapun masukan yang dibutuhkan oleh sistem adalah *carrier audio* yaitu *file* audio yang akan dijadikan media penyisipan pesan rahasia, pesan rahasia yang akan disisipkan, dan *stego audio* yang akan diekstraksi. Adapun keluaran yang diinginkan oleh pengguna adalah periodogram sebagai visualisasi sinyal audio, *stego audio* yang berisi pesan rahasia hasil proses *embedding*, dan pesan rahasia yang telah diekstraksi dari *file stego audio* hasil dari proses *extracting*.

3.3.2.2 DFD Level 1



Gambar 3.5 DFD Level 1

Gambar 3.5 menjelaskan proses yang terjadi di dalam sistem yaitu proses *embedding* yang merupakan proses penyisipan pesan rahasia ke dalam *carrier audio* dan *extracting* yang merupakan proses pemisahan pesan rahasia dari *stego audio*. Pada kedua proses tersebut membutuhkan masukan, keluaran dan penyimpanan masing-masing sebagai berikut :

a. Proses *Embedding* :
   1. Masukan : *Carrier audio* berupa *file* audio yang akan dijadikan media penyisipan pesan rahasia, pesan rahasia yang akan disisipkan ke dalam *carrier audio*.
   2. Keluaran : Periodogram sebagai visualisasi *carrier audio*, *stego audio* sebagai hasil dari penyisipan pesan rahasia ke dalam *carrier audio.*
   3. Penyimpanan : *Embedding*

b. Proses *Extracting*
   1. Masukan : *Stego audio* yang akan dilakukan proses ekstraksi pesan rahasia
   2. Keluaran : Periodogram sebagai visualisasi *stego audio,* dan pesan rahasia hasil ekstraksi dari *stego audio.*
   3. Penyimpanan : *Extracting*

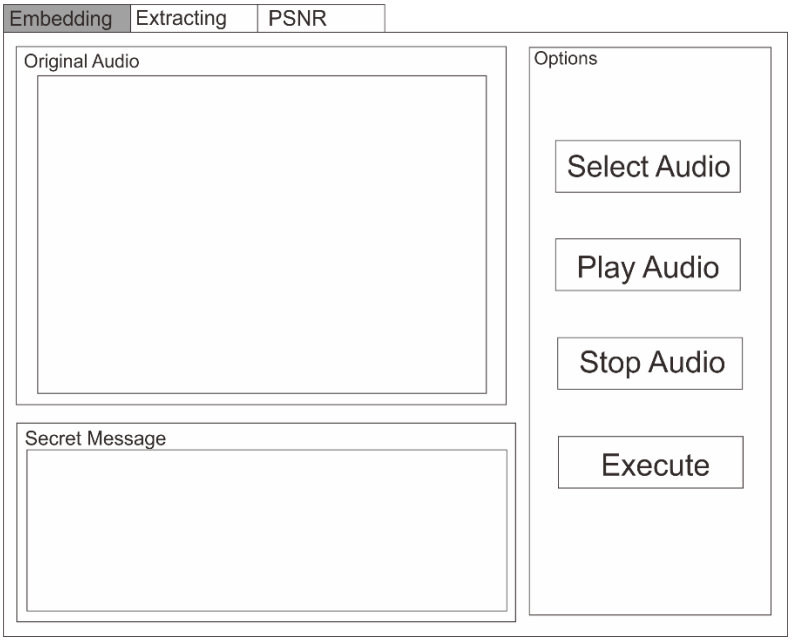
3.4 Perancangan Desain Antarmuka

Perancangan desain antarmuka adalah suatu proses untuk mendesain tampilan antarmuka untuk pengguna yang berfokus pada *user experience* dan interaksi. Proses ini bertujuan untuk

membuat interaksi pengguna dan sistem sesederhana dan seefisien mungkin untuk mencapai tujuan pengguna.

Dalam penelitian ini, perancangan desain antarmuka menggunakan 1 halaman dengan 3 *tab* yang berbeda.

3.4.1 Desain antarmuka proses *embedding*



Gambar 3.6 Rancangan antarmuka proses *embedding*

Gambar 3.6 di atas merupakan rancangan dari desain antarmuka untuk proses *embedding.* Proses *embedding* merupakan proses untuk menyisipkan pesan rahasia ke dalam *carrier audio.* proses ini memiliki beberapa properti yang terdiri dari *pusbutton, edit text,* dan grafik dengan penjelasan sebagai berikut :

a. Panel *Original Audio*

   Dalam panel ini terdapat properti grafik yang akan menampilkan visualisasi dari sinyal *carrier audio* berupa spectrogram.

b. Panel *Secret Message*

   Dalam panel ini terdapat properti *edit text* yang akan menampung pesan rahasia yang kemudian akan disisipkan ke dalam *carrier audio*.

d. Panel *Options*

   Dalam panel ini terdapat beberapa properti *pushbutton* yang masing-masing berfungsi sebagai berikut :

1. *Pushbutton Select Audio*

   Tombol ini berfungsi untuk memilih *file carrier audio* oleh pengguna.

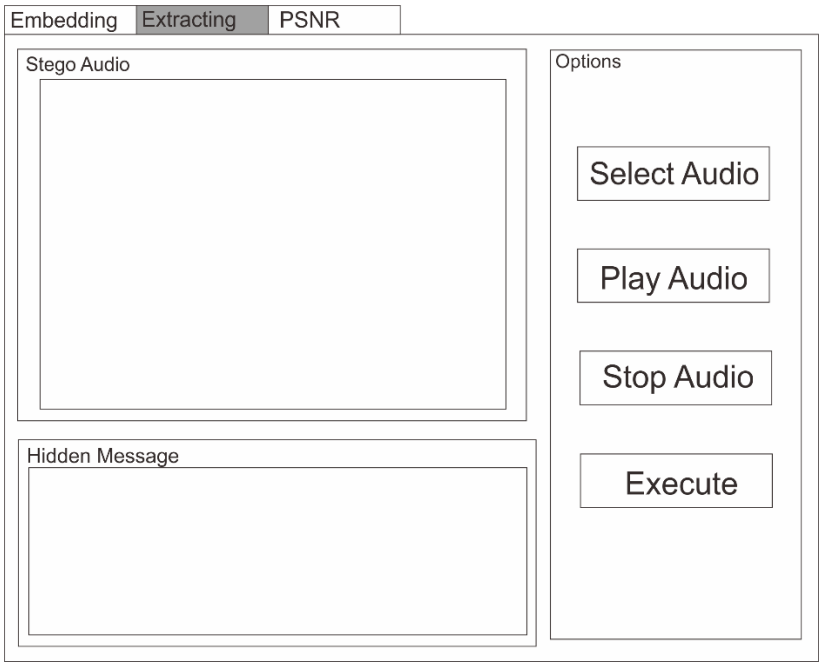
2. *Pushbutton Play Audio*

   Tombol ini berfungsi untuk memainkan *file audio* yang tadi sudah dipilih oleh pengguna.

3. *Pushbutton Hide Text*

   Tombol ini berfungsi untuk mengeksekusi proses *embedding* terhadap *carrier audio* dan pesan rahasia yang telah dimasukkan sebelumnya.

3.4.2 Desain antarmuka proses *extracting*



Gambar 3.7 Rancangan antarmuka proses *extracting*

Gambar 3.7 di atas merupakan rancangan dari desain antarmuka untuk proses *extracting.* Proses *extracting* merupakan proses untuk memisahkan atau mengekstrak pesan rahasia dari *stego audio.* proses ini memiliki beberapa properti yang terdiri dari *pusbutton, edit text,* dan grafik dengan penjelasan sebagai berikut :

a. Panel *Stego Audio*

   Dalam panel ini terdapat properti grafik yang akan menampilkan visualisasi dari sinyal *stego audio* berupa spectrogram.

b. Panel *Hidden Message*

Dalam panel ini terdapat properti *edit text* yang akan menampilkan pesan rahasia yang sudah diekstrak dari *stego audio*.

e. Panel *Options*

Dalam panel ini terdapat beberapa properti *pushbutton* yang masing-masing berfungsi sebagai berikut :

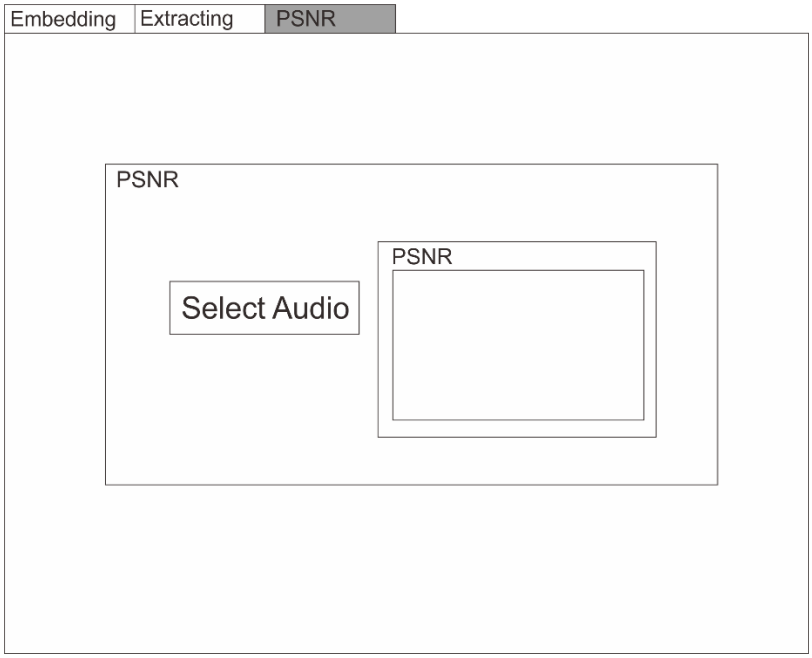
1. *Pushbutton Select Audio*

   Tombol ini berfungsi untuk memilih *file stego audio* yang dikehendaki oleh pengguna untuk diekstraksi.

