TEXT STEGANOGRAPHY PADA MEDIA *AUDIO* FORMAT M4A DENGAN MENGGUNAKAN METODE *LEAST SIGNIFICANT BIT*

SKRIPSI

AINUL QOLBY ALBANTANY
0701163077

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021

TEXT STEGANOGRAPHY PADA MEDIA *AUDIO* FORMAT M4A DENGAN MENGGUNAKAN METODE *LEAST SIGNIFICANT BIT*

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

AINUL QOLBY ALBANTANY
0701163077

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan

Assalamu'alaikum Wr. Wb

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudari,

Nama	: Ainul Qolby Albantany
Nomor Induk Mahasiswa	: 0701163077
Program Studi	: Ilmu komputer
Judul	: *Text Steganography* Pada Media *Audio* Format M4A Dengan Menggunakan Metode *Least Significant Bit*

Dapat disetujui untuk segera di*Munaqasyah*kan. Atas perhatiannya kami ucapkan terimakasih.

Medan, 1 Maret 2021
17 Rajab 1442 H

Komisi Pembimbing,

Pembimbing I,	Pembimbing II,
Dr. Mhd Furqan, S,Si., M.Comp.Sc	Yusuf Ramadhan Nasution, M.Kom
NIP. 198008062006041003	NIB. 1100000075

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama	: Ainul Qolby Albantany
Nomor Induk Mahasiswa	: 0701163077
Program Studi	: Ilmu Komputer
Judul	: *Text Steganography* Pada Media *Audio* Format M4A Dengan MenggunakanMetode *Least Significant Bit*

Dengan ini menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akdemik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 1 Maret 2021

Ainul Qolby Albantany
NIM. 0701163077

KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: http://saintek.uinsu.ac.id, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.149/ST/ST.V.2/PP.01.1/08/2021

Judul	: *Text Steganography* Pada Media *Audio* Format M4A Dengan Menggunakan Metode *Least Significant Bit*
Nama	: Ainul Qolby Albantany
Nomor Induk Mahasiswa	: 0701163077
Program Studi	: Ilmu Komputer
Fakultas	: Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal	: Selasa, 23 Maret 2021
Tempat	: Via Zoom Meeting

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, M.Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,	Penguji II,
Dr. Mhd. Furqan, S.Si., M.Comp.Sc	Yusuf Ramadhan Nasution, M.Kom
NIP. 198008062006041003	NIB. 1100000075
Penguji III,	Penguji IV,
Heri Santoso, M.Kom	Sriani, M.Kom
NIB. 55201005	NIB. 1100000108

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd.Syahnan,M.A
NIP: 196609051991031002

ABSTRAK

Mengirim informasi menjadi semakin rentan terhadap penyadapan, yang dapat mengubah otentikasi dan integritas pesan. Seringkali seseorang ingin mengirim pesan yang sangat sensitif atau pribadi. Oleh karena itu, diperlukan sistem keamanan data untuk melindungi pesan pribadi dan sensitif agar dapat dijangkau oleh orang yang berhak menerimanya. Salah satunya dengan menggunakan teknik yang disebut steganografi. Steganografi adalah teknik untuk mengamankan data pribadi atau rahasia. Salah satu metode steganografi digital adalah *Least Significant* Bit yang bekerja dengan memodifikasi bit terakhir dalam *byte file* audio untuk menyembunyikan urutan *byte* yang berisi data rahasia. Skripsi ini membahas tentang perancangan aplikasi steganografi berbasis desktop menggunakan metode *Least Significant* Bit dilanjutkan dengan pembuatan kunci yang menyembunyikan *file* teks (*.txt) dalam *file* audio (*.m4a) yang diimplementasikan menggunakan perangkat lunak *Microsoft Visual Studio* 2012. Proses *embedding* dengan metode LSB menghasilkan steganografi audio yang tidak banyak berpengaruh pada kualitas suara audio, hal ini tentunya membuat data *file* teks aman pada *file* audio m4a karena dalam suara tidak ada yang mencurigakan dari steganografi audio m4a, dan hasil pada proses pengambilan pesan berupa teks berhasil dilakukan tanpa merubah kapasitas *file* sebelum dan sesudah disisipkan. Oleh karena itu, kecil kemungkinannya kerahasiaan *file* teks yang akan dikirimkan ke penerima dengan objek audio (*.m4a) mengalami kebocoran.

Kata Kunci : Steganografi, *Least Significant* Bit, Audio, M4A

ABSTRACT

Sending information is becoming increasingly vulnerable to eavesdropping, which can compromise message authentication and integrity. Often a person wants to send a very sensitive or private message. Therefore, a data security system is needed that can protect private and sensitive messages so that they can be reached by those who are entitled to receive them. One of them by using a technique called steganography. Steganography is a technique for securing private or confidential data. One method of digital steganography is LSB which works by modifying the last bit in the byte of the audio file to hide the sequence of bytes that contain secret data. This thesis discusses the design of a desktop-based steganography application using the LSB method followed by the creation of a key that hides a text file (*.txt) in an audio file (*.m4a) which is implemented using Microsoft Visual Studio 2012 software. The embedding process with the Least Significant Bit method produces audio steganography which does not have much effect on the audio sound quality, this of course makes the text file data safe in the m4a audio file because in the sound there is nothing suspicious from the m4a audio steganography, and the results in the message retrieval process are in the form of the text was successfully executed without changing the file capacity before and after it was inserted. Therefore, it is unlikely that the confidentiality of a text file that will be sent to a recipient with an audio object (*.m4a) is leaked.

Keywords: Steganography, Least Significant Bit, Audio, M4A

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillahirabbil'alamin, Puji syukur penulis ucapkan kepada Allah SWT atas yang telah memberikan rahmat dan karunia-Nya. Tak lupa juga sholawat dan salam kepada Nabi kita Muhammad SAW. Sehingga penulis dapat menyelesaikan penulisan proposal skripsi ini yang berjudul **"*Text Steganography* pada Media Audio Format M4A dengan Menggunakan Metode Least significant bit"** disusun sebagai salah satu syarat untuk memperoleh gelar sarjana komputer pada Jurusan Ilmu Komputer Universitas Islam Negeri Sumatera Utara.

Penulis menyadari bahwa penyelesaian skripsi ini tidak lepas dari dukungan, bantuan, bimbingan dan saran semua pihak dalam proses penyusunanya. Pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibunda tercinta, ibu Arviani yang telah memberikan yang banyak memberikan bantuan rohani dan material, semangat dan do'a kepada penulis.
2. Bapak Prof. Dr. Syahrin Harahap, MA, selaku Rektor Universitas Islam Negeri Sumatera Utara.
3. Bapak Dr. Mhd. Syahnan, M.A, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
4. Bapak Ilka Zufria, M.Kom selaku Ketua Jurusan Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
5. Bapak Dr. Mhd Furqan, S,Si.,M.Comp.Sc selaku dosen pembimbing skripsi I yang telah berkontribusi memberikan masukan, saran, kritik dan arahan kepada penulis selama pengerjaan skripsi ini.
6. Bapak Yusuf Ramadhan Nasution, M.Kom selaku dosen pembimbing skripsi II yang mana telah berkontribusi dalam membantu penulis seperti memberikan masukan, saran, kritik dan arahan kepada penulis selama pengerjaan skripsi ini.
7. Bapak Rakhmat Kurniawan, M.Kom selaku Sekertaris Jurusan Ilmu Komputer dan selaku dosen pembimbing akademik.

8. Seluruh tenaga pengajar dan pegawai program studi S1 Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
9. Teman-teman kelas Ilmu komputer 3 yang selalu memberikan dukungan serta arahan kepada penulis.
10. Adik kandung penulis, Maulana Luthfi Albantany terima kasih atas dukungan, doa dan semangatnya yang selalu diberikan untuk penulis.
11. Dan semua pihak yang telah membantu penulis namun tidak dapat disebutkan satu persatu.

Penulis sangat menyadari bahwa dalam proses penulisan skripsi ini masih jauh dari sempurna. Oleh karena itu, saya berharap para pembaca dapat memberikan kritik dan saran yang membangun. Semoga proposal skripsi ini dapat bermanfaat bagi penulis dan pembaca. Aamin Ya Rabbal'alamin.

Medan, 23 Maret 2021

Hormat saya,

Ainul Qolby Albantany

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

Lampiran **Judul Lampiran**

BAB I
PENDAHULUAN

1.1. Latar Belakang

Keamanan data online melalui komputer lokal dan layanan Internet merupakan peran penting. Hal ini disebabkan semakin banyaknya kejahatan teknis yang menggunakan berbagai teknologi. Tanpa teknologi keamanan khusus, orang-orang yang tidak bertanggung jawab dapat dengan mudah mendapatkan data tersebut. Seperti yang dijelaskan dalam Q.S Al Hahr:23

هُوَ ٱللَّهُ ٱلَّذِى لَآ إِلَٰهَ إِلَّا هُوَ ٱلْمَلِكُ ٱلْقُدُّوسُ ٱلسَّلَمُ ٱلْمُؤْمِنُ ٱلْمُهَيْمِنُ ٱلْعَزِيزُ ٱلْجَبَّارُ ٱلْمُتَكَبِّرُ ۚ سُبْحَٰنَ ٱللَّهِ عَمَّا يُشْرِكُونَ ٢٣

Artinya : Dialah Allah Yang tiada Tuhan selain Dia, Raja, Yang Maha Suci, Yang Maha Sejahtera, Yang Mengaruniakan Keamanan, Yang Maha Memelihara, Yang Maha Perkasa, Yang Maha Kuasa, Yang Memiliki segala Keagungan, Maha Suci Allah dari apa yang mereka persekutukan.

Pengiriman informasi semakin rentan terhadap penyadapan yang dapat mengubah autentifikasi dan integritas pesan. Seringkali ketika seseorang ingin mengirim pesan, mereka tidak ingin isi pesan tersebut diketahui oleh orang lain. Beberapa pesan hanya dimaksudkan untuk diketahui oleh pengirim dan penerima. Oleh karena itu, yang dibutuhkan adalah sistem keamanan data yang dapat melindungi pesan yang bersifat pribadi dan sensitif sehingga dapat dijangkau oleh pihak yang berwenang untuk menerimanya. (Benny Kuniady, 2017).

Salah satunya adalah dengan menggunakan teknik steganografi. Steganografi mirip dengan kriptografi tetapi kedua disiplin tersebut memiliki beberapa perbedaan. Steganografi bertujuan untuk menyembunyikan atau menyembunyikan pesan rahasia melalui suatu media. Sedangkan kriptografi adalah suatu cara untuk menyamarkan suatu pesan rahasia yang tidak dapat dilihat, tetapi tidak menyembunyikannya. Steganografi biasanya diimplementasikan melalui

media digital, dimana pesan yang disematkan atau cover object dapat berupa teks, gambar, audio, atau video.Tingkat keamanan penyembunyian informasi adalah aspek terpenting dari steganografi, yang mengacu pada seberapa besar ketidakmampuan pihak ketiga dalam mendeteksi keberadaan informasi yang tersembunyi (Ulan, 2017)

Salah satu metode steganografi adalah LSB (*Least Significant Bit*), Metode LSB adalah salah satu metode yang paling populer pada audio steganografi. Cara kerja metode ini adalah dengan 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. Tetapi penggunaan citra sebagai media penyisipan pesan sudah banyak dianalisa dan dikembangkan, sedangkan penggunaan media arsip audio relatif jarang dan juga media audio dengan format (*.m4a) yang belum ada. Banyak metode yang ditemukan untuk melakukan audio steganografi, salah satunya dengan mengunakan algoritma LSB (*Least Significant Bit).*

Berdasarkan hal tersebut penulis bermaksud untuk merancang dan menganalisa sistem aplikasi keamanan data dengan cara Steganografi yaitu menyembunyikan *file* teks (*.txt) pada *file* audio (*.m4a) dengan menggunakan metode LSB (*Least Significant Bit*) yang diterapkan menggunakan *software Microsoft Visual Studio*. Penulis akan membuat sebuah rancang bangun aplikasi steganografi yang dapat melakukan penyembunyian atau penyisipan dan tentu pengambilan kembali suatu informasi. Sehingga penulis mengangkat judul Tugas Akhir **"*Text Steganography* Pada Media Audio Format M4a Dengan Menggunakan Metode *Least Significant Bit*".** Aplikasi ini diharapkan dapat meminimalisir terjadinya pencurian informasi digital oleh pihak yang tidak bertanggung jawab.

1.2. Rumusan Masalah

Permasalahan yang dapat dikaji lebih lanjut dari latar belakang yang ada masalah :

1. Bagaimana menganalisis penyisipan dan pengekstrakan dengan metode LSB (*Least Significant Bit*) terhadap pesan rahasia ?
2. Bagaimana menghasilkan aplikasi berbasis desktop yang mampu menyisipkan pesan rahasia yaitu berupa teks (*.txt) menggunakan metode LSB ke dalam media penampung audio (*.m4a) ?
3. Bagaimana perubahan yang terjadi terhadap *file* audio (*.m4a) setelah dilakukan proses steganografi ?

1.3. Batasan Masalah

1. Metode steganografi yang digunakan dalam penelitian ini adalah LSB.
2. *Input* pesan rahasia berupa *file* format (*.txt).
3. Media penampung berupa audio format (*.m4a).
4. Media steganografi yang digunakan adalah 3 *file* audio berformat (*.m4a).
5. Software yang digunakan adalah *Microsoft Visual Studio* 2012 dan bahasa pemograman yang digunakan adalah C#.
6. Penelitian berfokus pada proses penyembunyian pesan rahasia dan ekstraksi pesan rahasia.
7. Penilaian kualitas suara pada media penampung audio m4a menggunakan *software* Spek

1.4. Tujuan Penelitian

1. Untuk mengetahui proses penyisipan dan pengekstrakan dengan menggunakan metode LSB terhadap pesan rahasia.
2. Untuk merancang suatu aplikasi berbasis desktop yang mampu menyisipkan pesan rahasia yaitu berupa teks (*.txt) menggunakan metode LSB ke dalam media penampung audio (*.m4a).
3. Untuk mengetahui perubahan yang terjadi terhadap *file* audio setelah dilakukan proses steganografi.

1.5. Manfaat Penelitian

1. Dapat menyembunyikan pesan teks format (*.txt) ke dalam media audio format (*.m4a) menggunakan metode *least significant bit*.
2. Dapat mengetahui apa saja perubahan yang terjadi terhadap *file* audio setelah dilakukan proses steganografi.
3. Dapat meminimalisir kasus bocornya data dan informasi yang bersifat rahasia.
4. Menjadi acuan untuk penelitian-penelitian selanjutnya tentang audio steganografi dengan algoritma LSB.

BAB II
KAJIAN PUSTAKA

2.1. Keamanan dan Kerahasiaan Data

Ketika data berharga, keamanan dan kerahasiaan data sangat penting. Misalnya, mengapa kita perlu dilindungi secara pribadi sebagai warga negara? dikarenakan pemilik data harus dimintai pertanggungjawaban karena data tersebut dapat digunakan oleh orang yang tidak memenuhi syarat untuk melakukan kejahatan. Keamanan data ini memiliki beberapa aspek: (Harun, 2018):

1. Privasi (Kerahasiaan)

   Secara umum, setiap orang tidak ingin data mereka diketahui oleh orang asing. Selain itu, data ini melibatkan hak akses yang mahal, seperti kata sandi untuk mengakses kartu kredit. Pada umumnya terdapat 3 (tiga) aspek dari privasi yaitu pertama privasi mengenai pribadi seseorang (*Privasy of a Person's Persona*), kedua privasi tentang data seseorang (*Privasy of Data about a Parson*), dan ketiga privasi atas komunikasi seseorang (*Privasy of a Person's Communication*). Penggunaan data seseorang oleh lembaga pemerintah ataupun swasta, perusahaan ataupun perseorangan tanpa seizin pemilik merupakan pelanggaran privasi seseorang.

2. *Integrity* (Konsisten)

   Integritas data digunakan untuk memastikan bahwa data yang ada benar-benar asli. Integritas harus dapat memverifikasi bahwa data yang dikirim tidak berubah. Untuk memastikan data ini benar, dan dikirim oleh orang yang tepat juga memerlukan metode. Enkripsi adalah metode yang umum digunakan.

3. *Authenticity* (Keaslian)

   Keaslian data yang diterima oleh penerima informasi hendaknya benar-benar terjaga. Jika data yang didapatkan ternyata sudah diganti oleh orang yang tidak berhak maka akan sangat berbahaya. Jadi, keaslian data merupakan hal yang sangat penting.

4. *Avabaility* (Ketersediaan)

   Data dan informasi yang terdapat dalam sistem komputer tersedia dan dapat digunakan oleh orang yang berwenang. Aspek *avabaility* atau ketersediaan berhubungan dengan ketersediaan informasi saat dibutuhkan.

5. *Access Control*

   Merupakan cara mengakses informasi pada suatu sistem komputer. Contoh konkret yang sering kita jumpai adalah saat mengunakan komputer, sering kali komputer itu diberikan *user id* dan *password* dengan demikian akan jelas siapa yang sedang mengakses komputer. Boleh juga dijelaskan bahwa *Accsess Control* adalah sebuah mekanisme untuk mengatur ‘siapa dan boleh melakukan apa’, ‘dari mana boleh kemana’.