2. *Pushbutton Play Audio*

   Tombol ini berfungsi untuk memainkan *file audio* yang tadi sudah dipilih oleh pengguna.

3.4.3 Desain antarmuka proses hitung PSNR



Gambar 3.8 Rancangan antarmuka proses hitung PSNR

Gambar 3.8 di atas merupakan rancangan dari desain antarmuka untuk proses hitung PSNR. Proses hitung PSNR adalah proses yang menghitung sinyal noise dari dua buah sinyal audio untuk kemudian nilai tersebut dapat dijadikan sebagai acuan derajat kesamaan antara kedua sinyal audio tersebut. proses ini memiliki 2 properti yaitu *pusbutton* dan *edit text* dengan penjelasan sebagai berikut :

a. *Pushbutton Select Audio*

Tombol ini berfungsi untuk memilih dua *file* audio yang akan dihitung nilai PSNR nya.

b. Panel *PSNR*

Dalam panel ini terdapat properti *edit text* yang akan menampilkan nilai PSNR yang telah dihitung.

3.5 Rancangan Pengujian

Pengujian adalah suatu proses untuk memastikan aplikasi berjalan dengan baik dan mencari celah dimana kesalahan mungkin terjadi dalam suatu sistem aplikasi. Pada penelitian ini akan dilakukan pengujian pada proses *embedding, extracting,* dan perhitungan PSNR.

3.5.1 Rancangan pengujian proses *embedding*

Tabel 3.1 Daftar *file* audio yang akan dijadikan *carrier audio*

No.	Nama File	Ukuran
1	Al-Fatihah.wav	1,24 MB (1.306.186 *bytes*)
2	Bung Tomo.wav	1,24 MB (1.306.186 *bytes*)
3	In the stone.wav	1,24 MB (1.306.186 *bytes*)
4	Nocturne.wav	1,24 MB (1.306.186 *bytes*)
5	Sweet child o mine.wav	1,24 MB (1.306.186 *bytes*)

Pengujian pada proses *embedding* bertujuan untuk memastikan proses tersebut berjalan sesuai dengan apa yang diinginkan dan juga untuk mencari kekurangan yang terdapat dalam proses tersebut. Pengujian pada proses *emebedding* meliputi:

a. Melakukan proses *embedding* pesan rahasia terhadap semua masukan yang telah ditentukan.
b. Menguji semua tombol yang terdapat pada *tab* proses *embedding.*
c. Menguji proses *embedding*.
d. Menguji proses penyimpanan hasil *embedding.*

3.5.2 Rancangan pengujian proses *extracting*

Pengujian pada proses *extracting* bertujuan untuk memastikan proses tersebut berjalan sesuai dengan apa yang diinginkan dan juga untuk mencari kekurangan yang terdapat dalam proses tersebut. Pengujian pada proses *extracting* meliputi :

a. Melakukan proses *extracting* pesan tersembunyi dari *file* masukan yang telah melalui proses *embedding*.
b. Menguji semua tombol yang terdapat pada *tab* proses *extracting.*
c. Menguji jalannya proses *extracting.*

3.5.3 Rancangan pengujian proses hitung PSNR

Pengujian pada proses hitung PSNR dimaksudkan untuk memastikan proses tersebut berjalan sesuai dengan apa yang diinginkan dan juga untuk mencari kekurangan dan kesalahan yang terdapat dalam proses tersebut. Pengujian pada proses hitung PSNR meliputi :

a. Melakukan proses perhitungan PSNR terhadap dua audio setelah dan sebelum dilakukannya proses *embedding.*
b. Menguji semua tombol yang terdapat pada *tab* proses hitung PSNR.
c. Menguji jalannya proses hitung PSNR.

BAB IV
IMPLEMENTASI HASIL DAN ANALISIS

Pada bab ini akan dipaparkan penjelasan mengenai hasil dari implementasi rancangan yang telah dibuat dan analisis pengujian aplikasi audiosteganografi dengan algoritma LSB. Yang akan dibahas dalam bab ini meliputi implementasi, uji kinerja aplikasi, dan analisis hasil pengujian.

4.1 Implementasi

Pada subbab ini akan dijelaskan hasil implementasi atau penerapan yang telah dilakukan terhadap rancangan-rancangan dan teori-teori yang telah dipaparkan pada bab-bab sebelumnya.

4.1.1 Lingkungan implementasi

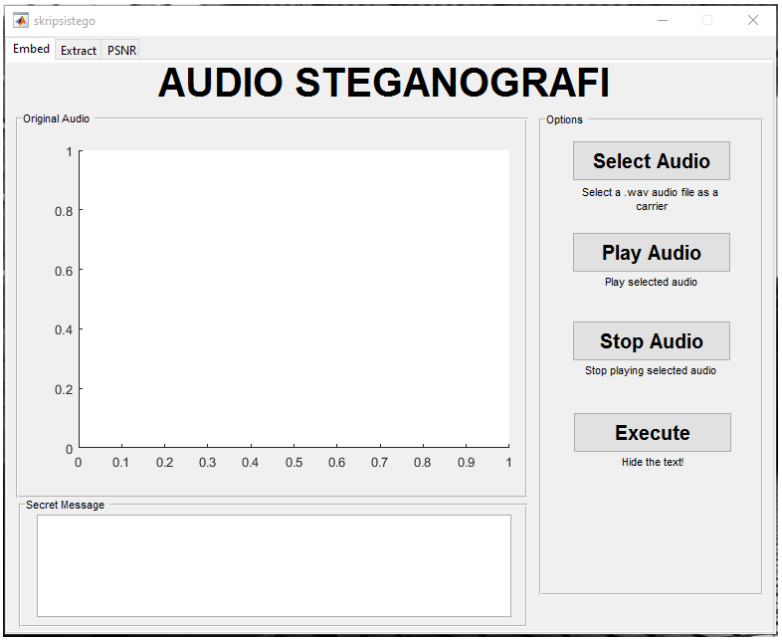
Spesifikasi komponen perangkat keras yang digunakan dalam pembuatan dan pengujian aplikasi audio steganografi dengan algoritma LSB ini adalah sebagai berikut :

a. Perangkat komputasi yang digunakan adalah *laptop* Toshiba Satellite C40.
b. Perangkat masukan berupa *mouse* dan *keyboard* built-in laptop
c. Perangkat keluaran berupa *monitor laptop* dan *speaker*
d. Processor AMD APU E1
e. Memori RAM DDR3 2 GB
f. Harddisk 500 GB

Sementara perangkat lunak yang digunakan dalam penelitian ini memiliki spesifikasi sebagai berikut :

a. Sistem operasi Microsoft Windows 10 Professional 64-bit
b. Perangkat lunak pemrograman Matlab R2016a
c. Perangkat lunak *audio editor* FL Studio 12 Producer Edition

4.1.2 Implementasi antarmuka

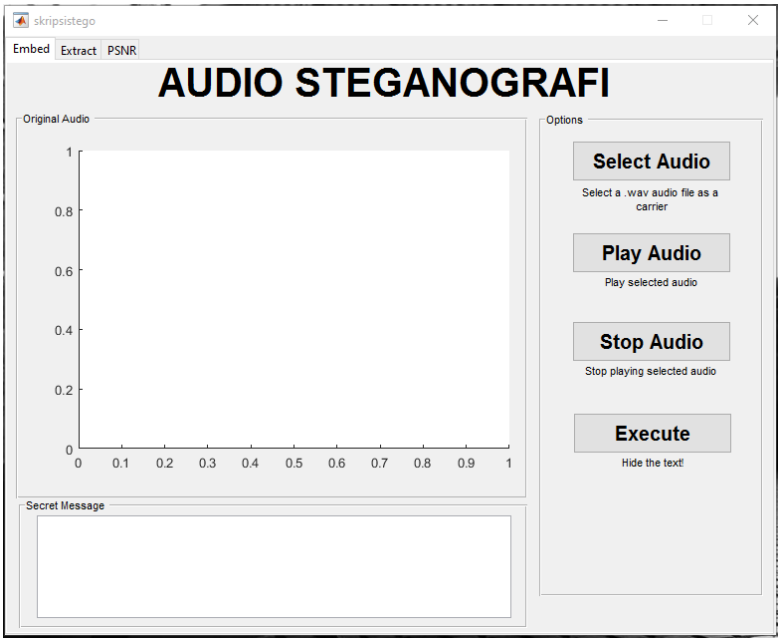


Gambar 4.1 Tampilan awal aplikasi

Gambar 4.1 merupakan tampilan antarmuka ketika aplikasi pertama kali mulai dijalankan. Tampilan ini juga merupakan tampilan untuk melakukan proses *embedding.*

4.1.2.1 Proses *embedding*

Proses *embedding* merupakan proses penyisipan pesan rahasia ke dalam *carrier audio.* Berikut merupakan tampilan antarrmuka ketika proses *embedding* dijalankan.



Gambar 4.2 Tampilan awal proses *embedding*

Gambar 4.2 merupakan tampilan dari proses *embedding.* Tampilan di atas memiliki beberapa properti yang dibagi menjadi 3 buah panel dengan penjelasan sebagai berikut :

a. Panel *Original Audio*

Dalam panel ini terdapat properti grafik yang akan menampilkan visualisasi dari sinyal *carrier audio* berupa periodogram.

b. Panel *Secret Message*

Dalam panel ini terdapat properti *edit text* yang akan menampung pesan rahasia yang kemudian akan disisipkan ke dalam *carrier audio*.

c. Panel *Options*

Dalam panel ini terdapat beberapa properti *pushbutton* yang masing-masing berfungsi sebagai berikut :

1. *Pushbutton Select Audio*

   Tombol ini berfungsi untuk memilih *file carrier audio* oleh pengguna.