2.2. Pesan Teks

Pesan di bahasa Perancis adalah pesan tertulis (diucapkan: *mesaz*), berasal dari bahasa Latin "*missus*", yang berarti mengirim. Sejak akhir abad ke-11, pembicara atau peserta komunikasi telah menggunakan kata "pesan" yang berarti "apa yang kita kirim". Menurut Hafied Cangara, pesan adalah sesuatu yang disampaikan oleh pengirim kepada penerima.

Menggunakan kata “pesan” sebagai unsur komunikasi untuk memasukkan informasi yang dikirimkan oleh sumber pesan kepada penerima, seperti percakapan langsung atau melalui media massa, seperti telepon, media cetak, telepon genggam, internet, dan produk elektronik lainnya, dikemas dalam bentuk pesan. Isinya bisa berupa ilmu pengetahuan, hiburan, informasi, nasehat atau publisitas (Purwasito, 2017). Teks adalah sekelompok karakter yang terdiri dari huruf, angka, dan simbol. Kode ASCII digunakan untuk menunjukkan bahwa setiap karakter teks adalah 1 byte atau 8 *bit*.

Sesuai dengan yang telah diuraikan, dapat disimpulkan bahwa pesan teks merupakan representasi dari pikiran komunikator, dipertukarkan dalam bentuk simbol-simbol tertentu (huruf, angka, dan simbol), dan isinya mengandung tujuan tertentu. Berita biasanya disampaikan secara sengaja oleh penyebar kepada

penyebar untuk memperoleh hasil tertentu, dan hasil ini biasanya sudah ditentukan sebelumnya.

2.3. Audio

Audio adalah bunyi yang dihasilkan ketika molekul-molekul di udara berubah karena adanya gerakan yang disebabkan oleh benda yang menghasilkan getaran. Contohnya petikan gitar, suara, atau toples yang bergerak ketika ada energi untuk menggetarkannya. Saat program memainkan senar gitar pada gitar, senar akan bergerak maju mundur dengan jumlah getaran tertentu. Jumlah getaran ini disebut frekuensi getaran. Gerakan bolak-balik disebut *cycle* (putaran). Kemudian satuan frekuensi adalah *Cycle Per Second* atau CPS. Satuan ini sering disebut *Hertz* (Hz). *Kilohertz* (KHz) digunakan ketika getaran yang dihasilkan sangat cepat, dan perbandingan gerak yang dihasilkan sampai ribuan.

1. Audio Analog

   Audio analog merupakan format yang hanya dapat menyimpan suara dalam jumlah terbatas karena audio analog iyalah jenis koordinasi suara yang hanya dapat dihasilkan oleh suara sintetis yang disimpan dalam bentuk media kaset.

2. Audio Digital

   Audio digital merupakan Format digital dapat menyimpan data dalam jumlah besar melalui jaringan jangka panjang dan tersebar luas karena audio jenis ini menggunakan harmonisasi suara yang dibuat dengan menggunakan rekaman konvensional dan suara sintetis yang disimpan pada media berbasis teknologi komputer. Audio digital menggunakan sinyal digital untuk memutar suara. Dalam proses digitalisasi format rekaman musik analog, lagu atau musik digital memiliki format yang berbeda-beda tergantung dari teknologi yang digunakan.

2.4. Format M4A

M4A adalah format audio berdasarkan MPEG-4, tidak sama seperti MP4 yang berisi video. M4A adalah *file* MPEG dengan 4 lapisan yang berisi audio.

Format ini terutama digunakan untuk penyimpanan *file* audio dan video terkompresi.

Format ini mulai popular ketika Apple menggunakan *file* M4A di layanan iTunes. Karena ukurannya yang lebih kecil dan kualitas yang lebih baik dan format M4A yang telah disempurnakan, M4A diperkenalkan untuk menggantikan format MP3. Selain itu format ini juga memiliki kekayaan audio yang luas dibandingkan dengan MP3, *file* M4A biasanya lebih baik. Ini karena format M4A yang menggantikan MP3 telah diperbaiki.

2.5. Steganografi

Steganografi ("tersembunyi/tersembunyi" dalam bahasa Yunani, dan grapfein, "menulis") adalah seni dan ilmu komunikasi dengan menyembunyikan keberadaan informasi dari komunikasi. Steganografi menyembunyikan keberadaan pesan dengan menyematkannya dalam media yang disebut *file* pembawa (Narayana, 2010). Kegunaan steganografi adalah untuk menyembunyikan keberadaan data yang bersifat rahasia, sehingga sulit ditemukan, dan untuk melindungi hak cipta produk. Informasi rahasia yang tersembunyi dapat ditemukan kembali seperti informasi aslinya (Abdul, 2017).

Menyembunyikan data rahasia di media digital mengubah kualitas media. Kriteria yang perlu dipertimbangkan saat menyembunyikan data meliputi : (Kuniadi, 2017)

1. *Fidelity*

   Kualitas *file* kontainer tidak banyak berubah. *File* steganografi terlihat bagus bahkan setelah menambahkan data sensitif. Pengamat tidak menyadari bahwa *file* tersebut berisi data sensitif.

2. *Robustness*

   Data tersembunyi harus mampu menahan operasi yang dilakukan pada audio kontainer, dan data tersembunyi tidak boleh rusak.

3. *Recovery*

   Harus dimungkinkan untuk mengungkap kembali data yang tersembunyi (*recovery*). Karena tujuan steganografi adalah untuk menyembunyikan data, data

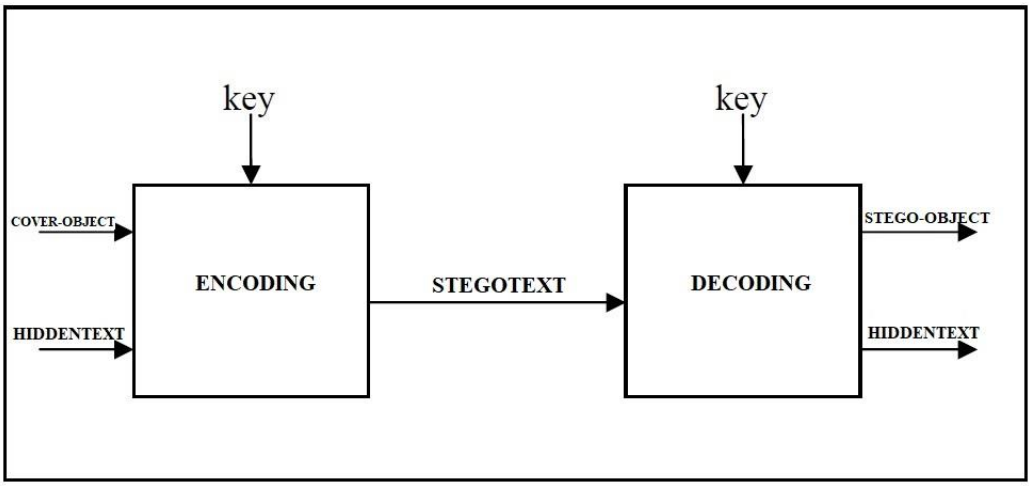
rahasia dalam *file* kontainer harus diambil setiap saat untuk digunakan lebih lanjut.

2.5.1. Konsep Steganografi

Ada beberapa istilah yang berkaitan dengan pokok bahasan steganografi:

1. *Hidden text* : Pesan yang disembunyikan.
2. *cover-object* : Untuk menyembunyikan *Hidden text*
3. *stego-object* : Pesan yang sudah berisi *Hidden text*

Dalam steganografi *hidden text* yang dimaksudkan adalah teks yang akan disisipkan ke dalam *cover-object* yaitu *file* audio yang digunakan sebagai media penampung pesan yang akan disisipkan. Dari hasil *encoding* pesan kedalam *file* audio akan dihasilkan *stego-object* yang merupakan *file* audio yang berisikan pesan *embedding*. Penyisipan pesan ke dalam media *cover-object* dinamakan *encoding*, sedangkan ekstraksi pesan dari *stego-object* dinamakan *decoding*. Kedua proses ini memerlukan kunci rahasia (*stegokey*) agar hanya pihak yang berhak saja yang dapat melakukan penyisipan dan ekstraksi pesan (Anti, 2017), seperti pada gambar di bawah ini



Gambar 2.1 Proses Steganografi

Faktanya, tidak semua data dapat digunakan sebagai *cover* transmisi data rahasia. Perlu mengubah data agar tidak menampilkan data rahasia. Data yang digunakan sebagai *cover* harus lebih besar dari data rahasia agar tidak terlihat. Data yang digunakan sebagai cover harus digunakan satu kali. Jika menggunakannya

berkali-kali, kecurigaan pihak lain akan meningkat (Saragih, 2006). Dalam steganografi, format *file* audio lebih unggul dari format *file* citra dan video. Dibandingkan dengan format *file* citra gambar, *file* audio biasanya berukuran relatif besar, sehingga dapat menyimpan lebih banyak pesan rahasia. Format *file* video relatif sangat besar, tetapi ukurannya yang besar mengurangi kepraktisannya, dan juga kurangmya algoritma untuk mendukung formatnya (Furqan, 2020).

Inilah performa yang harus diperhatikan dalam penyembunyian ke dalam format *file* audio, yaitu (Riko, 2006):

1. Kualitas Audio

   Dasar terpenting dari steganografi adalah bahwa penyisipan data tidak boleh mengubah kualitas suara dari sinyal audio yang digunakan sebagai *cover-object*. Oleh karena itu, pendengar tidak dapat mendeteksi data rahasia yang disisipkan. Kinerja ini sangat penting dalam aplikasi perlindungan hak cipta.

2. *Bit Rate*

   Tujuan dari *bit rate* adalah untuk menghitung jumlah data rahasia yang mungkin dimasukkan ke sinyal audio *cover-object* per satuan waktu.

3. Keamanan

   Agar pesan yang dimasukkan oleh pihak lain tidak dikenali dan dihapus, maka proses penyisipannya harus dilakukan seaman mungkin. Setiap aplikasi membutuhkan tingkat keamanan yang berbeda. Jika algoritma ini diketahui dapat digunakan, skema penyisipan data aman. Sekalipun pengirim dan penerima mengetahui bahwa sinyal audio berisi data lain dan mengetahui cara memasukkan data, pihak selain pengirim dan penerima tidak dapat mengambil data tersebut.

4. Kesulitan Perhitungan

   Kesulitan komputasi didasarkan pada proses yang diperlukan untuk memasukkan data ke dalam sinyal audio dan mengekstrak data dari sinyal audio. Ini bagian paling penting jika aplikasi digunakan secara online untuk memasukkan dan mengambil data. Kesulitan algoritma juga merupakan faktor penting yang mempengaruhi pilihan struktur pemasangan atau arsitektur pemrosesan sinyal digital.

2.5.2. Metode-Metode Steganografi

Ada beberapa metode yang dapat digunakan untuk menyembunyikan informasi digital di dalam informasi digital lainnya (steganografi), antara lain:

1. *Least Significant Bit* (LSB)

   LSB (*Least Significant Bit*), adalah salah satu metode umum yang digunakan untuk steganografi yaitu untuk memasukkan setiap bagian dari pesan dalam bit paling tidak signifikan dari *cover* audio secara deterministik. (Sayed, 2020).

2. *Spread Spectrum*

   Spread spectrum adalah metode lain yang digunakan untuk menyimpan informasi tersembunyi dalam *file* audio. Metode ini bekerja dengan menyandikan pesan dan menyebarkannya ke semua *spektrum frekuensi* yang memungkinkan. Metode ini sulit untuk diselesaikan kecuali data yang disimpan dapat diakses atau sinyal acak yang digunakan untuk menyebarkan pesan direkonstruksi

3. *Data Echo Hiding*

   Metode *Data Echo Hiding* atau yang sering disebut dengan *Echo Hiding* melakukan proses penyisipan dengan menambahkan *echo* pada sinyal suara kedalam data suara digital (Piarsa, 2010).

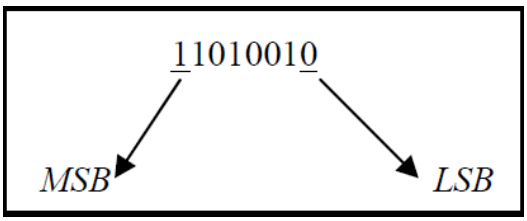
4. *Redundant Pattern Encoding*

   Salah satu cara untuk menanamkan pesan dalam teknologi steganografi. Metode pengkodean pola redundan memasukkan redundansi (penggandaan) dalam informasi atau pesan yang akan disembunyikan (jika salinannya sama dengan pesan asli atau belum dimodifikasi) dan menyebarkan pesan ke seluruh *file* penampung (Firman, 2010).

2.6. Metode *Least Significant Bit* (LSB)

Metode Steganografi yang paling populer pada format suara adalah *Least Significant Bit.* Metode ini banyak digunakan karena perhitungan yang tidak terlalu rumit dan pesan tersembunyi sangat aman. Metode ini mengubah niali bit yang paling kurang signifikan dari jumlah bit dalam 1 *byte file carrier*. Bit yang memiliki signifikansi paling tinggi adalah angka dengan nilai tertinggi (misal, $2^7 = 128$), Ini

berarti bahwa mengubah bit ini akan menghasilkan perubahan yang sangat signifikan. Bit yang memiliki signifikansi paling rendah adalah angka dengan nilai terendah (misal, $2^0 = 1$), artinya jika terjadi perubahan pada bagian ini maka akan mengakibatkan perubahan yang sangat tidak signifikan (Utami, 2009). Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit paling signifikan yang disebut MSB (*Most Significant Bit*) dan bit yang paling tidak signifikan atau LSB (*Least Significant Bit*). Seperti gambar berikut.



Gambar 2.2 Susunan Bit Pada MSB dan LSB

Misalnya, akan dilakukan proses penyembunyian karakter ‘L’ (ASCII 76) pada *file carrier* yang berukuran 8 *byte*. Berkas *carrier* berukuran 8 *byte* dalam biner :

‘10010101 00001101 11001001 10010110
00001111 11001011 10011111 00010000’

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

‘01001100’

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

‘10010100 00001101 11001000 10010110
00001111 11001011 10011110 00010000’

Pada contoh di atas, perubahan hanya terjadi pada bit-bit paling terakhir dari *file carrier* (ditandai dengan karakter berwarna merah). Berdasarkan contoh di atas, kita dapat memperoleh teori probabilitas bahwa sedikit akan berubah adalah sekitar 50%. Ini karena kemungkinan perubahan adalah antara 1 dan 0, dan mengubah LSB tidak mengubah ukuran *file* pembawa dan akan sulit dideteksi.

Ketika semua informasi telah dilampirkan ke *file* pembawa, outputnya disebut *Stegofile*. Jika suatu informasi yang disematkan akan diekstraksi kembali, maka bit-bit yang paling tidak signifikan pada *stegofile* akan diambil dan disatukan kembali sehingga menjadi informasi atau disebut *decoding/retrieving* (Oktaviani, 2015).

2.6.1. Proses Penyisipan Dengan LSB

Proses penyisipan (*embedding*) menggunakan metode *least significant bit* adalah sebagai berikut :

1. *Input carrier-audio* format (.m4a).
2. *Input file* teks (.txt) yang akan disisipkan..
3. Ubah *carrier-audio* ke dalam *byte*.
4. Ubah *file* teks ke dalam *byte*.
5. Ubah nilai bit audio terakhit dengan *bit file* teks yang akan disisipkan.
6. Kelompokkan menjadi *file audio* yang baru atau *stego-audio* (.m4a).

Contoh :

Menyisipkan karakter ‘b’ (ASCII 98) ke dalam *file audio.*

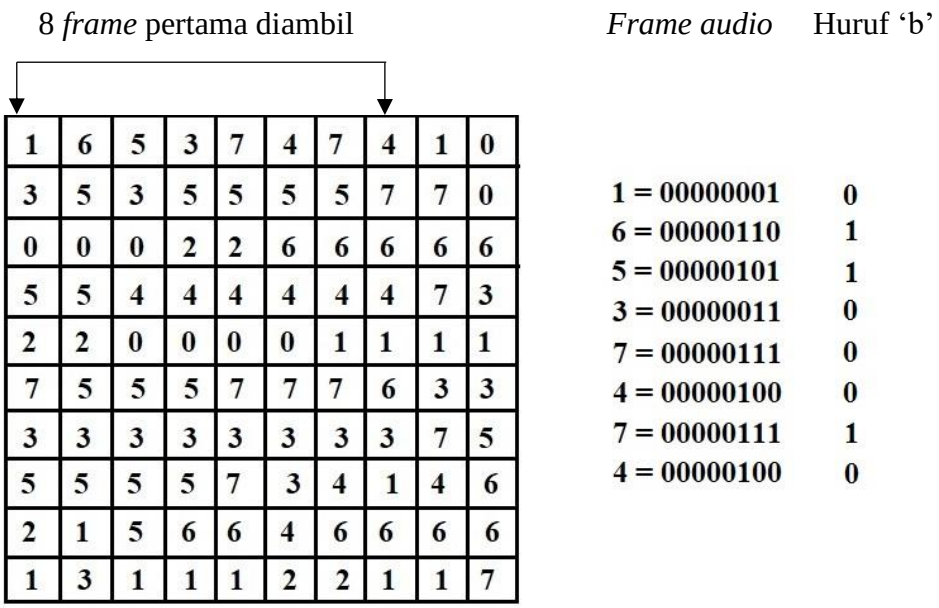
Penyelesaiannya :

File Audio

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	7	3	4	1	4	6
2	1	5	6	6	4	6	6	6	6
1	3	1	1	1	2	2	1	1	7

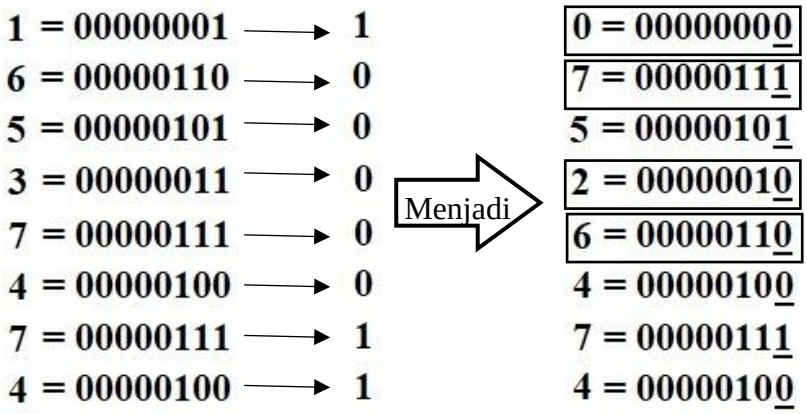
Gambar 2.3 *Frame Audio*

Yang pertama harus dilakukan adalah mengganti data huruf ‘b’ dan data *file audio* menjadi bilangan biner. 01100010 adalah nilai biner dari ‘b’. Karena jumlah digit biner huruf ‘b’ hanya 8 bit maka jumlah *frame audio* yang dibutuhkan cukup 8 *frame* saja. Berikut 8 *frame* awal dari *audio* yang diubah menjadi biner.

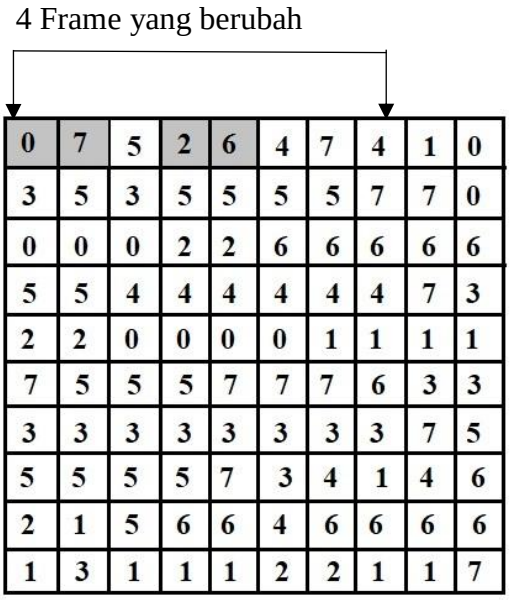


Gambar 2.4 Pengubahan *File* Audio Menjadi Biner

Selanjutnya ialah mengubah bit terakhir dari *frame* audio dengan bit-bit dari huruf “b”.



Gambar 2.5 Penyisipan dengan LSB



Gambar 2.6 Perubahan yang terjadi pada *frame audio*

Hasil akhirnya *frame* audio yang diubah intensitasnya hanya ±1. Jadi, dari perspektif *visual*, ini sangat berpengaruh. Semua *frame* kecuali beberapa mengalami perubahan intensitas.

2.6.2. Proses Ekstraksi

Proses ekstraksi menggunakan metode LSB adalah sebagai berikut:

1. *Input stego-file* (.m4a).
2. Ganti nilai audio yang telah disisipkan pesan ke dalam *byte*.
3. Pisahkan setiap bit terakhir audio.
4. Kelompokkan setiap 8 bit menjadi {b1, b2, …, bn}.
5. Ganti setiap kelompok menjadi teks.

Contoh :

Mendeteksi pesan yang disisipkan pada *file audio* (.m4a).

0	7	5	2	6	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	7	3	4	1	4	6
2	1	5	6	6	4	6	6	6	6
1	3	1	1	1	2	2	1	1	7

Gambar 2.7 *File audio* yang telah disisipkan pesan

Yang pertama harus dilakukan adalah mengubah *file* audio (.m4a) menjadi biner dan mengambil setiap bit terakhir *setego-audio*

Frame Audio

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	7	3	4	1	4	6
2	1	5	6	6	4	6	6	6	6
1	3	1	1	1	2	2	1	1	7

0 = 00000000
7 = 00000111
5 = 00000101
2 = 00000010
6 = 00000110
4 = 00000100
7 = 00000111
4 = 00000100

Gambar 2.8 Pengubahan *stego-audio* menjadi biner

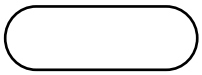
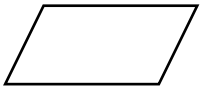
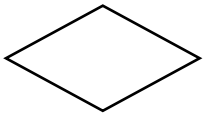
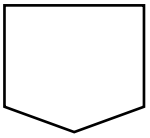
Selanjutnya iyalah menyusun bit dan dikelompokkan, setiap kelompok terdiri dari 8 bit. Karena pada contoh ini hanya ada 8 bit yang berubah, jadi hanya ada 1 blok saja yaitu : "0 1 1 0 0 0 1 0". Selanjutnya kelompok tersebut diubah menjadi plainteks. Maka yang akan terbaca yaitu huruf "b" (Syafitri, 2013).

2.7. *Flowchart*

Flowchart adalah representasi grafis dari langkah-langkah dalam sebuah program. Flowchart membantu proses pemecahan masalah menjadi skala yang

lebih kecil dan membantu menganalisis cara lain dalam operasinya. Salah satu tujuan dari *flowchart* adalah untuk membantu menyelesaikan masalah dengan lebih mudah.

Tabel 2.1 Simbol *Flowchart*

	Flow Direction Mempresentasikan alur kerja		***Terminator*** Awal atau akhir flowchart
	***Process*** Proses Perhitungan		***Input/Output Data*** *Input* data atau output data
	***Decision*** Simbol pemilihan proses berdasarkan kondisi		***Predefine Process*** Permulaan sub program
	***Preparation*** Mempersentasikan inisialisasi		***On Page Connector*** Penyambungan proses pada halaman yang sama
	***Document*** I/O dalam format yang dicetak		***Off Page Connector*** Penyambung proses pada halaman yang beda

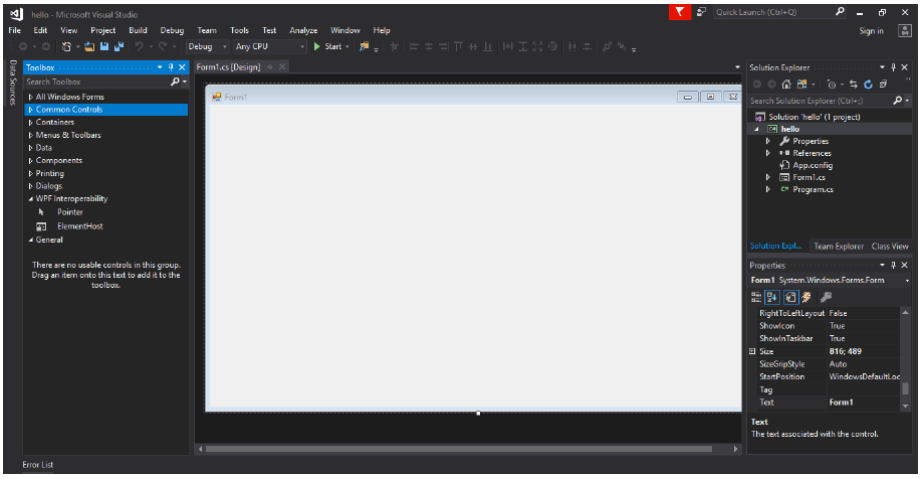
2.8. Microsoft Visual Studio

Visual Studio adalah IDE (*Integrated Development Environment*) yang dapat digunakan untuk mengembangkan aplikasi-aplikasi *Windows*. *Visual Studio* dirancang untuk fokus pada produktifitas. *Tool* ini disebut juga disebut *Rapid Application Development tools* (RAD tools) karena dirancang dan dilengkapi untuk meningkatkan produktifitas. *Visual Studio* juga memudahkan pengguna untuk belajar dan memenuhi kebutuhan *programmer*.

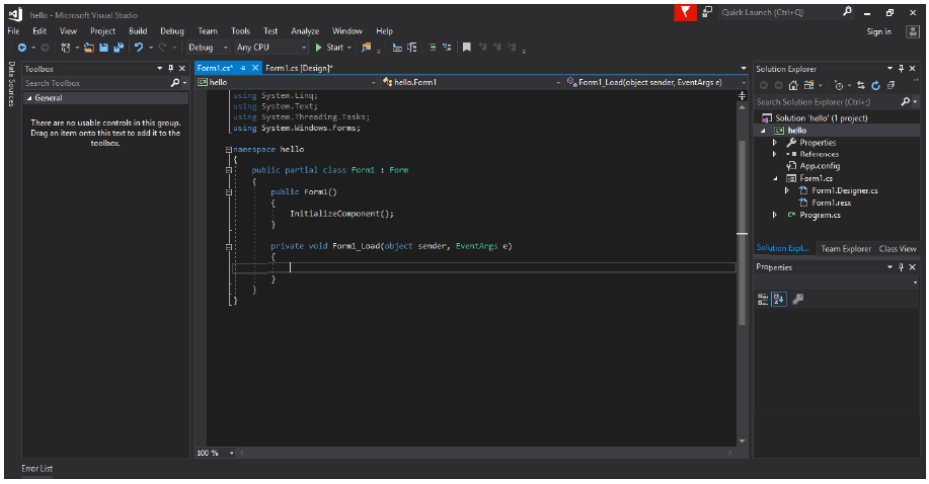
Visual Studio tool merupakan *tool* yang ideal untuk membangun sebuah aplikasi secara cepat. Pekerjaan kantor atau tugas sekolah juga dapat diselesaikan

menggunakan *tool* ini. Dengan *tool* ini, kita bisa belajar pemrograman dengan cara yang tidak membosankan dan lebih menyenangkan. Maka dari itu, *Visual Studio* didesain untuk pada *programmer* baik pemula, orang-orang yang ingin mempelajari pemrograman, mereka yang mencari cara mudah untuk membuat aplikasi *windows* dan juga pada *programmer* ahli.

Pada penelitian ini, bahasa pemrograman yang digunakan yaitu bahasa C#. Dan untuk simulasi menggunakan fasilitas GUI (*Grapichal User Interface*) yang ada di *Microsoft Visual Studio* yaitu Windows *Forms Designer*. Pada tampilan *windows forms* ada dua bagian yang digunakan dalam pengerjaan suatu aplikasi. Merancang dari tampilan GUI dan bagian penulisan kode program untuk GUI agar lebih dinamis.



Gambar 2.9 Tampilan Form GUI



Gambar 2.10 Tampilan Form Editor

2.8.1. Bahasa C#

Visual C# merupakan bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa C# ini adalah untuk menyederhanakan bahasa pemrograman pada platform.NET yang diterbitkam pada tahun 2002. Bahasa C# secara teknis mengadopsi sintak bahasa C/C++ tetapi lebih dipermudah karena tidak akan dipusingkan dengan *memory management.* Konsistensi API membuat bahasa C# mendukung *object-oriented* dan juga *dynamics programing, i*ni menambah kemudahan untuk pengguna bahasa C#.

2.9. Penelitian Terkait

Adapun penelitian terkait judul yang diambil dapat dilihat pada tabel di bawah ini:

Tabel 2.2 Penelitian Terkait

No.	Penelitian	Asal & Tahun	Judul	Kesimpulan
1	Assyahid, Muhammad Maulana Rihartanto, Rihartanto Utomo,Didi Susilo Budi	Politeknik Negeri Samarinda, 2018	Implementasi Steganografi Pesan Text ke Dalam Audio Dengan Metode Spread Spectrum	Pengujian kinerja MSE dan PSNR menunjukkan bahwa pengujian tersebut tiga kali rata-rata saat menggunakan data audio yang berlangsung selama dua detik dan tiga pesan dengan jumlah karakter yang berbeda. *Mean Squared Error* adalah 2.2692e-06 dengan nilai rata-rata *Peak Signal to-Noise Ratio* (PSNR) sebesar 111,9842 dB.
2	Toni Sahata Pandapotan, Taronisokhi Zebua, M.Kom	Teknik Informatika Komputer STMIK Budi Darma Medan, 2016	Analisa Perbandingan Least Significant Bit dan End of *File* Untuk Steganografi Citra Digital	Metode LSB memperoleh keuntungan lebih dari metode EOF karena citra setelah pesan disisipkan hanya mengalami sedikit penurunan kualitas, yang tidak memberikan pengaruh yang signifikan

			Menggunakan Matlab	jika dilihat oleh mata manusia, sedangkan metode EOF mengalami perubahan yang signifikan dalam hal kualitas.
3	Dyna Marisa Khairina, Ulan Ari Anti, Awang Harsa Kridalaksana,	Teknologi Informasi Universitas Mulawarman, 2017	Steganografi Pada Video Menggunakan Metode Least Significant Bit (LSB) Dan End Of *File* (EOF)	Aplikasi stego video ini mampu menyimpan pesan teks, namun ukuran teks tidak melebihi kapasitas *frame cover* video. Jumlah waktu, jumlah *frame*, dan dimensi video sangat menentukan berapa lama waktu yang dibutuhkan steganografi menggunakan metode EOF dan LSB. Penyematan teks LSB membutuhkan waktu lebih sedikit daripada EOF. Sebaliknya, ekstraksi EOF membutuhkan waktu lebih sedikit daripada LSB.
4	Teguh Budi Harjo1, Marly Kapriati2, Dwi Andrian Susanto	Universitas Budi Luhur, 2016	Aplikasi Steganografi Menggunakan LSB (Least Significant Bit) dan Enkripsi Triple Des Menggunakan Bahasa Pemrograman C#	Pesan atau dokumen yang termasuk dalam *file* gambar dapat dipulihkan sepenuhnya atau dengan kata lain pesan yang dimasukkan sebelum proses *enkripsi* dan setelah proses *dekripsi* memiliki hasil yang sama tanpa ada perubahan atau gangguan pesan. Pada metode LSB, citra yang disisipkan tidak menunjukkan perbedaan yang terlalu jauh dengan citra berwarna kecuali jika dimasukkan pesan atau dokumen dengan ukuran yang besar.

BAB III
METODOLOGI PENELITIAN

3.1. Tempat Waktu dan Penelitian

3.1.1. Tempat Penelitian

Tempat penelitian ini diadakan di Laboratorium Multimedia Fakultas Sains dan Teknologi yang berada di Jalan IAIN No. 1 Medan, Sumatera Utara.

3.1.2. Waktu Penelitian

Waktu penelitian ini dilaksanakan pada semester ganjil tahun ajaran 2020/2021 yaitu antara bulan Mei sampai dengan bulan Agustus 2020 dengan tabel sebagai berikut :

Tabel 3.1 Waktu Penelitian

No	Kegiatan	Bulan 2020			
		Mei	Juni	Juli	Agustus
1.	Tahap Persiapan Penelitian				
	a. Penyusunan dan Pengajuan Judul				
	b. Pengajuan Proposal				
	c. Perijinan Penelitian				
2.	Tahap Pelaksanaan				
	a. Analisis Data				
3.	Tahap Penyusunan Laporan				

3.2. Bahan dan Alat Penelitian

3.2.1. Perangkat Keras

Perangkat keras yang digunakan pada pembuat sistem ini diperlukan sebagai berikut:

1. Prosessor : Inter(R) Core(TM) i3-005U CPU @ 2.00GHz
2. Memori : *Random Acces Memory* (RAM) 4GB

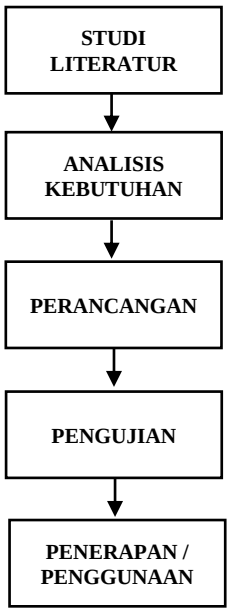
3.2.2. Perangkat Lunak

Perangkat lunak yang digunakan pada pembuatan sistem ini diperlukan sebagai berikut :

1. *Operating System Windows* 10 64 bit
2. *Microsoft Visual Studio* 2012

3.3. Prosedur Kerja

Prosedur kerja dalam Penelitian ini dilakukan melalui tahapan-tahapan, sebagai berikut :



Gambar 3.1 Diagram Alir Prosedur Kerja

3.3.1. Teknik Pengumpulan Data

Pengumpulan data adalah cara untuk mengumpulkan informasi tentang kasus dan masalah laporan ini. Yang paling dibutuhkan penulis adalah informasi yang relevan tentang metode yang digunakan dalam studi kasus ini, yaitu *least significant* bit (LSB). Ada dua cara yang dilakukan penulis adalah mendapatkan informasi atau mengumpulkan data, yaitu:

1. Penelitian kepustakaan

   Metode penelitian kepustakaan adalah sumber data yang dapat digunakan untuk meneliti. Metode penelitian kepustakaan bersumber teori dari buku-buku referensi, penelitian sebelumnya (seperti jurnal dan tesis), sehingga memiliki orientasi yang luas dalam masalah yang dipilih dan diangkat dengan cara mengumpulkan teori, metode dan teknik penelitian yang terkait dengan penulisan dan analisis data dalam penelitian sebagai bahan pelengkap penelitian ini.