2. *Pushbutton Play Audio*

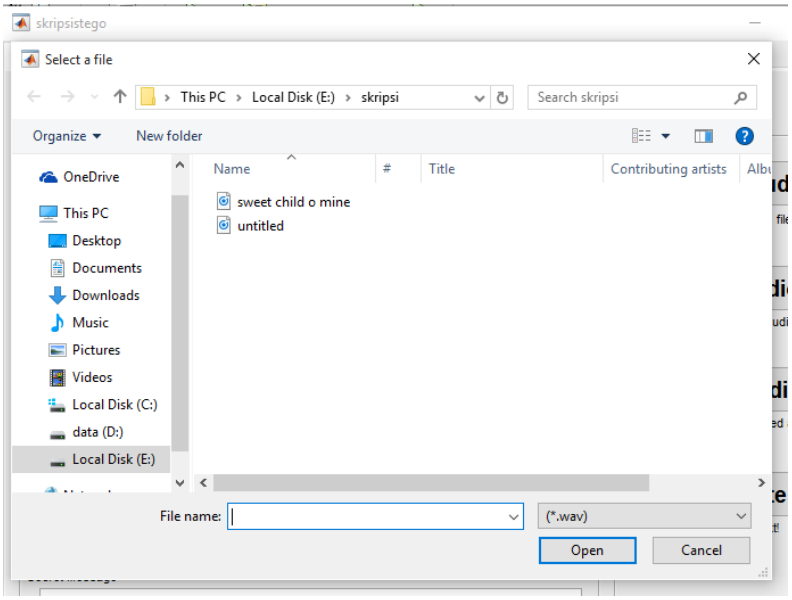
   Tombol ini berfungsi untuk memainkan *file audio* yang tadi sudah dipilih oleh pengguna.

3. *Pushbutton Stop Audio*

   Tombol ini berfungsi untuk menghentikan *file audio* yang sedang dimainkan.

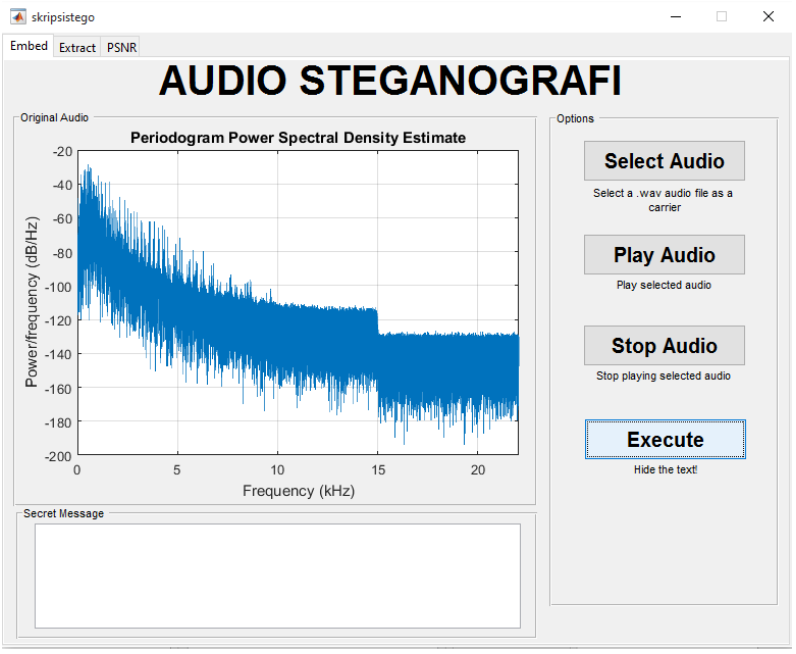
4. *Pushbutton Execute*

Tombol ini berfungsi untuk mengeksekusi proses *embedding* terhadap *carrier audio* dan pesan rahasia yang telah dimasukkan sebelumnya.



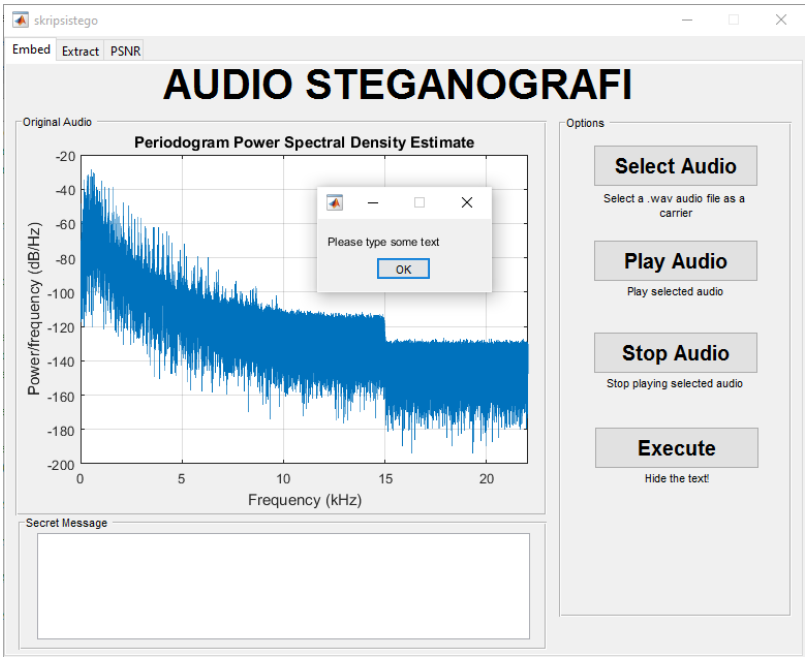
Gambar 4.3 Tampilan proses memilih file *carrier audio*

Gambar 4.3 merupakan tampilan setelah tombol *select audio* pada proses *embedding* ditekan untuk kemudian pengguna diarahkan untuk memilih sebuah file audio untuk dijadikan *carrier audio.*



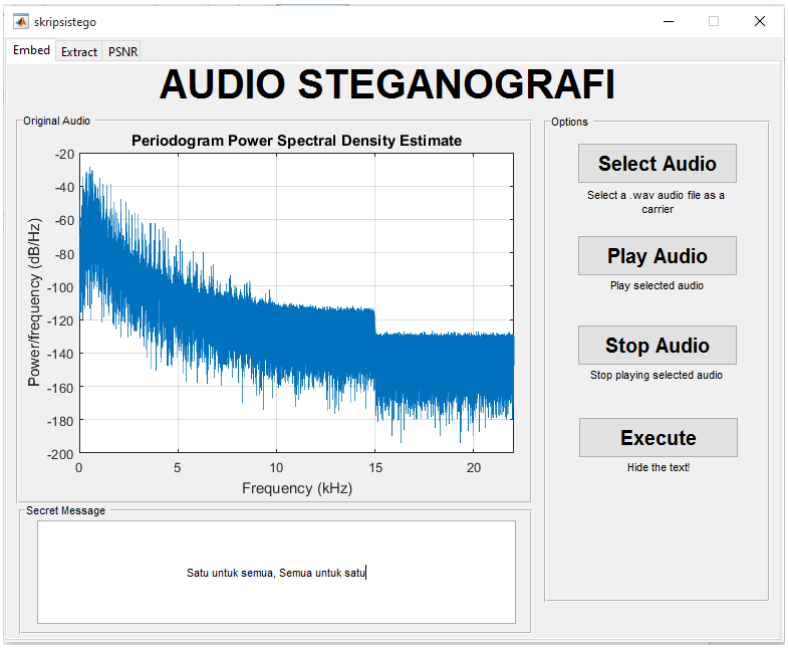
Gambar 4.4 Tampilan periodogram pada proses *embedding*

Gambar 4.4 merupakan tampilan visualisasi sinyal audio berupa periodogram dari *file* audio yang sebelumnya telah dipilih oleh pengguna dalam proses *embedding*.



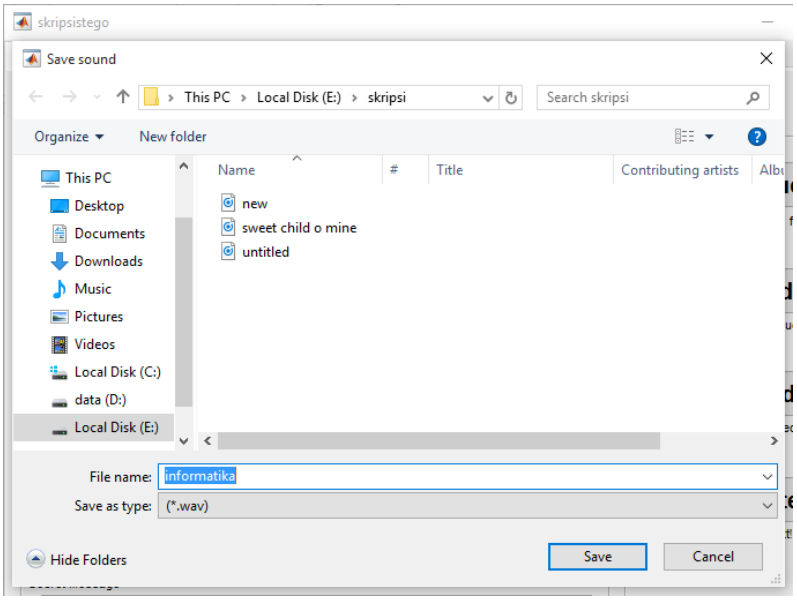
Gambar 4.5 Tampilan *pop up message*

Gambar 4.5 merupakan tampilan dari *pop up message* yang mumncul ketika pengguna menekan tombol *execute* sebelum memasukkan pesan rahasia yang akan disembunyikan.