2. Study Literatural

   Studi Literatural merupakan rangkaian proses yang berkaitan dengan proses pencarain informasi, termasuk membaca dan mencatat, serta mengolah bahan penelitian atau mencari referensi terkait kasus atau masalah yang berkaitan dengan tugas akhir. Tujuannya adalah untuk memberikan kerangka kerja bagi pengembangan tinjauan konseptual pada metode penelitian dengan menggunakan tinjauan pustaka.

3.3.2. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap setelah pengumpulan data dan informasi kasus dalam penelitian ini. Analisis berarti suatu metode khusus untuk menganalisis masalah dalam kasus-kasus yang timbul dalam perjalanan penelitian, dan dapat dimulai dengan analisis plot steganografi, lalu menganalisa model hingga rancang bangun aplikasi steganografi mengunakan metode LSB.

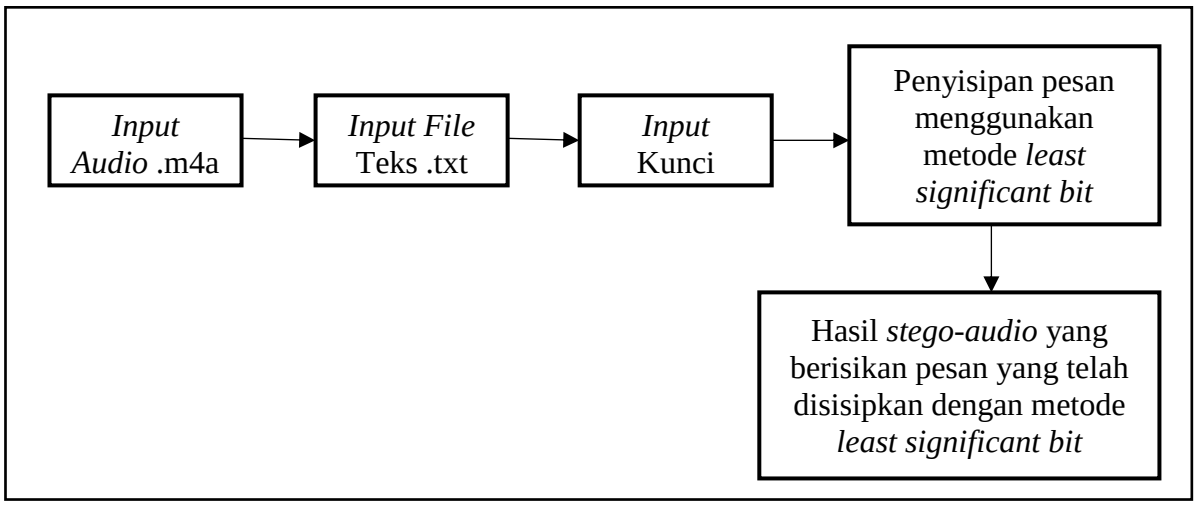
Dalam perancangan sistem steganografi audio ini, diperlukan *file* audio (.m4a) yang akan digunakan sebagai kontainer penampung pesan dan juga *file* teks

(.txt) sebagai pesan rahasia yang akan disisipkan menggunakan metode *least significant* bit.

3.3.3. Perancangan

Setelah melakukan tahap analisis kebutuhan, dapat dilanjutkan dengan perancangan sistem. Dikarenakan proses steganografi memiliki dua proses utama, yaitu *encoding* dan *decoding.* Maka, Perancangan yang akan dijelaskan meliputi diagram proses *encoding* dan *encoding*.

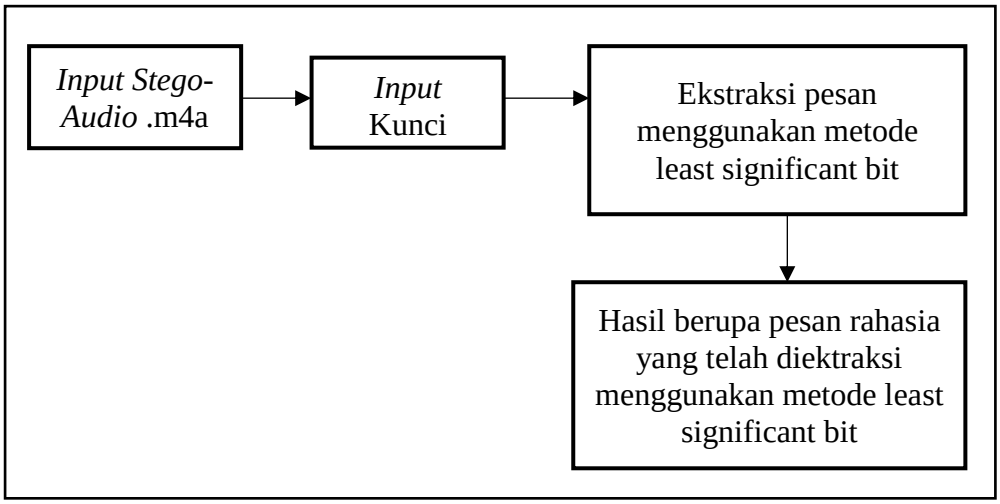
1. Diagram Proses *Encoding*



Gambar 3.2 Diagram Proses *Encoding*

Proses penyembunyian pesan ke dalam *file* penampung (.m4a) pada gambar 3.2 dimulai dengan memasukkan *Carrier file* (.m4a) kemudian *input* pesan yang akan disembunyikan *file* teks (.txt) yang sudah disiapkan dan *input* kunci sebagai *password* yang diketahui oleh pengirim dan penerima. Proses *Encoding* ke dalam *file* penampung dimulai dengan mengganti *file* audio (.m4a) dan *file* teks (.txt) menjadi *byte*, kemudian ganti bit terakhir pada audio dengan bit pesan yang akan disisipkan, lalu tetapkan menjadi *file* audio yang baru atau *stego-audio* (.m4a).

2. Diagram Proses *Decoding*



Gambar 3.3 Diagram Proses *Decoding*

Proses pengambilan informasi dari stego-audio (.m4a) seperti pada gambar 3.3 dimulai dengan *input* stego-audio(.m4a), *input* juga kunci sebagai password apabila kunci sesuai maka akan lanjut ke proses *Decoding* jika kunci tidak sesuai maka kembali ke tahap *input* kunci. Proses *Decoding* dimulai dengan mengubah stego-audio ke dalam bentuk byte lalu mengambil setiap bit terakhir pada stego-audio, kemudian setiap blok 8-bit menjadi {b1,b2,…, bn} lalu ubah setiap blok bit menjadi teks, maka *output* adalah pesan rahasia yang telah disisipkan.

Ada dua proses utama dalam steganografi utama yaitu proses *Encoding* atau penyisipan informasi pada *file* penampung dan proses *Decoding* atau pengambilan informasi dari *file* penampung. Berdasarkan diagram sebelumnya bahwa proses *Encoding* membutuhkan tiga *input* dan menghasilkan satu *output*, *input* berupa *file* penampung (.m4a), pesan (.txt), dan *file* kunci sedangkan *output* berupa *file* penampung (.m4a). Proses *Decoding* membutuhkan dua *input* dan menghasilkan satu *output*. *Input* berupa *file* penampung format (.m4a) dan *file* kunci. Sedangkan *output* berupa pesan yang disisipkan (.txt).

3.3.4. Pengujian

Penelitian ini perlu dilakukan pengujian agar mengetahui apakah sistem bekerja dengan baik dan mencari celah dimana kesalahan dapat terjadi dalam sistem

aplikasi. Untuk mengetahui perbandingan kualitas audio sebelum dan sesudah proses penyisipan pesan, pengujian pada penelitian ini yaitu terhadap kualitas *stegofile* akan menggunakan *software* spek dengan melihat grafik dari dB audio.

3.3.5. Penerapan / Penggunaan

Penerapan / Penggunaan pada sistem ini ialah penyembunyian atau penyisipan pesan rahasia berupa teks (.txt) ke dalam wadah penampung berupa audio (.m4a) dimana bukan hanya penyisipan tetapi juga dapat mengambil kembali suatu informasi yang telah disisipkan. Dengan menggunakan sistem aplikasi keamanan data dengan menggunakan metode LSB bermaanfaat untuk membantu menyembunyikan pesan penting sehingga meminimalisir pencurian digital oleh pihak yang tidak bertanggung jawab.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pembahasan

Pada penelitian ini, pembahasan terdiri dari analisis data, representasi data Hasil analisis data, serta perancangan *interface*.

4.1.1 Analisis Data

Proses penyisipan data teks dengan teknik stganografi memerlukan sebuah objek, pada penelitian ini data objek yang digunakan adalah audio dengan format m4a. Adapun audio yang digunakan adalah audio sampel dengan nama sample1.m4a seperti pada gambar di bawah ini :

Gambar 4.1 *File* Audio Sampel

Berdasarkan pada gambar 4.1, didapati sebuah objek audio yang akan menjadi penampung nilai sebuah teks yang berisi karakter. Selanjutnya adalah menentukan teks karakter yang akan disisipkan pada objek audio sample. Adapun contoh data teks yang akan disisipkan pada proses penerapan manual adalah teks dengan karakter " AINUL".

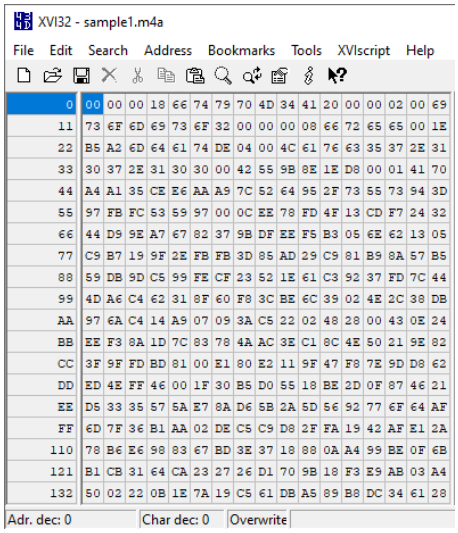
4.1.2 Representasi Data

Berdasarkan pada analisis data, didapatkan objek audio sebagai media penampung dari karakter teks yang akan disisipkan. Selanjutnya untuk proses penyisisipan dengan metode LSB, *file* audio sample diekstraksi terlebih dahulu nilainya.

1. Data Audio Sample

   Pada penelitian ini penulis menggunakan aplikasi XV32 dimana aplikasi tersebut dapat merubah *audio* M4a menjadi bilangan hexa dengan cara memasukkan *audio* M4a kedalam aplikasi dan secara otomatis nilai hexa dari

audio M4a akan muncul untuk keperluan hitungan manual. Adapun nilai audio yang telah didapatkan menggunakan bantuan aplikasi XV32 adalah sebagai berikut:



Gambar 4.2 Nilai Hexa Audio Sample

Berdasarkan pada gambar 4.2, diambil nilai sample audio sebanyak 40 *byte* dalam bentuk hexa dan dirubah kedalam bentuk biner seperti pada tabel di bawah ini:

Tabel 4.1 Nilai Biner Audio Sampel

No	Audio Objek		
	Hexa	Des	Biner
1	00	0	00000000
2	00	0	00000000
3	00	0	00000000
4	18	24	00011000
5	66	102	01100110
6	74	116	01110100
7	79	121	01111001
8	70	112	01110000
9	4D	77	01001101
10	34	52	00110100
11	41	65	01000001
12	20	32	00100000
13	00	0	00000000

Tabel Lanjutan 4.1 Nilai Biner Audio Sampel

No	Audio Objek		
	Hexa	Des	Biner
14	00	0	00000000
15	02	2	00000010
16	00	0	00000000
17	69	105	01101001
18	73	115	01110011
19	6F	111	01101111
20	6D	109	01101101
21	69	105	01101001
22	73	115	01110011
23	6F	111	01101111
24	32	50	00110010
25	00	0	00000000
26	00	0	00000000
27	00	0	00000000
28	08	8	00001000
29	66	102	01100110
30	72	114	01110010
31	65	101	01100101
32	65	101	01100101
33	00	0	00000000
34	1E	30	00011110
35	B5	181	10110101
36	A2	162	10100010
37	6D	109	01101101
38	64	100	01100100
39	61	97	01100001
40	74	116	01110100
41	DE	222	11011110
42	04	4	00000100
43	00	0	00000000
44	4C	76	01001100
45	61	97	01100001
46	76	118	01110110
47	63	99	01100011
48	35	53	00110101

Tabel Lanjutan 4.1 Nilai Biner Audio Sampel

No	Audio Objek		
	Hexa	Des	Biner
49	37	55	00110111
50	2E	46	00101110
51	31	49	00110001
52	30	48	00110000
53	37	55	00110111
54	2E	46	00101110
55	31	49	00110001
56	30	48	00110000

Berdasarkan pada tabel 4.1 di atas, didapati nilai biner dari audio sample yang akan dijadikan objek penampung data teks karakter “AINUL”.

2. Data Teks

Sebelum proses penyisipan terlebih dahulu setiap karakter dirubah kedalam bentuk biner, proses perubahan karakter kedalam bentuk biner dapat menggunakan tabel ASCII. Adapun nilai biner dari karakter “AINUL” adalah sebagai berikut “

Tabel 4.2 Biner Karakter

No.	Karakter	Nilai Desimal	Biner	Jumlah Bit
1	A	65	01000001	8
2	I	73	01001001	8
3	N	78	01001110	8
4	U	85	01010101	8
5	L	76	01001100	8
Total bit				40 Bit

Berdasarkan tabel 4.2 di atas, didapati data biner dari karakter teks yang akan disisipkan kedalam objek audio.

4.1.3 Hasil Analisis

1. Proses Penyisipan LSB

   Setelah nilai biner audio dan teks didapati, selanjutnya adalah melakukan proses penyisipan dengan metode *Least Significant Bit* (LSB). Pada proses penyisipan, steganografi membutuhkan sebuah kunci sebagai penanda awal serta penanda akhir sebagai pembatas dalam pengambilan bit biner pada audio m4a saat proses ekstraksi. Adapun penanda yang digunakan adalah karakter "#" yang dirubah kedalam bentuk biner **00100011**, sedangkan kunci yang digunakan dalam proses hitungan manual ini adalah *string* "UINSU1". Untuk mendapatkan nilai 8 bit kunci penanda awal dilakukan XOR nilai biner setiap karakter kunci seperti pada tabel berikut :

Tabel 4.3 Nilai Biner Kunci

Karakter	Nilai Desimal	Nilai Biner
U	85	01010101
I	73	01001001
N	78	01001110
S	83	01010011
U	85	01010101
1	1	00000001

Setiap nilai kunci akan dilakukan XOR seperti berikut ini :

U = 01010101
I = 01001001
_____ XOR
00011100
N= 01001110
_____XOR
01010010
S = 01010011
_____XOR
00000001
U = 01010101
_____XOR
01010100
1 = 00000001
_____XOR
01010101

Berdasarkan hasil XOR didapati nilai kunci stegano penanda awal adalah **01010101** dan nilai penanda akhir penyisipan teks adalah # dalam biner **00100011.** Pada bit data ke-8 dari nilai biner audio sample m4a proses penyisipkan data teks diaplikasikan pada tabel 4.1. Jumlah seluruh nilai yang bit yang akan dimasukkan kedalam bit audio adalah sebanyak 8 bit penanda awal, 40 bit nilai biner dari sampel karakter data teks dan 8 bit penanda akhir maka total seluruh bit yang akan dimasukkan pada audio sampel m4a adalah 56 bit. Berikut adalah keseluruhan data bit yang akan disisipan pada audio sample:

010101010100000101001001010011100101010101001100**00100011**

Proses menyisipkan atau memindahkan nilai biner pola karakter data teks pada objek audio m4a ditunjukkan pada tabel berikut:

Tabel 4.4 Proses Penyisipan Data Teks

No	Nilai Audio Awal			Nilai Bit Karakter	Nilai Audio Stegano		
	Hexa	Des	Biner		Hexa	Des	Biner
1	00	0	00000000	0	00	0	00000000
2	00	0	00000000	1	01	1	00000001
3	00	0	00000000	0	00	0	00000000
4	18	24	00011000	1	19	25	00011001
5	66	102	01100110	0	66	102	01100110
6	74	116	01110100	1	75	117	01110101
7	79	121	01111001	0	78	120	01111000
8	70	112	01110000	1	71	113	01110001
9	4D	77	01001101	0	4C	76	01001100
10	34	52	00110100	1	35	53	00110101
11	41	65	01000001	0	40	64	01000000
12	20	32	00100000	0	20	32	00100000
13	00	0	00000000	0	00	0	00000000
14	00	0	00000000	0	00	0	00000000
15	02	2	00000010	0	02	2	00000010
16	00	0	00000000	1	01	1	00000001

Tabel Lanjutan 4.4 Proses Penyisipan Data Teks

No	Nilai Audio Awal			Nilai Bit Karakter	Nilai Audio Stegano		
	Hexa	Des	Biner		Hexa	Des	Biner
17	69	105	01101001	0	68	104	01101000
18	73	115	01110011	1	73	115	01110011
19	6F	111	01101111	0	6E	110	01101110
20	6D	109	01101101	0	6C	108	01101100
21	69	105	01101001	1	69	105	01101001
22	73	115	01110011	0	72	114	01110010
23	6F	111	01101111	0	6E	110	01101110
24	32	50	00110010	1	33	51	00110011
25	00	0	00000000	0	00	0	00000000
26	00	0	00000000	1	01	1	00000001
27	00	0	00000000	0	00	0	00000000
28	08	8	00001000	0	08	8	00001000
29	66	102	01100110	1	67	103	01100111
30	72	114	01110010	1	73	115	01110011
31	65	101	01100101	1	65	101	01100101
32	65	101	01100101	0	64	100	01100100
33	00	0	00000000	0	00	0	00000000
34	1E	30	00011110	1	1F	31	00011111
35	B5	181	10110101	0	B4	180	10110100
36	A2	162	10100010	1	A3	163	10100011
37	6D	109	01101101	0	6C	108	01101100
38	64	100	01100100	1	65	101	01100101
39	61	97	01100001	0	60	96	01100000
40	74	116	01110100	1	75	117	01110101
41	DE	222	11011110	0	DE	222	11011110
42	04	4	00000100	1	05	5	00000101
43	00	0	00000000	0	00	0	00000000

Tabel Lanjutan 4.4 Proses Penyisipan Data Teks

No	Nilai Audio Awal			Nilai Bit Karakter	Nilai Audio Stegano		
	Hexa	Des	Biner		Hexa	Des	Biner
44	4C	76	01001100	0	4C	76	01001100
45	61	97	01100001	1	61	97	01100001
46	76	118	01110110	1	77	119	01110111
47	63	99	01100011	0	62	98	01100010
48	35	53	00110101	0	34	52	00110100
49	37	55	00110111	0	36	54	00110110
50	2E	46	00101110	0	2E	46	00101110
51	31	49	00110001	1	31	49	00110001
52	30	48	00110000	0	30	48	00110000
53	37	55	00110111	0	36	54	00110110
54	2E	46	00101110	0	2E	46	00101110
55	31	49	00110001	1	31	49	00110001
56	30	48	00110000	1	31	49	00110001

Proses penyisipan data teks biner menghasilkan nilai desimal atau heksadesimal dari sampel audio yang berubah. Hal ini mengakibatkan perubahan 1 nilai. Nilai keseluruhan untuk sampel audio yang disisipkan dalam data teks biner menggunakan metode LSB ditunjukkan pada tabel berikut.:

Tabel 4.5 Nilai Audio Stegano Sample

No	Nilai Audio Stegano		
	Hexa	Des	Biner
1	00	0	00000000
2	01	1	00000001
3	00	0	00000000
4	19	25	00011001
5	66	102	01100110
6	75	117	01110101

Tabel Lanjutan 4.5 Nilai Audio Stegano Sample

No	Nilai Audio Stegano		
	Hexa	Des	Biner
7	78	120	01111000
8	71	113	01110001
9	4C	76	01001100
10	35	53	00110101
11	40	64	01000000
12	20	32	00100000
13	00	0	00000000
14	00	0	00000000
15	02	2	00000010
16	01	1	00000001
17	68	104	01101000
18	73	115	01110011
19	6E	110	01101110
20	6C	108	01101100
21	69	105	01101001
22	72	114	01110010
23	6E	110	01101110
24	33	51	00110011
25	00	0	00000000
26	01	1	00000001
27	00	0	00000000
28	08	8	00001000
29	67	103	01100111
30	73	115	01110011
31	65	101	01100101
32	64	100	01100100
33	00	0	00000000
34	1F	31	00011111
35	B4	180	10110100

Tabel Lanjutan 4.5 Nilai Audio Stegano Sample

No	Nilai Audio Stegano		
	Hexa	Des	Biner
36	A3	163	10100011
37	6C	108	01101100
38	65	101	01100101
39	60	96	01100000
40	75	117	01110101
41	DE	222	11011110
42	05	5	00000101
43	00	0	00000000
44	4C	76	01001100
45	61	97	01100001
46	77	119	01110111
47	62	98	01100010
48	34	52	00110100
49	36	54	00110110
50	2E	46	00101110
51	31	49	00110001
52	30	48	00110000
53	36	54	00110110
54	2E	46	00101110
55	31	49	00110001
56	31	49	00110001

2. Proses Ekstraksi LSB

   Setelah dilakukanya penyisipan, maka untuk mengambil data hasil penyisipan pada objek audio m4a memerlukan teknik ekstraksi. Proses ekstraksi fungsinya untuk mengambil kembali teks yang telah disembunyikan atau dimasukkan ke dalam audio m4a. Tahapan proses ekstraksi berlawanan dengan tahapan proses penyisipan, dan kunci yang digunakan dalam proses penyisipan adalah kunci yang sama.. Adapun cara mendapatkan nilai kunci untuk penanda awal dilakukan dengan cara yang sama saat proses penyisipan, sehingga kunci

penanda awal adalah bit "**01010101".** Pada tahap pengembalian bit-bit yang telah disisipkan, bit-bit yang akan digunakan adalah bit-bit akhir dari audio m4a stegano. Ketika mendapatkan bit dari tag akhir # dalam biner **00100011**, proses pengambilan bit akan berhenti..

Adapun proses ekstraski data teks didalam audio m4a stegano dapat dilihat pada tabel dibawah ini :

Tabel 4.6 Proses Ekstraksi Data Teks

No	Nilai Audio Stegano			Bit Data Teks	Keterangan
	Hexa	Des	Biner		
1	00	0	00000000	0	**01010101** Biner sesuai dengan tanda biner awal, maka proses pengambilan bit berlanjut
2	01	1	00000001	1	
3	00	0	00000000	0	
4	19	25	00011001	1	
5	66	102	01100110	0	
6	75	117	01110101	1	
7	78	120	01111000	0	
8	71	113	01110001	1	
9	4C	76	01001100	0	**01000001** Proses pengambilan bit berlanjut karena biner tidak cocok dengan penanda akhir # biner,
10	35	53	00110101	1	
11	40	64	01000000	0	
12	20	32	00100000	0	
13	00	0	00000000	0	
14	00	0	00000000	0	
15	02	2	00000010	0	
16	01	1	00000001	1	
17	68	104	01101000	0	**01001001** Proses pengambilan bit berlanjut karena biner tidak cocok dengan penanda akhir # biner,
18	73	115	01110011	1	
19	6E	110	01101110	0	
20	6C	108	01101100	0	
21	69	105	01101001	1	
22	72	114	01110010	0	
23	6E	110	01101110	0	
24	33	51	00110011	1	

Tabel Lanjutan 4.6 Proses Ekstraksi Data Teks

No	Nilai Audio Stegano			Bit Data Teks	Keterangan
	Hexa	Des	Biner		
25	00	0	00000000	0	**01001110** Proses pengambilan bit berlanjut karena biner tidak cocok dengan penanda akhir # biner,
26	01	1	00000001	1	
27	00	0	00000000	0	
28	08	8	00001000	0	
29	67	103	01100111	1	
30	73	115	01110011	1	
31	65	101	01100101	1	
32	64	100	01100100	0	
33	00	0	00000000	0	**01010101** Proses pengambilan bit berlanjut karena biner tidak cocok dengan penanda akhir # biner,
34	1F	31	00011111	1	
35	B4	180	10110100	0	
36	A3	163	10100011	1	
37	6C	108	01101100	0	
38	65	101	01100101	1	
39	60	96	01100000	0	
40	75	117	01110101	1	
41	DE	222	11011110	0	**01001100** Proses pengambilan bit berlanjut karena biner tidak cocok dengan penanda akhir # biner,
42	05	5	00000101	1	
43	00	0	00000000	0	
44	4C	76	01001100	0	
45	61	97	01100001	1	
46	77	119	01110111	1	
47	62	98	01100010	0	
48	34	52	00110100	0	

Tabel Lanjutan 4.6 Proses Ekstraksi Data Teks

No	Nilai Audio Stegano			Bit Data Teks	Keterangan
	Hexa	Des	Biner		
49	36	54	00110110	0	**00100011** Biner ini cocok dengan penanda akhir #, proses pengambilan bit dihentikan.
50	2E	46	00101110	0	
51	31	49	00110001	1	
52	30	48	00110000	0	
53	36	54	00110110	0	
54	2E	46	00101110	0	
55	31	49	00110001	1	
56	31	49	00110001	1	

Proses ekstraksi biner # sesuai tag awal dan akhir tidak digunakan pada langkah berikutnya. Maka hasil pengembalian bit yang disisipkan adalah sebagai berikut:
01000001 01001001 01001110 01010101 01001100
Bit biner dibagi menjadi 8 bit untuk setiap kelompok dan dikonversi ke format desimal untuk mendapatkan karakter pertama. Maka karena itu, hasilnya adalah:

Tabel 4.7 Karakter Teks Hasil Ekstraski

No.	Biner	Nilai Desimal	Karakter Teks
1	01000001	65	A
2	01001001	73	I
3	01001110	78	N
4	01010101	85	U
5	01001100	76	L

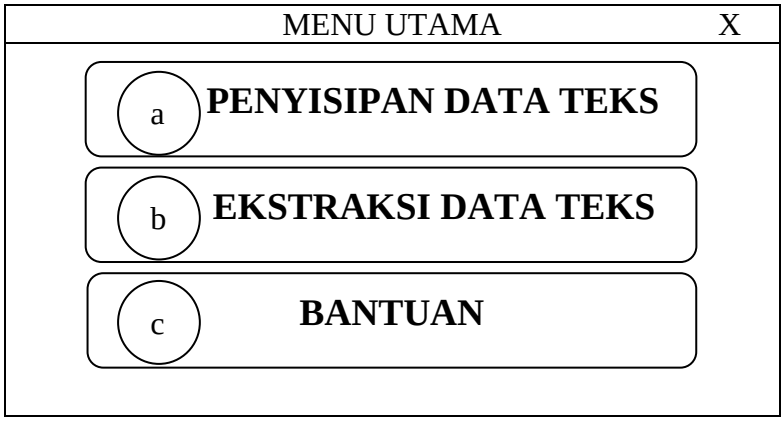
Berdasarkan pada tabel di atas, nilai biner dirubah kembali kedalam bentuk nilai desimal dan mendapakan karakter teks awal sebelum disisipkan.

4.1.4 Perancangan Sistem

Perancangan *interface input output* bermaksud untuk merancang antarmuka aplikasi yang akan dibangun ke dalam suatu perangkat lunak agar lebih mudah dalam membuat aplikasi dan mudah dipahami. Berikut ini adalah bentuk perancangan sistem *input* dan *output* yang nantinya akan diimplementasikan ke dalam sebuah aplikasi.

1. Rancangan menu utama

   Menu utama adalah form yang muncul pertama kali ketika memulai aplikasi. Tampilan desain form menu utama adalah:



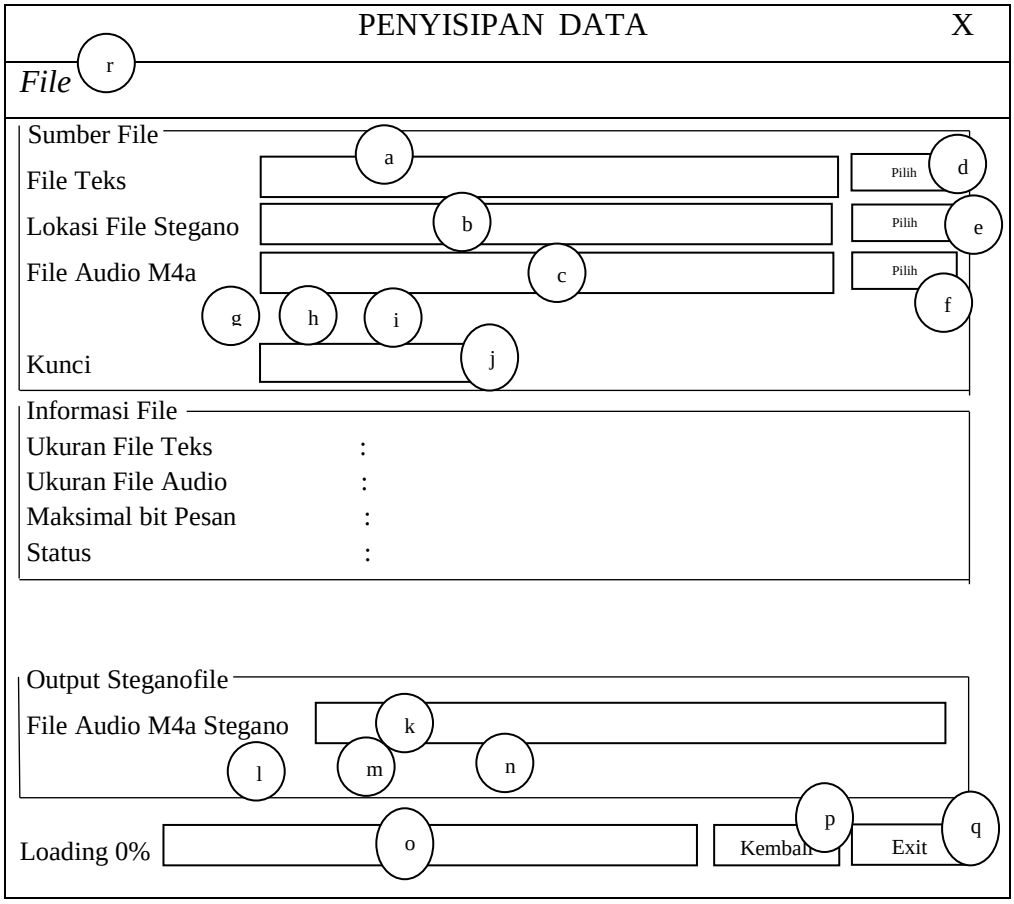
Gambar 4.3 Rancangan Menu Utama

Sesuai gambar 4.3 di atas, adapun keterangannya adalah sebagai berikut ::

a. Menu yang menampilkan *form* untuk melakukan proses penyisipan data teks kedalam objek *audio* m4a.
b. Menu yang menampilkan *form* untuk melakukan proses ekstraksi data teks didalam objek *audio* m4a.
c. Menu yang menampilkan *form* informasi tentang penulis dan tentang aplikasi

2. Rancangan *form* penyisipan data

   Form penyisipan data bertindak sebagai antarmuka bagi pengguna aplikasi saat memasukkan data teks ke objek audio m4a. *Form* penyisipan data memungkinkan pengguna untuk memilih *file* teks untuk dimasukkan ke dalam objek audio m4a yang dipilih. Desain antarmuka untuk *form* penyisipan data ditunjukkan pada gambar berikut.:



Gambar 4.4 Rancangan *Form* Penyisipan Data Teks

Sesuai gambar 4.4 rancangan *form* penyisipan data di atas, adapun keterangannya adalah :

a. *TextBox* untuk menampilkan lokasi informasi sumber *file* teks
b. *TextBox* untuk mencari lokasi penyimpanan *file* audio stegano setelah diproses.
c. *TextBox* untuk menampilkan lokasi informasi sumber *file* audio.
d. *Button* untuk melakukan proses pemilihan *file* teks.
e. *Button* untuk melakukan proses pemilihan lokasi *file* audio setelah diproses.
f. *Button* untuk melakukan proses pemilihan *file* audio.
g. *Button* untuk *play* musik audio.
h. *Button* untuk *pause* musik audio.
i. *Button* untuk *stop* musik audio.
j. *Textbox* untuk menampung karakter kunci.

k. *TextBox* untuk menampilkan lokasi informasi sumber *file* audio stegano.

l. *Button* untuk *play* musik audio stegano.

m. *Button* untuk *pause* musik audio stegano.

n. *Button* untuk *stop* musik audio stegano.

o. *Progressbar* untuk menampilkan informasi proses penyisipan

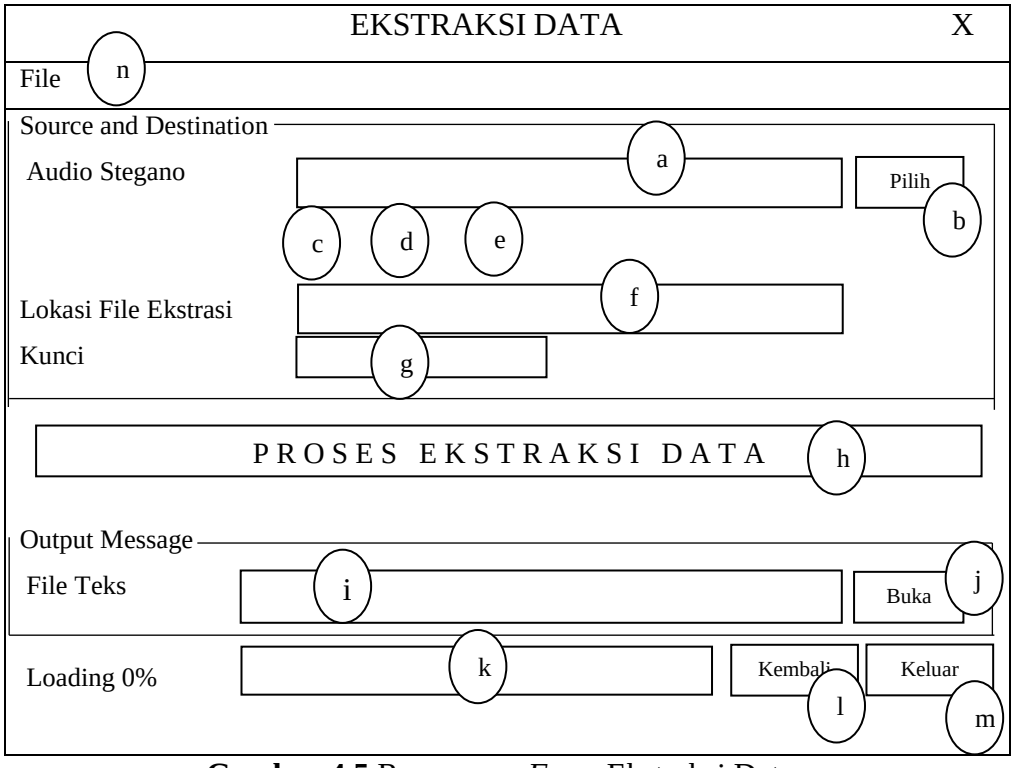
p. *Button* untuk kembali pada menu utama aplikasi.

q. *Button* untuk keluar dari aplikasi.

r. *Button* untuk keluar dari aplikasi.