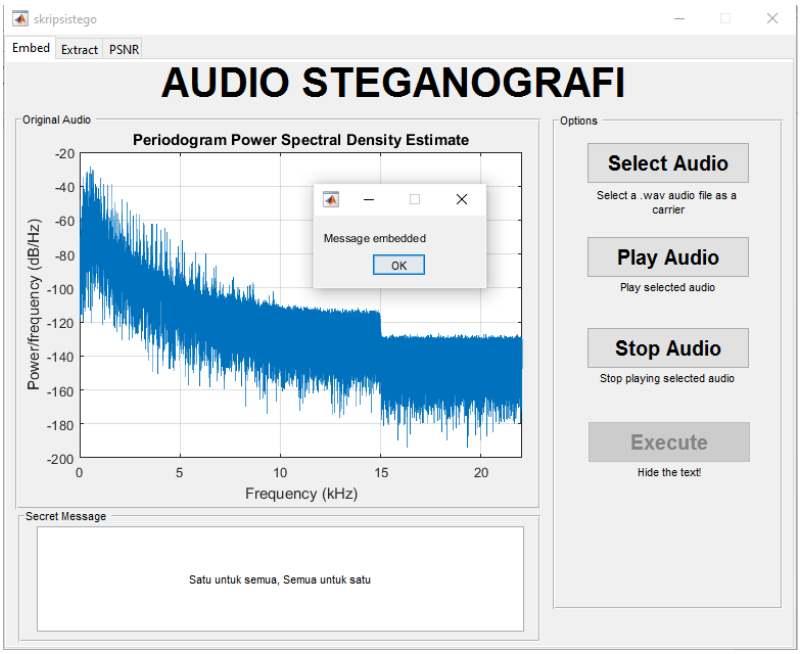
Gambar 4.6 Tampilan proses *embedding*

Gambar 4.6 menunjukkan tampilan pada proses *embedding* setelah pengguna memilih *file carrier audio* dan telah memasukkan pesan yang akan disembunyikan.



Gambar 4.7 Tampilan proses pemberian nama untuk hasil *embedding*

Gambar 4.7 merupakan tampilan setelah tombol *execute* ditekan. kemudian pengguna akan diarahkan untuk memberi nama *file stego audio* sesuai dengan nama yang diinginkan.

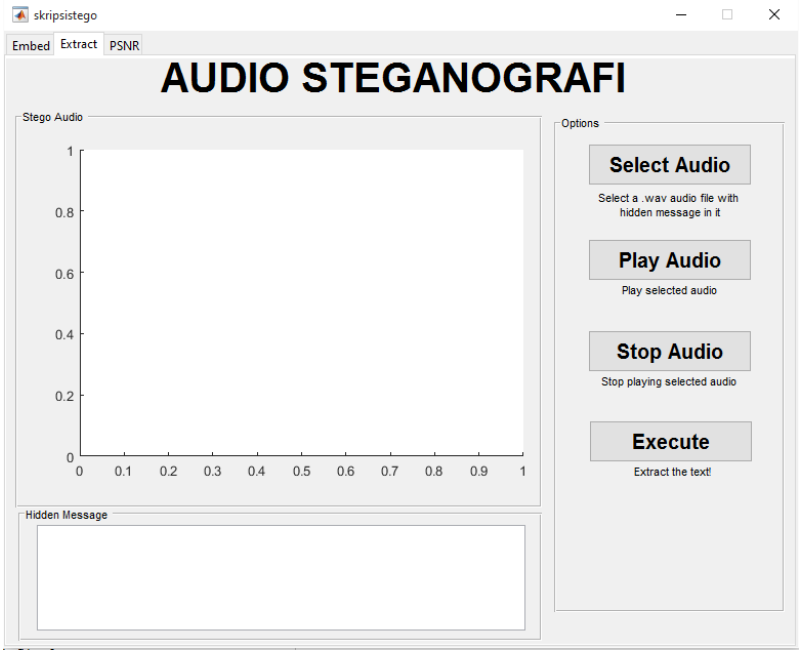


Gambar 4.8 Tampilan *pop up message* setelah proses *embedding* berhasil dilakukan

Gambar 4.8 merupakan tampilan setelah penyisipan pesan rahasia ke dalam sinyal audio pada proses *embedding* selesai dilakukan.

4.1.2.2 Proses *extracting*

Proses *extracting* merupakan proses pengambilan pesan rahasia dari dalam *stego audio*. Berikut merupakan tampilan antarrmuka ketika proses *extracting* dijalankan.



Gambar 4.9 Tampilan awal prose *extracting*

Gambar 4.9 merupakan tampilan dari proses *extracting*. Tampilan di atas memiliki beberapa properti yang dibagi menjadi 3 buah panel dengan penjelasan sebagai berikut :

a. Panel *Stego Audio*

Dalam panel ini terdapat properti grafik yang akan menampilkan visualisasi dari sinyal *carrier audio* berupa periodogram.

b. Panel *Hidden Message*

Dalam panel ini terdapat properti *edit text* yang akan menampilkan pesan rahasia yang berada di dalam *stego audio*.

c. Panel *Options*

Dalam panel ini terdapat beberapa properti *pushbutton* yang masing-masing berfungsi sebagai berikut :

1. *Pushbutton Select Audio*

   Tombol ini berfungsi untuk memilih *file stego audio* oleh pengguna.

2. *Pushbutton Play Audio*

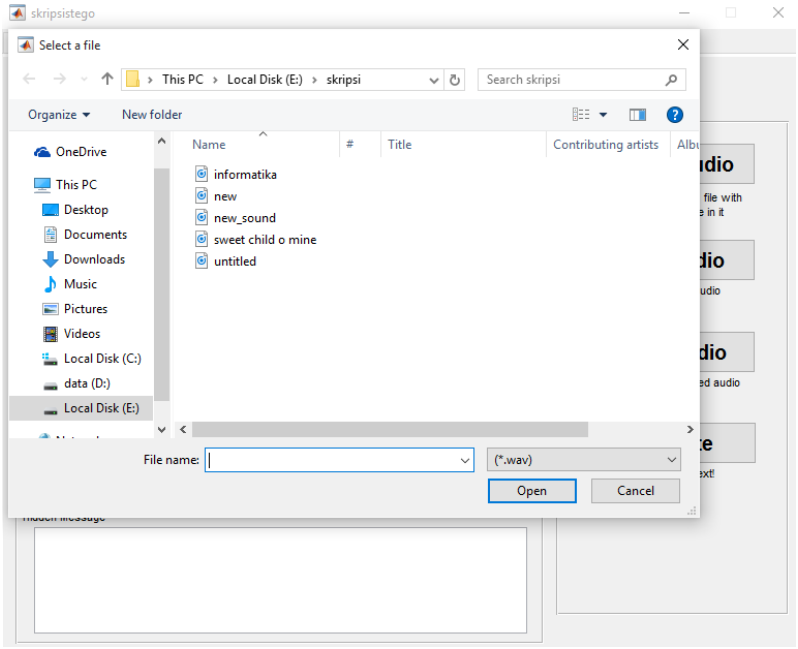
   Tombol ini berfungsi untuk memainkan *file audio* yang tadi sudah dipilih oleh pengguna.

3. *Pushbutton Stop Audio*

   Tombol ini berfungsi untuk menghentikan *file audio* yang sedang dimainkan.

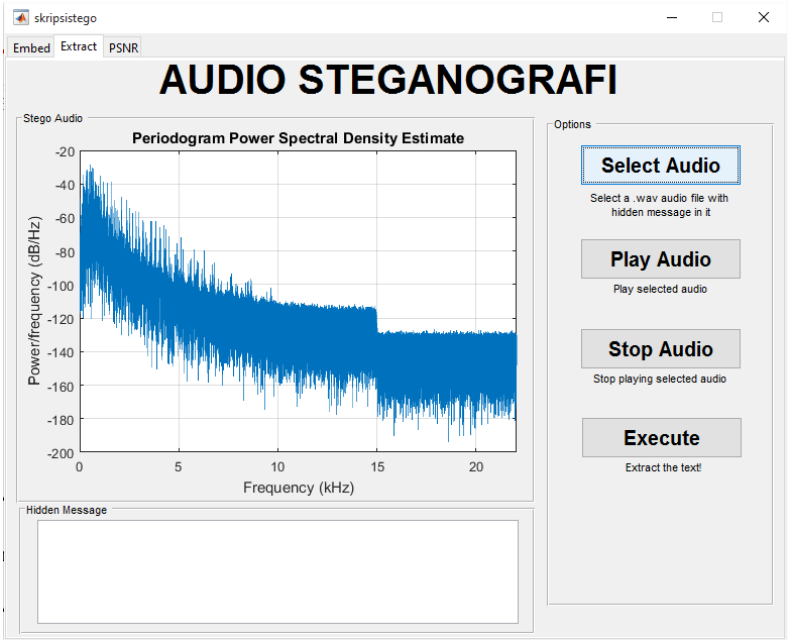
4. *Pushbutton Execute*

   Tombol ini berfungsi untuk mengeksekusi proses *extracting* terhadap *stego audio* untuk menampilkan pesan rahasia yang telah di sembunyikan.