3. Rancangan *form* ekstrasi data

Form ekstraski data merupakan *interface* yang akan muncul bagi pengguna untuk melakukan proses ekstraksi *file audio* m4a stego yang untuk mendapatkan kembali data teks. Gambar berikut ini menunjukkan desain antarmuka untuk *form* ekstraksi data.



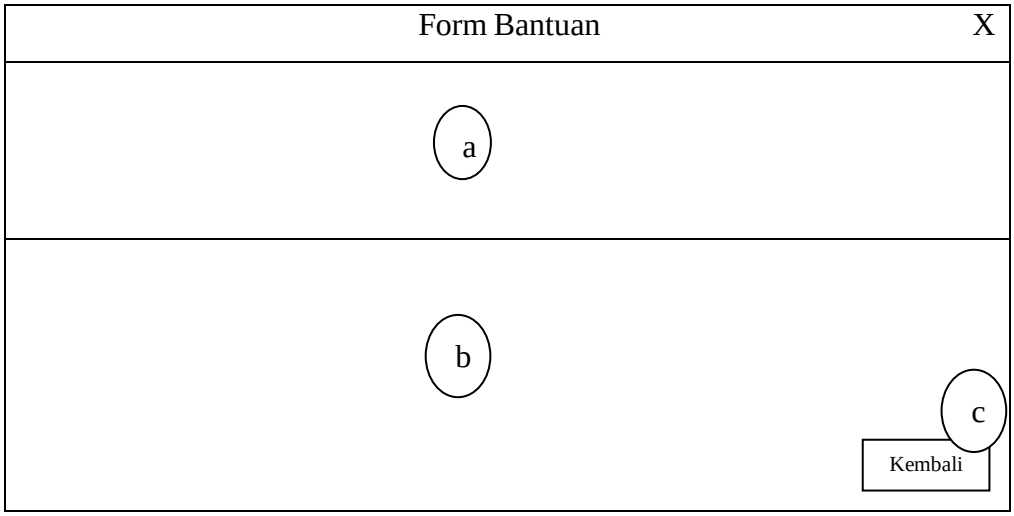
Gambar 4.5 Rancangan *Form* Ekstraksi Data

Berdasarkan gambar 4.5 rancangan *form* Ekstraksi data di atas, adapun keterangannya adalah :

a. *TextBox* untuk menampilkan informasi sumber *file audio* stegano.
b. *Button* untuk melakukan proses pencarian *file* audio stegano.
c. *Button* untuk *play* musik audio stegano.
d. *Button* untuk *pause* musik audio stegano.
e. *Button* untuk *stop* musik audio stegano.
f. *TextBox* untuk mencari lokasi penyimpnan *file* teks hasil ekstraksi
g. *TextBox* untuk menampung kunci ekstraksi.
h. *Button* untuk memproses ekstraksi.
i. *TextBox* untuk menampilkan informasi lokasi *file* teks hasil ekstraksi.
j. *Button* untuk membuka *file* teks hasil ekstraksi.
k. *ProgressBar* untuk menampilkan informasi proses ekstraksi.
l. *Button* untuk kembali pada *form* menu utama.
m. *Button* untuk menutup *form* ekstraksi data.
n. *Button* untuk menutup *form* ekstraksi data.

3. Rancangan *form* Bantuan

Form bantuan merupakan *interface* yang akan muncul bagi pengguna untuk melihat informasi aplikasi dan penulis. Adapun rancangan *interface form* bantuan dapat dilihat pada gambar berikut :



Gambar 4.6 Rancangan *Form* bantuan

Berdasarkan gambar 4.6 rancangan *form* bantuan di atas, adapun keterangannya adalah :

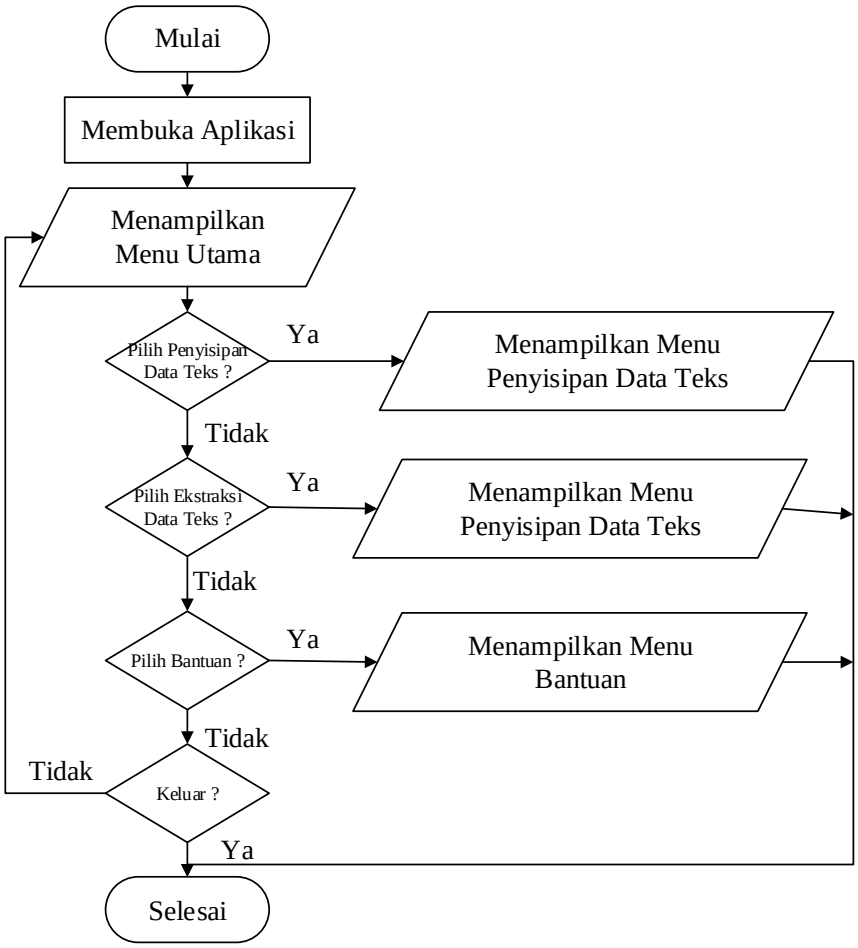
a. *RichTextBox* untuk menampilkan informasi aplikasi.
b. *RichTextBox* untuk menampilkan informasi tentang penulis.
c. *Button* untuk keluar dari *form* bantuan aplikasi.

4.1.4.1 Flowchart Sistem

Flwochart sistem berfungsi untuk menunjukan alur proses dari sistem yang akan dibangun. Adapun *flowchart* sistem dibagi menjadi tiga bagian, yaitu *flowchart* menu utama, *flowchart* penyisipan data teks, *flowchart* ekstraksi data teks dan *flowchart* bantuan. Berikut adalah keseluruhan dari *flowchart* sistem yang akan dibangun:

1. *Flowchart* Menu Utama

   Flowchart menu utama adalah diagram skematis dari alur saat pengguna berada di menu utama. Berikut ini adalah desain *flowchart* menu utama:

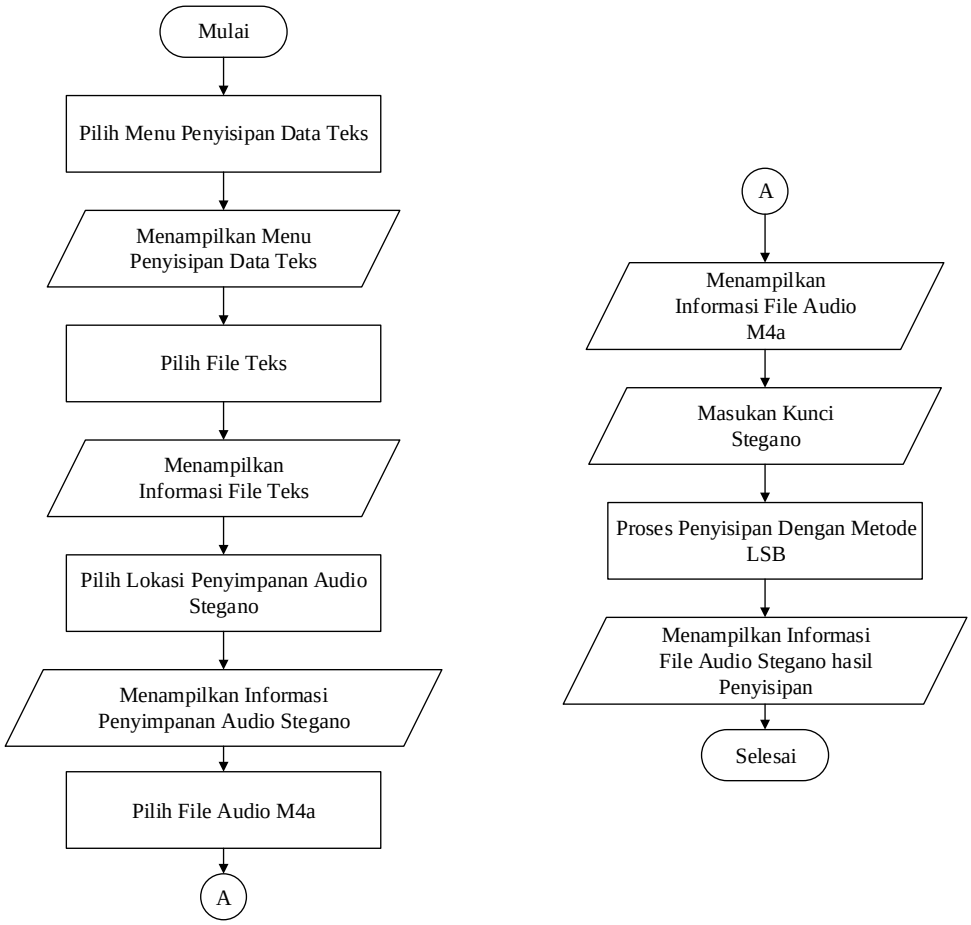


Gambar 4.7 *Flowchart* Menu Utama

Berdasarkan pada gambar *flowchart* menu utama, dapat dijelaskan bahwa langkah pengguna yang membuka aplikasi dihadapkan dengan 3 menu yaitu menu penyisipan data teks yang digunakan untuk memproses penyisipan data teks kedalam audio m4a, menu ekstraksi data teks yaitu menu yang digunakan untuk proses penarikan data teks dari audio m4a serta menu bantuan yang digunakan untuk menampilkan informasi aplikasi dan tentang penulis skripsi ini.

2. *Flowchart* Penyisipan Data Teks

   Flowchart penyisipan data teks adalah gambar alur proses ketika pengguna melakukan penyisipan data teks kedalam audio m4a. Berikut adalah rancangan dari *flowchart* penyisipan data teks:



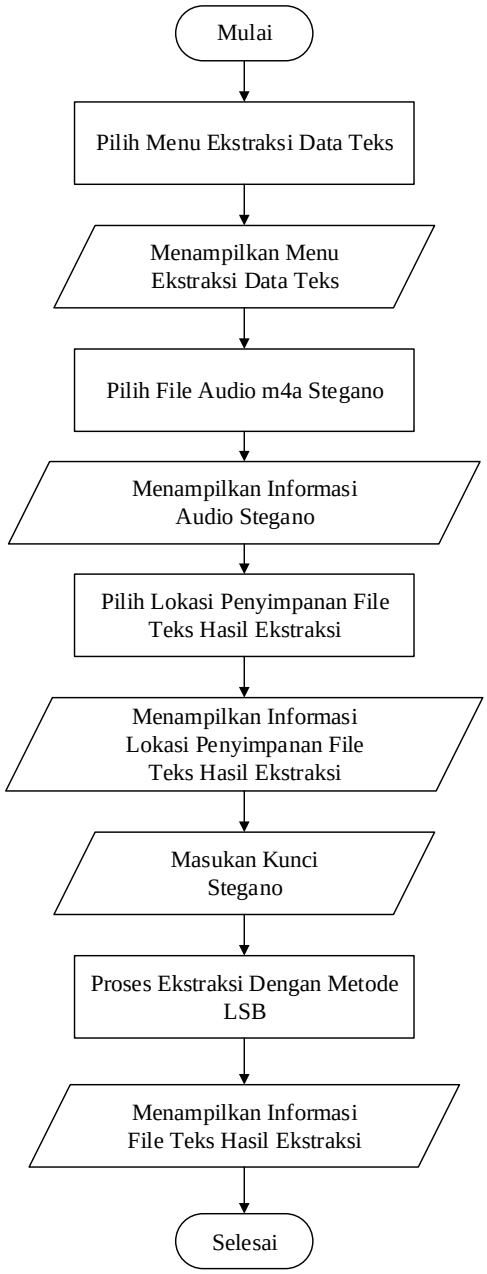
Gambar 4.8 *Flowchart* Penyisipan Data Teks

Berdasarkan pada gambar *flowchart* penyisipan data teks, dapat dijelaskan bahwa langkah awal pengguna melakukan pemilihan data teks yang akan disisipan, kemudian memilih lokasi penyimpanan audio m4a hasil penyisipan selanjutnya

memilih audio m4a yang akan menjadi objek penyisipan serta proses penyisipan dengan metode LSB, sehingga menghasilkan audio m4a stegano.

3. *Flowchart* Ektraksi Data Teks

   Flowchart ekstraksi data teks adalah gambar alur proses ketika pengguna melakukan ekstraksi data teks dari audio m4a stegano. Berikut adalah rancangan dari *flowchart* ekstraksi data teks:

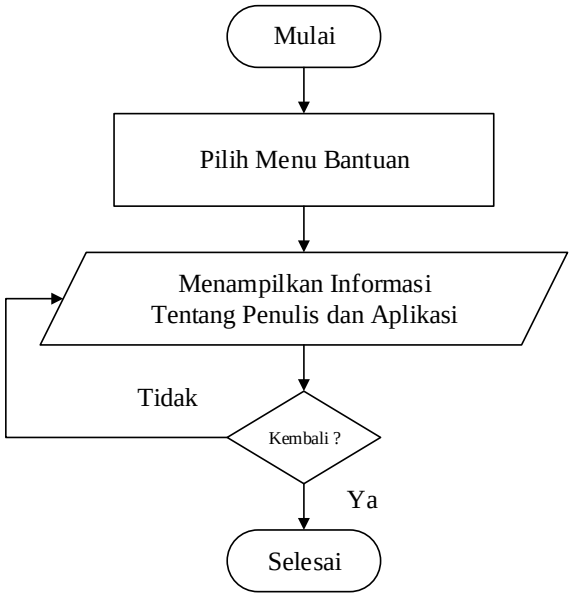


Gambar 4.9 *Flowchart* Ekstraksi Data Teks

Berdasarkan pada gambar *flowchart* ekstraksi data teks, dapat dijelaskan bahwa langkah awal pengguna melakukan pemilihan data audio stegano yang akan diekstraksi, kemudian memilih lokasi penyimpanan *file* teks hasil ekstraksi selanjutnya proses ekstraksi dengan metode LSB, sehingga menghasilkan *file* teks kembali.