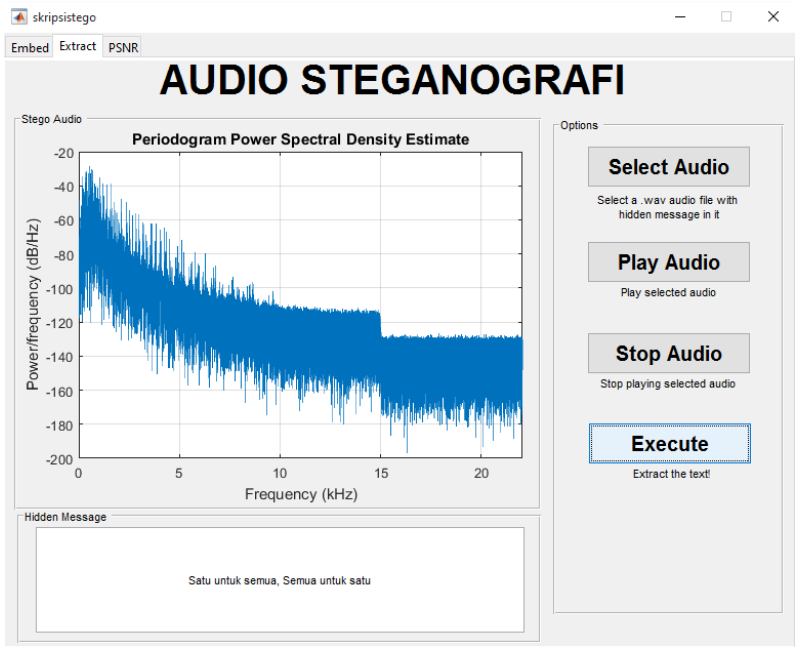
Gambar 4.10 Tampilan proses pemilihan *stego audio*

Gambar 4.10 merupakan tampilan setelah tombol *select audio* ditekan yang kemudian pengguna diminta untuk memilih file *stego audio* yang akan dijadikan objek proses *extracting.*



Gambar 4.11 Tampilan periodogram pada proses *extracting*

Gambar 4.11 merupakan tampilan visualisasi sinyal audio berupa periodogram dari *stego audio* yang telah dipilih oleh pengguna pada proses *extracting*.

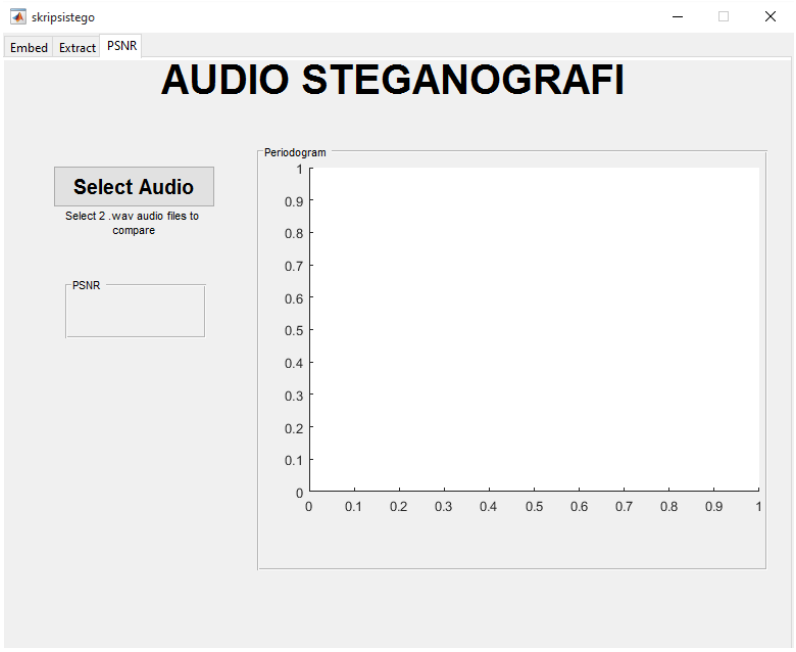


Gambar 4.12 Tampilan proses *extracting*

Gambar 4.12 merupakan tampilan pada proses *extracting* setelah pesan rahasia berhasil diekstrak dari *file stego audio.*

4.1.2.3 Proses hitung PSNR

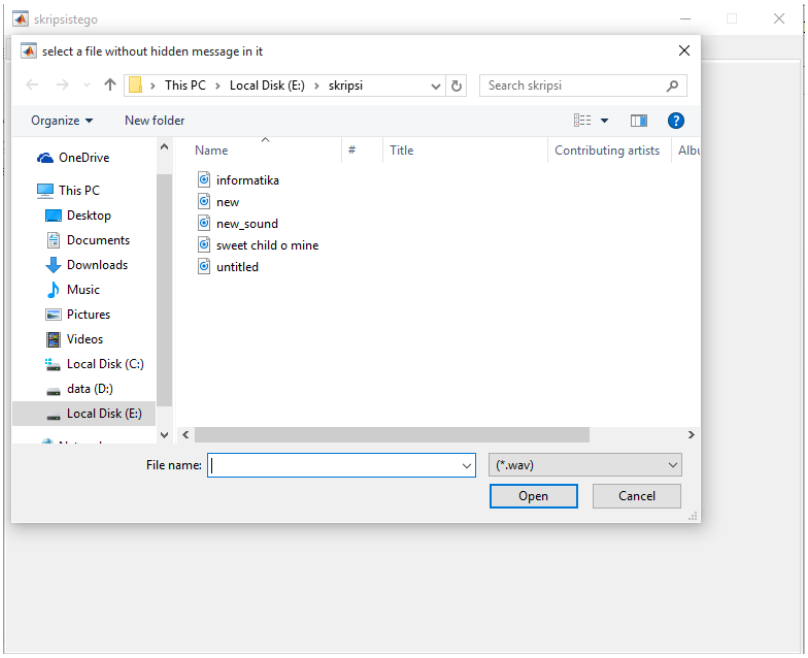
Proses hitung PSNR merupakan proses untuk menghitung perbedaan yang terdapat antara sinyal *carrier audio* dan *stego audio*. Berikut merupakan tampilan antarrmuka ketika proses hitung PSNR dijalankan.



Gambar 4.13 Tampilan awal proses hitung PSNR

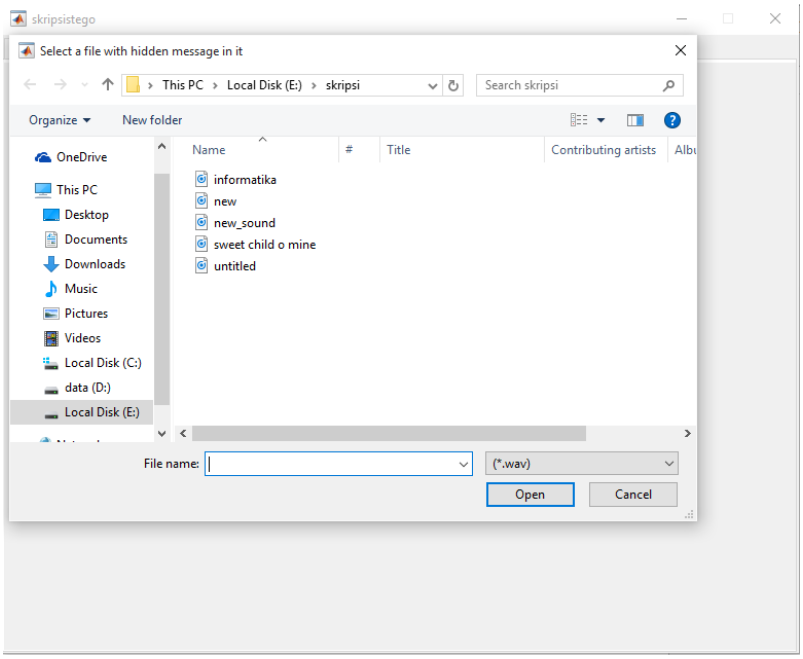
Gambar 4.13 merupakan tampilan antarmuka dari proses hitung PSNR. Antarmuka di atas memiliki beberapa properti sebagai berikut :

a. *Pushbutton Select audio :* Berfungsi untuk memilih *file carrier audio* dan *stego audio* yang akan dihitung nilai PSNR nya.
b. *Static text* : Berfungsi untuk menampilkan hasil perhitungan nilai PSNR.
c. *Axes* : Berfungsi untuk menampilkan perbandingan periodogram.



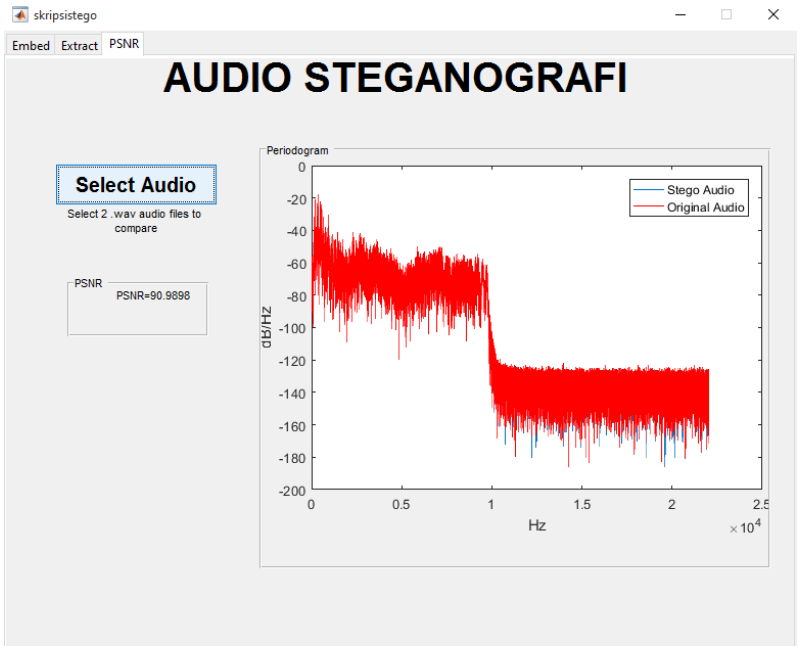
Gambar 4.14 Tampilan proses pemilihan *carrier audio*

Gambar 4.14 merupakan tampilan setelah tombol *select audio* ditekan. Pengguna akan diminta untuk memilih file *carrier audio* yang akan dihitung nilai PSNR nya.



Gambar 4.15 Tampilan proses pemilihan *stego audio*

Gambar 4.15 merupakan tampilan dari proses pemilihan *stego audio* yang akan dihitung nilai PSNR nya.



Gambar 4.16 Tampilan proses hitung PSNR

Gambar 4.16 merupakan tampilan dari proses hitung PSNR setelah selesai dilakukan perhitungan dan periodogram telah ditampilkan.

4.1.3 *Source Code* untuk Membangun Implementasi

Dalam proses implementasi aplikasi audio steganografi ini, dilakukan proses konversi dari teori dan rancangan ke dalam aplikasi matlab. Berikut adalah *source code* yang digunakan dalam proses konversi sehingga teori dan rancangan yang telah dibuat dapat dijalankan sebagai aplikasi Matlab.

4.1.3.1 Proses *embedding*

```
1. function pushbtnselectaudio_Callback(hObject, eventdata, handles)
2. % hObject    handle to pushbtnselectaudio (see GCBO)
3. % eventdata  reserved - to be defined in a future version of MATLAB
4. % handles    structure with handles and user data (see GUIDATA)
5. [handles.fname, handles.pname] = uigetfile('*.wav','Select a file');
6. %tampilan periodogram
7. axes(handles.axes1);
8. [signal,fs] = audioread(handles.fname);
9. signal = signal(:,1);
10. plot(psd(spectrum.periodogram,signal,'Fs',fs,'NFFT',length(signal)));
11. guidata(hObject, handles); %update handle structure
```

Gambar 4.17 *Source code* tombol *select audio*

Gambar 4.17 merupakan *source code* tombol *select audio* yang terdapat pada proses *embedding* yang diberi *tag* pushbtnselectaudio. Pada baris ke 5 terdapat kode untuk memilih

file audio WAV dari folder yang terdapat pada *path* Matlab dengan menggunakan perintah uigetfile. Pada beris ke 7 sampai baris ke 10 merupakan perintah untuk mengubah sinyal audio menjadi periodogram dan menampilkannya pada properti axes.

```
1.  function pushbtnexecute_Callback(hObject, eventdata, handles)
2.  % hObject    handle to pushbtnexecute (see GCBO)
3.  % eventdata  reserved - to be defined in a future version of MATLAB
4.  % handles    structure with handles and user data (see GUIDATA)
5.  msg=get(handles.edit1,'string');
6.      if isempty(msg)
7.          msgbox('Please type some text','Empty');
8.      else
9.          [y, fs]=audioread([handles.pname handles.fname],[1 2]);
10.         %buka file carrier audio
11.         fid1=fopen([handles.pname handles.fname],'r');
12.         %40 bytes pertama disimpan sebagai header
13.         header=fread(fid1,40,'uint8=>char');
14.         %byte ke 41 sampai bite ke 43, panjang dari sampel wav
15.         data_size=fread(fid1,1,'uint32');
16.         %salin data wav 16 bit mulai dari byte ke 44
17.         [dta,count]=fread(fid1,inf,'uint16');
18.         fclose(fid1);
19.         lsb=1;
```

Gambar 4.18 *Source code* tombol *execute*

Gambar 4.18 merupakan *source code* tombol *execute* yang terdapat pada proses *embedding* yang diberi *tag* pushbtnexecute. Pada baris kelima terdapat kode perintah untuk mengambil pesan rahasia yang terdapat pada properti edit1. Pada baris keenam terdapat percabangan dengan dua kondisi : Jika edit1 kosong (belum terdapat pesan rahasia), maka aplikasi akan memunculkan *pop up message* berupa peringatan untuk mengisi kolom pesan rahasia. Tapi jika edit1 terisi (sudah ada pesan rahasia), maka aplikasi akan dilanjutkan ke proses pengambilan *file carrier audio* yang sudah dipilih sebelumnya untuk kemudian menjadikan 40 byte pertama sebagai header, lalu byte ke 41 sampai byte ke 43 sebagai panjang *file carrier audio*. Kemudian dilanjutkan dengan menyalin seluruh file *carrier audio* dimulai dari byte ke 44 untuk dijadikan media penyimpan pesan rahasia.

```
msg=get(handles.edit1,'string');    %ambil pesan rahasia dari editbox
    if length(msg) > 100
        msgbox('Message too big, select small message');
    else
        [m_msg,n_msg]=size(msg);
        msg_double=double(msg);             %konversi ke double
        msg_bin=de2bi(msg_double,8);        %konversi ke binary
        [m,n]=size(msg_bin);                %ukuran pesan dalam binary
        msg_bin_re=reshape(msg_bin,m*n,1);  %reshape the message binary in a
column vector
        m_bin=de2bi(m_msg,1000)';
        n_bin=de2bi(n_msg,1000)';
        len=length(msg_bin_re);       %panjang pesan biner
        len_bin=de2bi(len,2000)';       %konversi panjang pesan ke biner
        %penyisipan identitas ke dalam 8 sampel pertama dari file wav
        identity=[1 0 1 0 1 0 1 0]';
        dta(1:8)=bitset(dta(1:8),lsb,identity(1:8));
        %penyisipan penjang pesan mulai dari sampel ke 9 sampai ke 28
        dta(9:18)=bitset(dta(9:18),lsb,m_bin(1:10));
        dta(19:28)=bitset(dta(19:28),lsb,n_bin(1:10));
        %penyisipan pesan rahasia mulai dari sampel ke 29
        dta(29:28+len)=bitset(dta(29:28+len),lsb,msg_bin(1:len)');
```

Gambar 4.19 *Source code* tombol *execute*

Setelah proses penyalinan selesai dan file audio ditutup kembali, terdapat perintah untuk mengambil pesan rahasia yang terdapat pada edit1 lagi dan kemudian terdapat percabangan sebagaimana yang terlihat pada Gambar 4.19. Terdapat percabangan dengan dua kondisi : jika pesan rahasia yang dimasukkan lebih dari 100 karakter, maka aplikasi akan memunculkan *pop up message* berupa peringatan bahwa pesan yang dimasukkan terlalu besar. Namun jika pesan yang dimasukkan kurang dari atau sama dengan 100, maka proses *embedding* akan dilanjutkan ke langkah selanjutnya. Kemudian pesan rahasia akan dikonversi menjadi data biner setelah sebelumnya dirubah menjadi tipe data *double*. Kemudian panjang dari pesan dikonversi menjadi data biner. Lalu dilanjutkan dengan penyisipan identitas (*stego key*) ke dalam 8 sampel pertama dari *file* audio. Kemudian panjang pesan yang telah dikonversi menjadi biner disisipkan ke dalam sampel ke 9 sampai sampel ke 28 dari *file* audio WAV. Lalu isi pesan rahasia disembunyikan ke dalam *file* WAV dimulai dari sampel ke 29 sampai akhir.

```
        %buka file wav yang baru dalam write mode
        fid2 = fopen('new.wav','w');
        %salin header file wav yang asli
        fwrite(fid2,header,'uint8');
        fwrite(fid2,data_size,'uint32');
        %salin data sampel wav denganpesan rahasia ke dalam wav yang baru
        fwrite(fid2,dta,'uint16');
        fclose(fid2);
        [y,fs]=audioread('new.wav');
      %penyimpanan file stego audio dengan nama sesuai keinginan pengguna
        [nfname,npath]=uiputfile('.wav','Save sound','new_sound.wav');
    if isequal(nfname,0) || isequal(npath,0)
        return
    else
        nwavfile=fullfile(npath, nfname);
        audiowrite(nwavfile,y,fs);
    end
        msgbox('Message embedded');
        set(hObject,'enable','off');
        guidata(hObject, handles);
    end
end
```

Gambar 4.20 *Source code* tombol *execute*

Kemudian proses dilanjutkan dengan membuka *file* WAV baru yang akan dijadikan *stego file* yang memiliki pesan rahasia di dalamnya. lalu header *file* WAV yang asli (*carrier audio*) disalin ke dalam *file* WAV yang baru (*stego audio*), dilanjutkan dengan menyalin seluruh *file carrier audio* yang sudah mengandung pesan rahasia ke dalam *file stego audio.* Setelah proses penyalinan selesai, terdapat kode uiputfile yang berfungsi untuk menyimpan audio dengan nama dan folder yang diinginkan. sebagaimana yang tertera pada Gambar 4.20.

4.1.3.2 Proses *extracting*

```
function pushbtnextract_Callback(hObject, eventdata, handles)
% hObject    handle to pushbtnextract (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% pilih file yang akan diekstrak
[handles.filename, handles.pathname] = uigetfile('*.wav','Select a file');
    % proses menampilkan periodogram
     axes(handles.axes2);
    [signal1, fs]=audioread(handles.filename);
    signal1 = signal1(:,1);
    plot(psd(spectrum.periodogram,signal1,'Fs',fs,'NFFT',length(signal1)));
guidata(hObject, handles);
```

Gambar 4.21 *Source code* tombol *select audio*

Gambar 4.21 merupakan *source code* untuk tombol *select audio* pada proses *extracting* yang diberi *tag* pushbtnextract. Pada baris keenam terdapat perintah uigetfile yang memungkinkan pengguna untuk memilih *file* audio dari folder yang sesuai dengan *path* matlab. Kemudian pada baris kedelapan sampai akhir adalah kode untuk proses menampilkan periodogram menggunakan properti axes2.

```
1. function pushbtnexecute2_Callback(hObject, eventdata, handles)
2. % hObject    handle to pushbtnexecute2 (see GCBO)
3. % eventdata  reserved - to be defined in a future version of MATLAB
4. % handles    structure with handles and user data (see GUIDATA)
5. %buka file stego audio
6. fid1=fopen([handles.pathname handles.filename],'r');
7. header=fread(fid1,40,'uint8=>char');
8. data_size=fread(fid1,1,'uint32');
9. %baca data sampel wav
10. [dta,count]=fread(fid1,inf,'uint16');
11. %tutup file wav
12. fclose(fid1);
13. lsb=1;
```

Gambar 4.22 *Source code* tombol *execute*

Gambar 4.22 merupakan *source code* dari tombol *execute* yang terdapat pada proses *extracting* yang diberi *tag* pushbtnexecute2. Pada baris kelima terdapat kode fopen dan fread yang berfungsi untuk membuka dan membaca *file stego audio*.

```
1.      if identity==[1 0 1 0 1 0 1 0]
2.          %ekstraksi panjang pesan dari sampel ke 9 sampai ke 28
3.          len_bin=zeros(20,1);
4.          m_bin=zeros(10,1);
5.          n_bin=zeros(10,1);
6.          m_bin(1:10)=bitget(dta(9:18),lsb);
7.          n_bin(1:10)=bitget(dta(19:28),lsb);
8.          %konversi penjang pesan ke desimal
9.          %len=bi2de((len_bin)');
10.         m=bi2de(m_bin');
11.         n=bi2de(n_bin');
12.         len=m*n*8;
13.         secmsg_bin=zeros(len,1);
14.         %ekstrak pesan rahasia dari stego audio
15.         secmsg_bin(1:len)=bitget(dta(29:28+len),lsb);
16.         secmsg_bin_re=reshape(secmsg_bin,len/8,8);
17.         secmsg_double=bi2de(secmsg_bin_re); %konversi ke desimal
18.         secmsg=char(reshape(secmsg_double,m,n));  %konversi ke char
19.         size(secmsg)
20.         secmsg=reshape(secmsg,m,n/8);
21.         set(handles.edit2,'string',secmsg);
22.     else
23.         msgbox('File has no hidden text','Empty');
24. end
```

Gambar 4.23 *Source code* tombol *execute*

Setelah file *stego audio* selesai dibaca dan ditutup kembali dengan fclose, proses dilanjutkan dengan percabangan dua kondisi sebagaimana yang tertera pada Gambar 4.23 : jika identitas 10101010 (sama dengan yang dimasukkan saat proses *embedding*), maka proses *extracting* akan dilanjutkan dengan mengekstrak panjang pesan rahasia dari sampel ke 9 sampai sampel ke 28. Kemudian panjang pesan tersebut dikonversi ke desimal. Setelah proses tersebut selesai, dilanjutkan dengan ekstraksi pesan rahasia dari *stego audio*, lalu dilanjutkan dengan konversi pesan rahasia ke dalam desimal, kemudian dikonversi lagi menjadi tipe data

char untuk kemudian ditampilkan dengan properti edit2. Namun jika identitas tidak sama dengan yang dilakukan pada saat proses *embedding*, maka aplikasi akan mengeluarkan *pop up message* berupa peringatan bahwa *file* audio tersebut tidak memiliki pesan rahasia.

4.1.3.3 Proses hitung PSNR

```
1. function pushbtnpsnr_Callback(hObject, eventdata, handles)
2. % hObject    handle to pushbtnpsnr (see GCBO)
3. % eventdata  reserved - to be defined in a future version of MATLAB
4. % handles    structure with handles and user data (see GUIDATA)
5. [handles.fname, handles.pname] = uigetfile('*.wav','select a file without
   hidden message in it');
6. [signal1, fs] =audioread(handles.fname);
7. [handles.filename, handles.pathname] = uigetfile('*.wav','Select a file with
   hidden message in it');
8. [signal2, fs] =audioread(handles.filename);
9. axes(handles.axes4);
10. signal1 = signal1(:,1);
11. signal2 = signal2(:,1);
12. [Pxx1,Fxx] = periodogram(signal2,[],length(signal2),fs);
13. [Pxx2,Fxx] = periodogram(signal1,[],length(signal1),fs);
14. plot(Fxx,10*log10(Pxx1)); hold on;
15. plot(Fxx,10*log10(Pxx2),'r'); hold on;
16. legend('Stego Audio','Original Audio','Location','NorthEast');
17. xlabel('Hz'); ylabel('dB/Hz');
18. MSE = sum(sum((signal1-signal2).^2));
19. PSNR = 10*log10(255/MSE);
20. set(handles.textpsnr,'string',[' PSNR=' num2str(PSNR)]);
```

Gambar 4.24 *Source code* tombol *select audio*

Gambar 4.24 merupakan *source code* dari tombol *select audio* yang terdapat pada *tab* PSNR tang diberi tag pushbtnpsnr. Pada baris ke 6 dan ke 9 terdapat perintah uigetfile yang berfungsi untuk memungkinkan pengguna untuk memilih *file carrier audio* dan *stego audio* yang akan dihitung nilai PSNR nya. Kemudian dilanjutkan dengan proses plotting dua periodogram ke dalam sati *axes.* Lalu proses dilanjutkan dengan perhitungan untuk mencari nilai MSE. Setelah nilai MSE didapatkan, kemudian nilai MSE tersebut digunakan untuk mencari nilai PSNR. Kemudian nilai PSNR yang berupa *number* dikonversi menjadi *string* untuk kemudian ditampilkan pada properti textpsnr.

4.2 Uji Kinerja Aplikasi

4.2.1 Uji proses *embedding* dan *extracting*

4.2.1.1 Uji proses *embedding*

Tabel 4.1 Hasil uji proses *embedding*

No	*Carrier audio*	Ukuran *Secret message*	Proses *embedding*	*Stego audio*
1	Al-Fatihah.wav	100 *bytes*	sukses	Al-Fatihah_Stego.wav
2	Bung Tomo.wav	100 *bytes*	sukses	Bung Tomo_Stego.wav
3	In the stone.wav	100 *bytes*	sukses	In the stone_Stego.wav
4	Nocturne.wav	100 *bytes*	sukses	Nocturne_Stego.wav
5	Sweet child o mine.wav	100 *bytes*	sukses	Sweet child o mine_Stego.wav

Tabel 4.1 merupakan tabel yang menunjukkan hasil dari proses *embedding* yang telah dilakukan terhadap *file carrier audio* yang tertera pada Tabel 3.1. Adapun pesan rahasia yang disisipkan ke dalam masing-masing *file* audio adalah teks berukuran 100 *byte* yang berisi 100 karakter angka tanpa spasi. Dari Tabel 4.1 di atas dapat dilihat bahwa proses *embedding* berhasil dilakukan terhadap semua *file carrier audio* yang telah ditentukan. Selain itu, *file* audio yang telah melewati proses *embedding* juga diberi nama pengganti untuk menandakan bahwa *file* tersebut adalah *file stego audio.*

4.2.1.2 Uji proses *extracting*

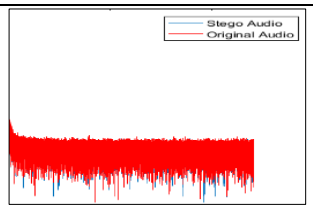
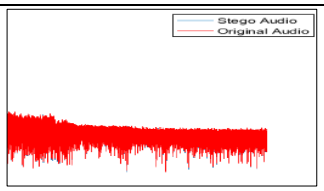
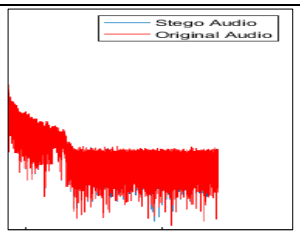
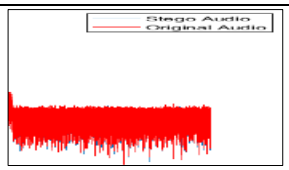
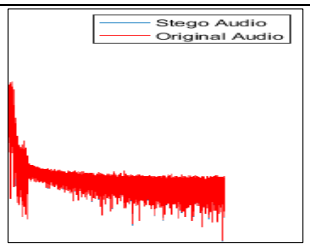
Tabel 4.2 Hasil uji proses *extracting*

No	*Stego audio*	Proses *embedding*	*Hidden message*
1	Al-Fatihah_Stego.wav	sukses	Teks 100 *bytes*
2	Bung Tomo_Stego.wav	sukses	Teks 100 *bytes*
3	In the stone_Stego.wav	sukses	Teks 100 *bytes*
4	Nocturne_Stego.wav	sukses	Teks 100 *bytes*
5	Sweet child o mine_Stego.wav	sukses	Teks 100 *bytes*

Tabel 4.2 merupakan tabel yang menunjukkan hasil pengujian proses *extracting.* Dapat dilihat bahwa proses *extracting* berhasil dilakukan terhadap semua *stego audio* hasil dari proses *embedding* yang telah dilakukan sebelumnya. pesan rahasia yang berhasil diekstrak juga merupakan pesan yang sama dengan yang sebelumnya disisipkan pada proses *embedding.*

4.2.2 Uji perbandingan *carrier audio* dengan *stego audio*

Tabel 4.3 Perbandingan *carrier audio* dengan *stego audio*

No	*Carrier audio*	Ukuran *carrier audio* (*byte*)	*Stego audio*	Ukuran *stego audio* (*byte*)	Perbandingan *Periodogram*
1	Al-Fatihah .wav	1.306.186	Al-Fatihah_Stego .wav	1.306.084	
2	Bung Tomo .wav	1.306.186	Bung Tomo_Stego .wav	1.306.084	
3	In the stone .wav	1.306.186	In the stone_Stego .wav	1.306.084	
4	Nocturne. wav	1.306.186	Nocturne _Stego .wav	1.306.084	
5	Sweet child o mine .wav	1.306.186	Sweet child o mine_Stego .wav	1.306.084	

Tabel 4.3 merupakan tabel yang berisi perbandingan antara *carrier audio* dengan *stego audio.* Pembanding yang digunakan adalah nama *file*, ukuran, dan tampilan periodogram. Berdasarkan Tabel 4.3 di atas, terdapat perubahan ukuran setelah proses *embedding* dilakukan. Dapat dilihat bahwa *stego audio* memiliki pengurangan 102 *bytes* dari *carrier audio.* Perubahan yang terjadi terhadap setiap *file* relatif sama dikarenakan *carrier audio* dan pesan rahasia yang disisipkan memiliki ukuran yang sama. Perubahan yang ditunjukkan oleh

periodogram pun tidak begitu terlihat sehingga dapat dipastikan bahwa proses *embedding* berhasil dan *stego file* yang dihasilkan aman dan tidak menimbulkan kecurigaan.

4.2.3 Uji PSNR

Tabel 4.4 Hasil uji PSNR

No	*Carrier Audio*	*Stego Audio*	PSNR (dB)
1	Al-Fatihah.wav	Al-Fatihah_Stego.wav	88,173
2	Bung Tomo.wav	Bung Tomo_Stego.wav	87,920
3	In the stone.wav	In the stone_Stego.wav	87,871
4	Nocturne.wav	Nocturne_Stego.wav	88,183
5	Sweet child o mine.wav	Sweet child o mine_Stego.wav	87,969

Tabel 4.4 merupakan tabel hasil pengujian PSNR terhadap *file carrier audio* dan *stego audio*. Uji PSNR menggunakan rumus dan perhitungan matematis untuk mengetahui nilai perbandingan antara kedua file yang diujikan. Seperti yang sudah dibahas sebelumnya, nilai PSNR dikatakan baik jika memiliki nilai di atas 30 dB. Dari Tabel 4.4 dapat ditarik kesimpulan bahwa kelima sampel audio yang diujikan memiliki kualitas yang baik dengan nilai PSNR rata-rata sebesar 88,023 dB.

4.3 Analisis Hasil Pengujian

Dari semua pengujian di atas dapat ditarik kesimpulan bahwa aplikasi audio steganografi dengan algoritma LSB berhasil dibangun dan diuji. Perangkat lunak ini memiliki tiga fungsi utama yaitu melakukan proses *embedding*, melakukan proses *extracting*, dan melakukan perhitungan PSNR. Masing-masing fungsi memiliki fitur yang dipicu oleh tombol-tombol dan semuanya dapat dieksekusi dengan baik. Program ini juga dapat menampilkan periodogram sebagai visualisasi sinyal audio.

Pengujian proses *embedding* telah dilakukan dan berjalan dengan baik pada setiap sampel sebagaimana yang ditampilkan pada Tabel 4.1. Proses *embedding* merupakan proses mpenyisipan pesan rahasia ke dalam *carrier audio.* Proses ini dilakukan terhadap 5 sampel *file* audio sebagai *carrier audio* dengan pesan rahasia berupa teks sebesar 100 *bytes*. dapat dilihat bahwa proses *embedding* berhasil dilakukan terhadap kelima sampel tersebut dan menghasilkan *stego file* dengan nama yang baru sesuai dengan keinginan pengguna.

Pengujian proses *extracting* juga telah dilakukan dan berjalan dengan baik sebagaimana yang ditampilkan pada Tabel 4.2. Proses *extracting* merupakan proses ekstraksi pesan rahasia dari *stego audio.* Proses ini dilakukan terhadap *stego audio* yang sebelumnya telah melewati proses *embedding*. Pada Tabel 4.2 dapat dilihat juga bahwa proses *extracting* berhasil dilakukan terhadap semua *stego audio* dan menghasilkan pesan tersembunyi yang sama dengan yang disisipkan pada proses *embedding.*

Pada Tabel 4.3 ditampilkan perbandingan antara *carrier audio* dengan *stego audio.* Dari tabel tersebut dapat dilihat bahwa nama *file carrier audio* berubah setelah dilakukan proses *embedding* menjadi *stego audio.* Dari Tabel 4.3 dapat dilihat juga perbedaan ukuran antara *carrier audio* dan *stego audio*. Perubahan yang terjadi pada setiap sampel relatif sama karena *file carrier audio* dan pesan rahasia yang disisipkan memiliki ukuran yang sama. Sedangkan perubahan yang terjadi pada periodogram tidak begitu terlihat secara kasat mata.

Hasil pengujian PSNR yang telah dilakukan pada Tabel 4.4 menunjukkan bahwa terdapat perubahan yang terjadi antara *carrier audio* dengan *stego audio*. Nilai PSNR yang ideal adalah di atas 30 dB. Makin tinggi nilai PSNR berarti makin banyak kemiripan yang dimiliki oleh kedua file. Sampel nomer 1 yang merupakan potongan dari surat Al-Fatihah menunjukkan nilai PSNR 88,173 dB yang mana nilai tersebut termasuk tinggi yang berarti kedua *file* menunjukkan kemiripan yang banyak. Sampel nomer 2 yang merupakan potongan dari pidato Bung Tomo mendapatkan nilai PSNR 87,920 dB yang mana nilai tersebut relatif besar. Sampel nomer 3 yang merupakan potongan sebuah lagu berjudul In the stone mendapatkan nilai PSNR sebesar 87,871 dB. Sampel nomer 4 yang merupakan potongan dari sebuah musik klasik berjudul Nocturne mendapatkan nilai PSNR sebesar 88,183 dB. Dan sampel nomer 5 yang merupakan potongan lagu Sweet child o mine mendapatkan nilai PSNR sebesar 87,969 dB.

Dari penjelasan mengenai Tabel 4.4 di atas, dapat dilihat bahwa yang memiliki nilai PSNR terbesar adalah sampel nomer 4 merupakan potongan dari lagu Nocturne dengan nilai PSNR 88,183 dB. Sementara sampel dengan nilai terendah adalah sampel nomer 3 yang merupakan potongan dari lagu In the stone dengan nilai PSNR sebesar 87,871 dB.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan sebagai berikut :

a. Implementasi teknik steganografi terhadap audio wav dengan algoritma LSB telah berhasil dilakukan.
b. Perbedaan yang terdapat pada *file* audio setelah dilakukan teknik steganografi adalah perubahan nama (opsional), perubahan ukuran, dan perubahan kualitas *file* audio yang dapat dihitung secara matematis dengan menggunakan rumus PSNR.
c. Dari kelima sampel suara yang dijadikan bahan uji, sampel suara terbaik untuk dijadikan media audio steganografi adalah musik nocturne karena memiliki nilai PSNR tertinggi sebesar 88,183 dB. Sementara sampel suara terburuk untuk dijadikan media audio steganografi adalah musik In the stone karena memiliki nilai PSNR terendah sebesar 87,871 dB.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka saran untuk penelitian selanjutnya mengenai audio steganografi dengan algoritma LSB adalah :

a. Membuat pembatasan panjang pesan rahasia berdasarkan jumlah *byte* homogen yang terdapat pada *file carrier* audio yang dipilih.
b. Meringkas panjang barisan kode sehingga aplikasi mampu melakukan proses eksekusi dengan lebih cepat.

DAFTAR PUSTAKA

Abidia Away, G. (2006). *Matlab*. Penerbit Informatika.

Bhalshankar, S., & Gulve, A. K. (2015). Audio Steganography: LSB Technique Using a Pyramid Structure and Range of Bytes, (August). Retrieved from http://arxiv.org/abs/1509.02630

Craig. (2008). WAVE PCM soundfile format. Retrieved from http://soundfile.sapp.org/doc/WaveFormat/

Dendy. (2014). Perbedaan Data Analog dan Data Digital. Retrieved from http://rendheng.blogspot.co.id/2013/04/model-transmisi-data.html

E-jurnal. (2016). Pengertian Suara dan Jenisnya. Retrieved from http://www.e-jurnal.com/2013/12/pengertian-suara-dan-jenis-jenisnya.html

Firdaus, V. (2009). Studi Dan Implementasi Steganografi Pada File Audio Dengan Teknik Spread Spectrum. *Media*, 1–4.

Haris, I. (2010). Format File Audio. Retrieved from https://blog.ub.ac.id/indracypid/2010/04/17/format-file-audio/

Imam, A. (2015). Pengertian dan Perbedaan Data dan Informasi. Retrieved from http://www.kuliah.info/2015/05/pengertian-dan-perbedaan-data-dan.html

Julianto, Y., & Bendi, K. J. (2016). Sistem Steganografi dengan Metode Least Significant Bit (LSB) Terlacak. *FLASH*, *2*(2), 116–123.

MARYAM. (2013). Audio Steganografi Dengan Metode Least Significant Bit Terhadap Pesan Terenkripsi dengan Algoritma Twofish.

Paulus, E., & Nataliani, Y. (2007). *Cepat Mahir GUI Matlab*. *Penerbit Andi*.

Purba, J. V., Situmorang, M., & Arisandi, D. (2012). Implementasi Steganografi Pesan Text Ke Dalam File Sound (.Wav) Dengan Modifikasi Jarak Byte Pada Algoritma Least Significant Bit (LSB). *Jurnal Dunia Teknoogi Informasi Vol. 1, No. 1, (2012) 50-55*, *1*(1), 50–55.

Salikin, R. (2008). Perbandingan Format WAV dengan Format MP3. Retrieved from http://xtalplanet.blogspot.com/2008/06/perbandingan-format-wav-dengan-format.html

Saragih, R. A. (2006). Metode Parity Coding Versus Metode Spread Spectrum Pada Audio Steganography. *Spectrum*, *2006*(Snati), 71–76. Retrieved from http://jurnal.uii.ac.id/index.php/Snati/article/view/1541

Sayogo, G. (2015). Data (Pengertian, Jenis, Metode Pengumpulan) dan Variabel Penelitian.

Retrieved from https://ganjarsayogo.wordpress.com/2015/04/24/data-pengertian-jenis-metode-pengumpulan-dan-variabel-penelitian/

Srikandi, R. (2015). Pengertian dan Metode Steganografi. Retrieved from http://seputarpengertian.blogspot.co.id/2015/08/pengertian-dan-metode-steganografi.html

Tanty. (2010). Tugas 4 - Data Analog dan Data Digital. Retrieved from http://t4eeenty.blogspot.co.id/2010/04/tugas-4-data-analog-dan-data-digital.html

Wikipedia. (2008). Data. Retrieved from https://id.wikipedia.org/wiki/Data#cite_ref-Vardiansyah,_Dani_2008_1-2

Wikipedia. (2018). Steganografi. Retrieved from https://id.wikipedia.org/wiki/Steganografi