4. Flowchart Bantuan

   Flowchart bantuan adalah gambar alur proses ketika pengguna membuka menu bantuan yang berisi informasi tentang aplikasi dan penulis. Berikut adalah rancangan dari *flowchart* bantuan:



Gambar 4.10 *Flowchart* Bantuan

Berdasarkan pada gambar *flowchart* bantuan, dapat dijelaskan bahwa pengguna disajikan dengan informasi tentang aplikasi dan penulis skripsi.

4.2 Hasil

Mengenai hasil yang didapat pada penelitian ada beberapa tahapan yang akan dibahas yaitu pengujian dan penerapan, yang akan dibahas sebagai berikut.

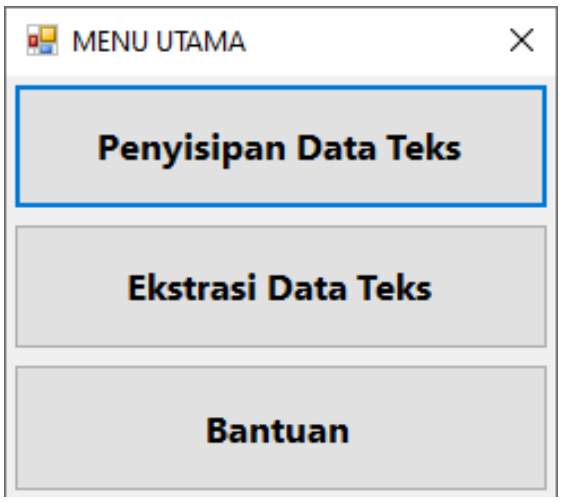
4.2.1 Pengujian Steganografi Pada Aplikasi

Sesuai dengan hasil perancangan aplikasi dan proses perhitungan manual penyisipan data teks pada audio m4amenggunakan Metode Least Significant Bit

maka diimplementasikan sebuah aplikasi perangkat lunak. Proses penerapan steganografi pada aplikasi ini terdiri dari proses penyisipan yaitu memasukkan pesan ke dalam objek audio m4a dan proses ekstraksi untuk mengeluarkan kembali pesan dari objek audio m4a. Proses dalam aplikasi adalah sebagai berikut:

1. Proses Penyisipan Data Teks

   Sebelum melakukan proses penyisipan, program aplikasi akan menampilkan menu utama seperti pada gambar di bawah ini ketika program aplikasi pertama kali dibuka maka

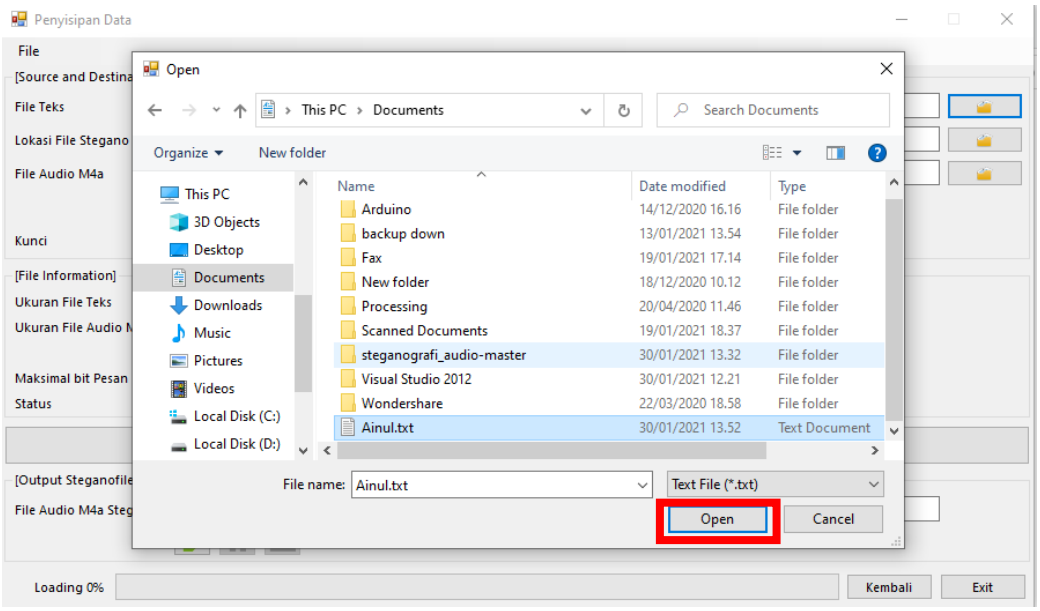


Gambar 4.11 Tampilan Menu Utama

Sesuai menu utama pada Gambar 4.11, terdapat beberapa menu dengan fungsinya masing-masing. Pada aplikasi ini menu “Penyisipan Data Teks” digunakan untuk menyembunyikan data teks dengan teknik steganografi, dan menu “Ekstraksi Data Teks” digunakan untuk memulihkan data teks dari *file* yang disembunyikan. Menu "Bantuan" adalah daftar informasi aplikasi dan informasi tentang penulis. Untuk menjalankan proses penyisipan, pengguna memilih menu “Penyisipan Data Teks” dan keluar tampilan gambar seperti berikut ini:

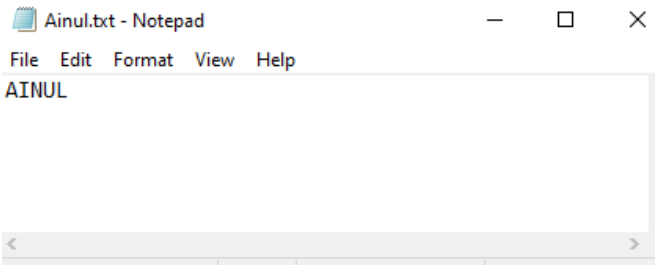
Gambar 4.12 Menu Penyisipan

Berdasarkan gambar 4.12, proses pertama adalah memilih *file* data teks dengan menekan *button* pencarian *file* teks dengan logo berkas pada urutan yang disediakan, akan muncul *pop up* pemilihan gambar seperti di bawah ini:



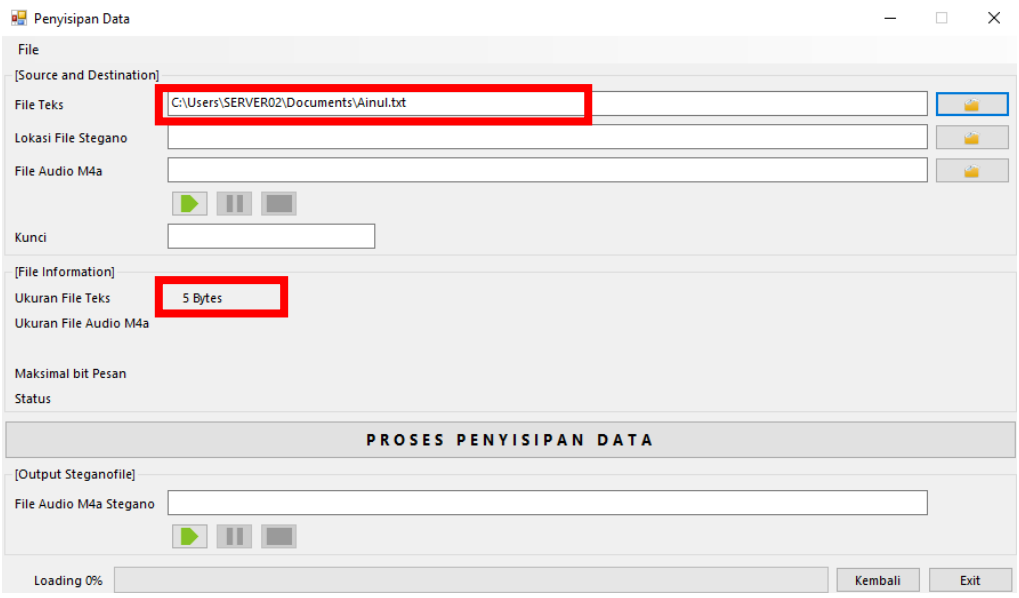
Gambar 4.13 *Pop Up* Pilih Teks Pada Menu Penyisipan

Berdasarkan gambar di atas, *file* data teks yang digunakan bernama Ainul.txt dengan isian karakter “AINUL” seperti gambar bawah ini:



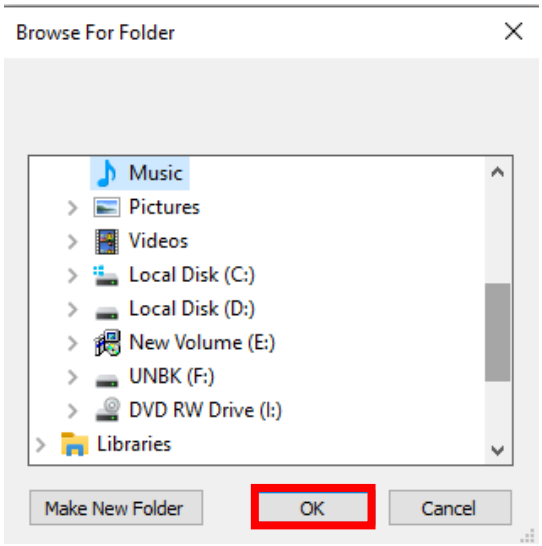
Gambar 4.14 Isi *File* Teks

Berdasarkan pada gambar di atas, setelah *file* data teks, kemudian menekan *button* "*Open*", sehingga tampil informasi *file* teks seperti pada gambar di bawah ini



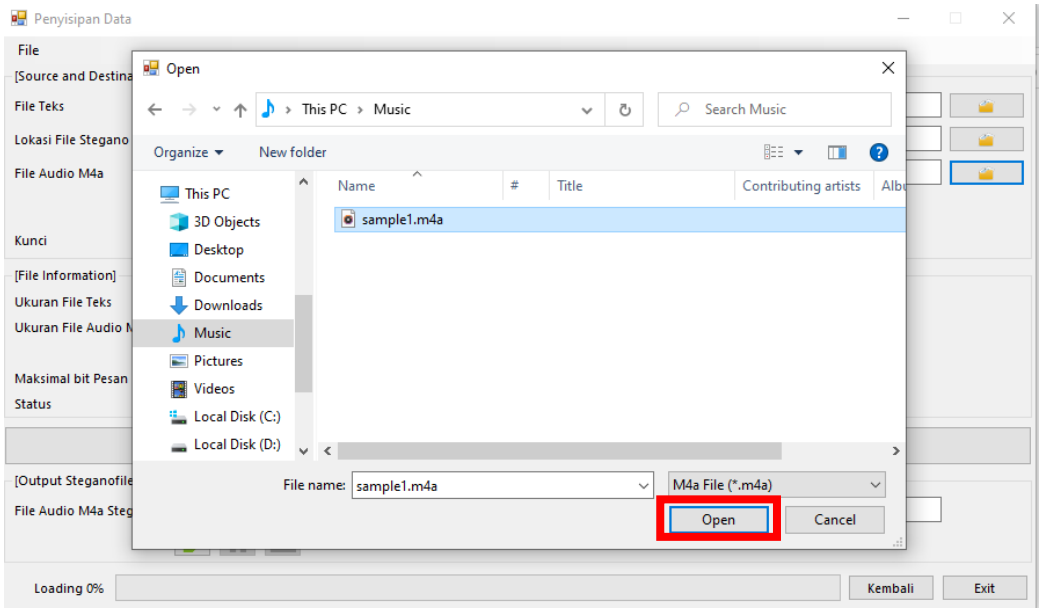
Gambar 4.15 Informasi *File* Teks Pada Menu Penyisipan

Berdasarkan pada gambar di atas, didapati informasi *file* teks seperti lokasi penyimpanan serta ukuran dalam satuan data byte. Adapun selanjutnya adalah memilih lokasi penyimpanan data audio hasil proses dengan menekan button berkas pada lokasi *file* stegano sehingga tampil pop up menu pencarian seperti pada gambar di bawah ini:



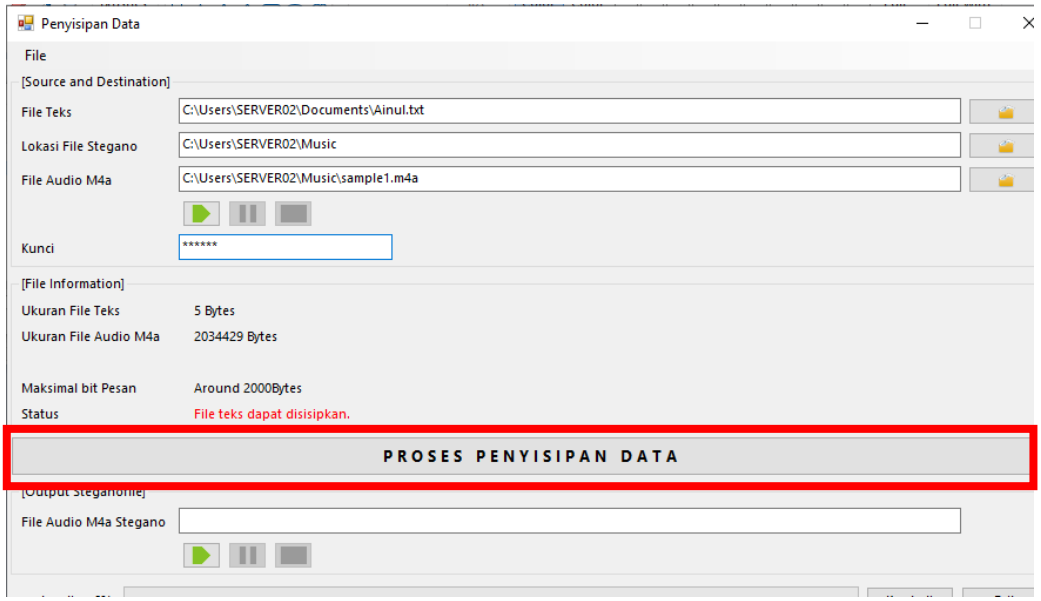
Gambar 4.16 *Pop Up* Pencarian Lokasi Penyimpanan *File* Audio Stegano

Berdasarkan pada gambar di atas, klik ok sehingga tampil informasi lokasi penyimpanan *file* audio. Selanjutnya adalah pemilihan *file* audio m4a yang akan menjadi objek dalam penyisipan *file* teks, berikut proses pencarian *file* audio m4a dalam aplikasi:



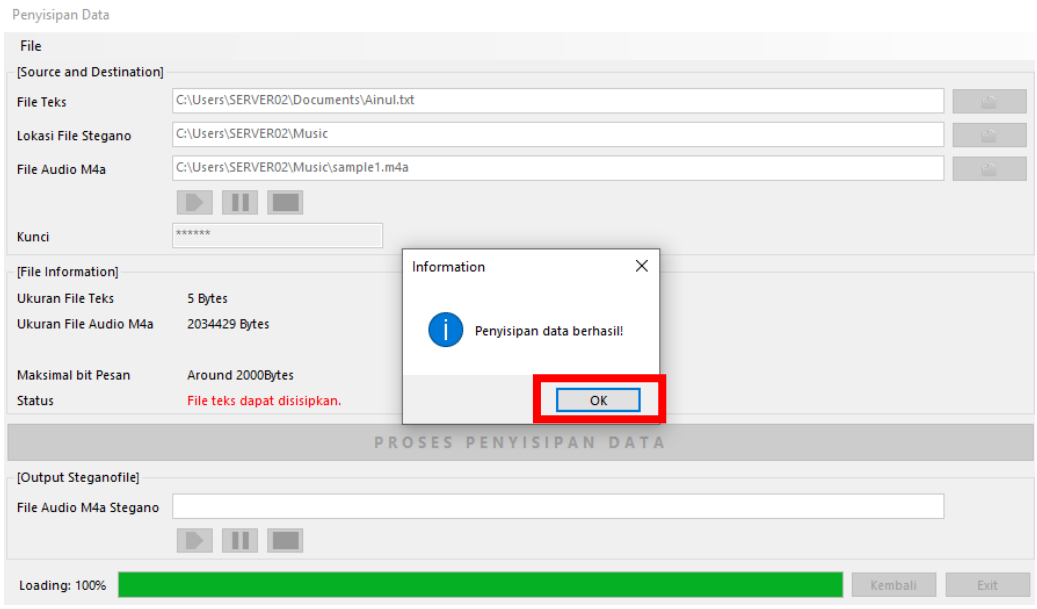
Gambar 4.17 Pemilihan *File* Audio M4a

Berdasarkan pada gambar di atas, *file* audio dengan format m4a dipilih, kemudian *klik button* "*open*" sehingga muncul tampilan seperti gambar berikut:



Gambar 4.18 Proses Pemilihan *File* Audio dan *Input* Kunci

Berdasarkan pada gambar di atas, ditampilkan informasi lokasi *file* audio dan ukuran data *file* audio dalam satuan *byte* serta batas maksimal bit pesan yang dapat disisipkan. Pada sistem aplikasi audio dapat diputar secara langsung untuk menderkan suara dari *file* audio yang di*input*kan. Adapun kunci yang dimasukan adalah kunci yang sama saat proses hitungan manual yaiut "UINSU1", selanjutnya melakukan proses penyisipan pesan dengan menekan *button* "Proses Penyisipan Data" sehingga menampilkan hasil seperti gambar di bawah ini:



Gambar 4.19 Proses Penyisipan Data Berhasil

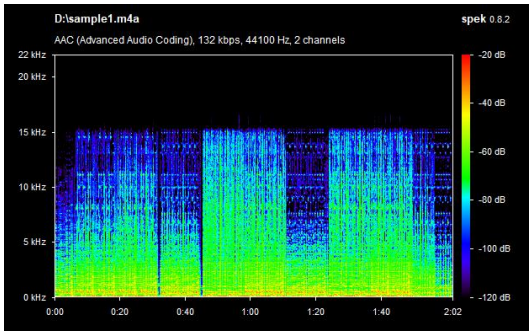
Berdasarkan pada gambar di atas, adapun hasil dari proses penyisipan adalah didapati *file* audio stegano beserta dengan informasi lokasi penyimpanan *file* seperti pada gambar di bawah ini:

Gambar 4.20 Audio Stegano

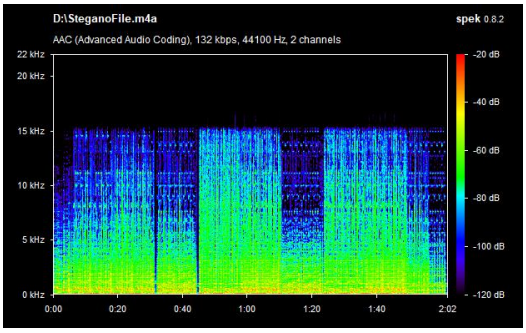
Berdasrkan pada gambar 4.20 di atas, didapati hasil audio stegano dengan format m4a. *File* audio stegano langsung dapat diputar didalam aplikasi sehingga bisa mendengarkan perbedaan sesaat sebelum disisipkan *file* teks. Hasil dari audio stegano disimpan kedalam format .m4a dengan nama *file* audio "Steganofile.m4a". Adapun perbandingan audio m4a sebelum dan sesudah disisipkan dengan *file* teks secara ukuran dapat dilihat pada gambar di bawah ini:

Gambar 4.21 Perbedaan Audio Sebelum dan Sesudah Penyisipan

Berdasarkan pada gambar di atas, sekilas tidak dapat perbedaan audio sample1 sebelum disisipkan *file* teks dengan audio stegano setelah disisipkan *file* teks. Adapun dari segi suara *file* audio juga tidak mengalami perubahan yang signifikan. Hal ini dapat ditandai dengan uji frequensi suara audio menggunakan aplikasi spek sebelum dan sesudah disisipkan teks seperti pada gambar di bawah ini:



Gambar **4.22** Frekuensi Audio Sebelum Disisipkan *File* Teks

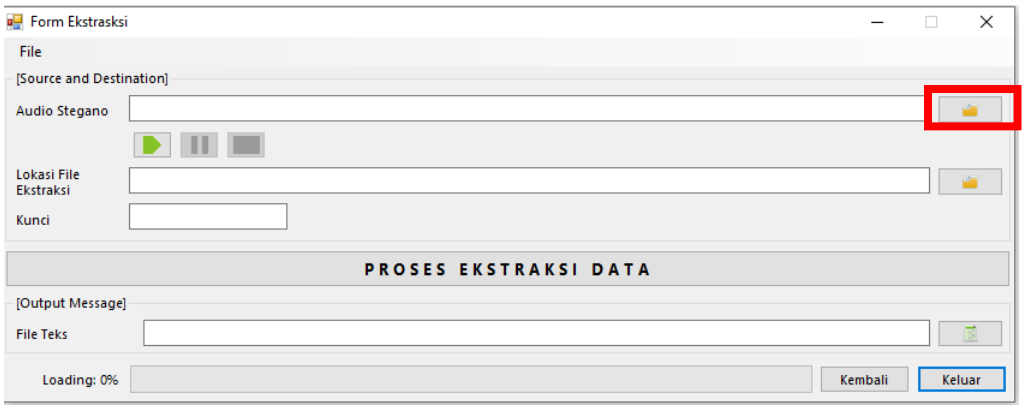


Gambar **4.23** Frekuensi Audio Sesudah Disisipkan *File* Teks

Berdasarkan pada gambar 4.21 dan 4.23 di atas, dapat dipastikan audio stegano tidak mengalami perubahan suara secara signifikan, hal ini dikarenakan tidak adanya data bit yang hilang serta frekuensi yang tetap sama.

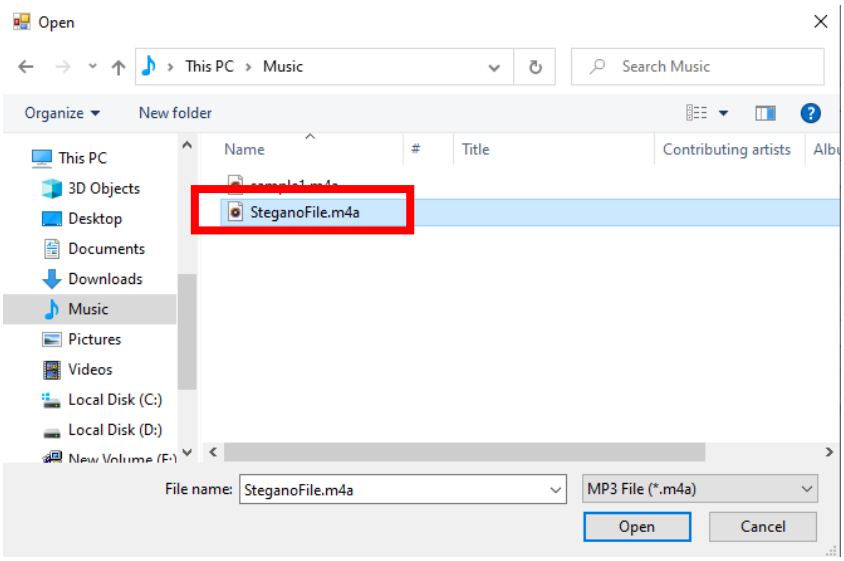
2. Proses Ekstraksi Data Teks

Setelah selesai dilakukannya proses penyisipan data teks kedalam audio m4a, kemudian untuk mngambil kembali *file* teks yang ada di audio stegano maka dilakukan proses ekstraksi dengan memilih menu “Ekstraki Data Teks” di menu utama sehingga keluar menu ekstraksi seperti gambar yang ada di bawah ini:



Gambar 4.24 Tampilan Menu Ekstraksi

Sesuai dengan menu ekstraksi data teks pada Audio stegano, pertama pilih *file* audio stegano yang disisipkan dalam format .m4a dengan cara menekan tombol yang berlogo *file* dan menampilkan menu *pop up*, pilih audio stegano seperti gambar berikut ini:

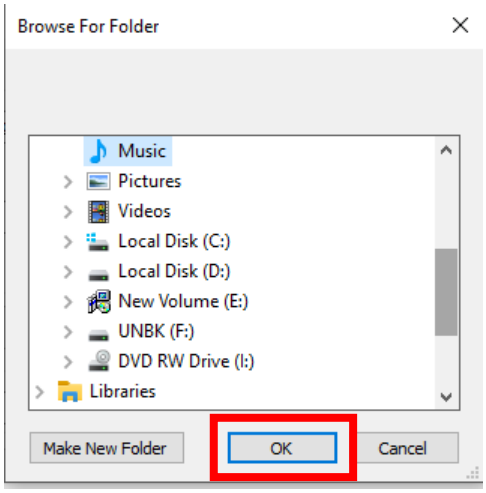


Gambar 4.25 *Pop Up* Audio Stegano Pada Menu Ekstraksi

Berdasarkan Gambar 4.25 di atas, pilih suara stegano dari proses penyisipan sebelumnya dengan nama *file* "Steganofile.m4a" lalu tekan tombol "*Open*" dan tampilkan hasilnya seperti gambar di bawah ini:

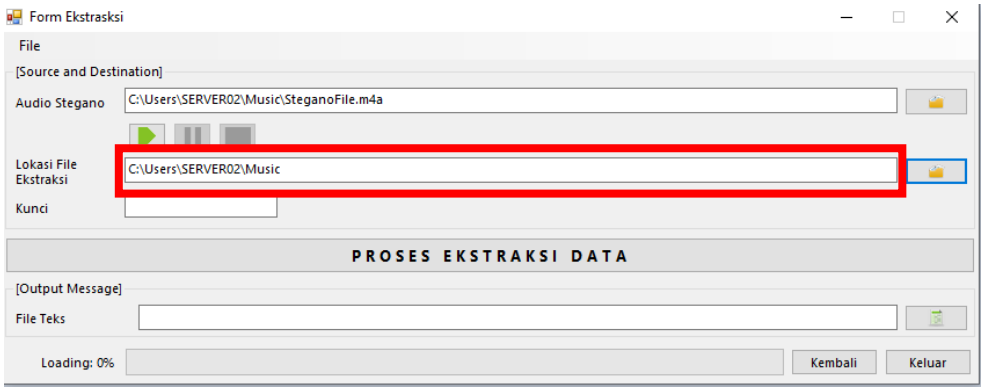
Gambar 4.26 Audio Stegano Pada Menu Ekstraksi

Berdasarkan pada gambar 4.26, didapati informasi lokasi audio stegano. Proses selanjutnya adalah memilih lokasi penyimpnan *file* teks hasil ekstraksi dengan menekan button lokasi *file* ekstrasi sehingga tampilan *pop up* menu seperti pada gambar berikut ini :



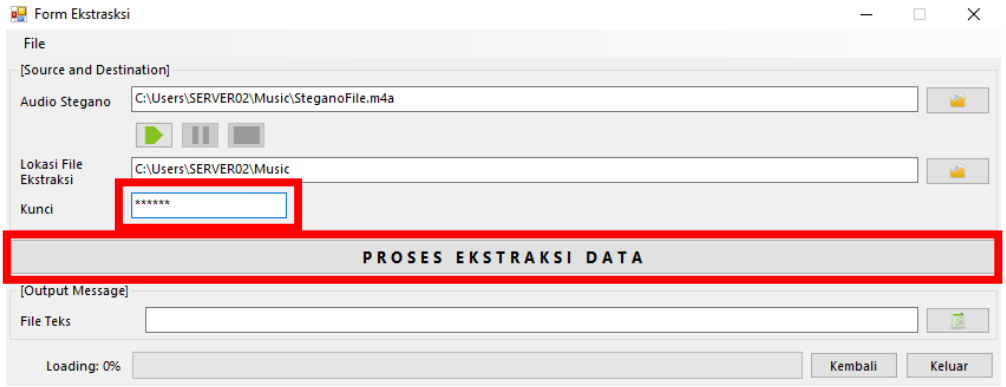
Gambar 4.27 Pemilihan Lokasi Penyimpanan *File* Teks Ekstraksi

Berdasarkan pada gambar di atas, adapun tampilan selanjutnya adalah sebagai berikut :



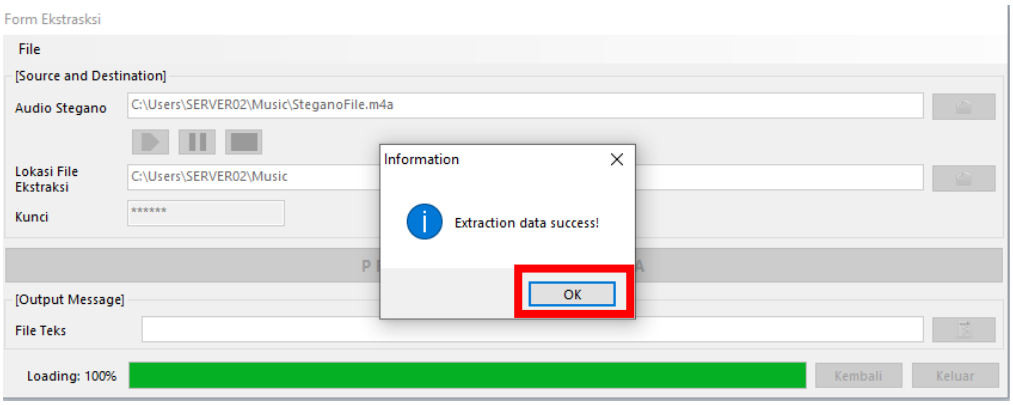
Gambar 4.28 Lokasi Penyimpanan *File* Teks Ditentukan

Berdasarkan pada gambar di atas, selanjutnya adalah memasukan kunci ekstraksi.



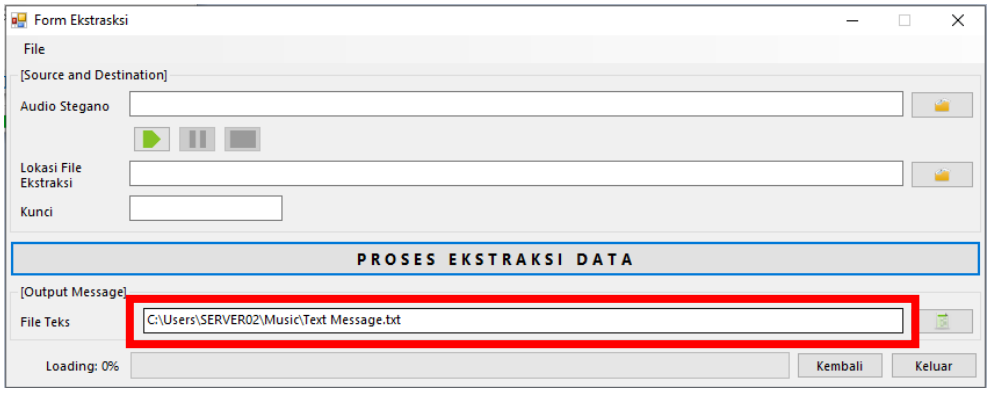
Gambar 4.29 Masukan Kunci Ekstraksi

Kunci ekstraksi adalah kunci yang digunakan ketika proses penyisipan yaitu "UINSU1". Selanjutnya untuk memulai proses ekstraksi *file* teks pada audio dengan LSB *user* menekan *button* "Proses Ekstraksi Data" sehingga tampilan seperti berikut ini:



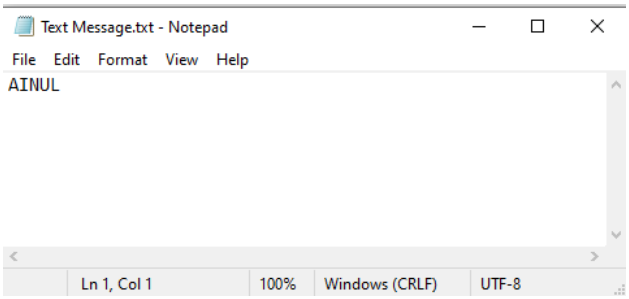
Gambar 4.30 Proses Ekstraksi Berhasil

Berdasarkan pada gambar 4.30 di atas, adapun hasil proses ekstraksi dapat dilihat pada gambar di bawah ini:



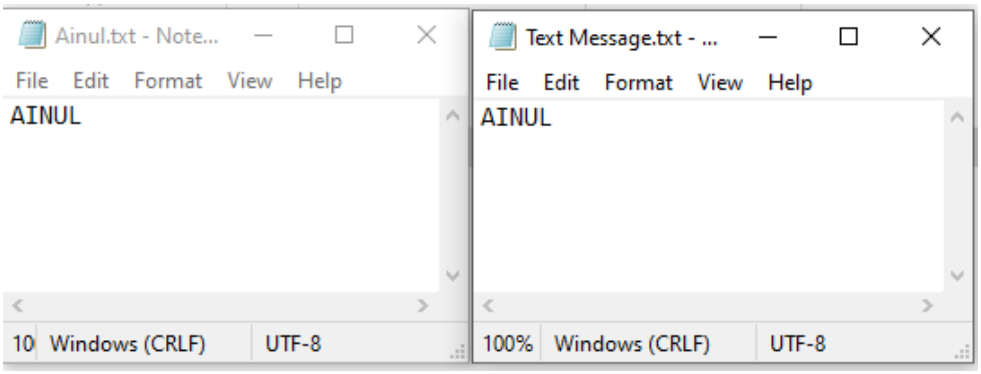
Gambar 4.31 Hasil Proses Ekstraksi

Berdasarkan pada gambar di atas, *file* teks berhasil didapatkan sehingga *file* teks dapat dibuka dengan menekan *button* yang telah disediankan.



Gambar 4.32 Isi *File* Teks Hasil Ekstraksi

Berdasarkan pada gambar di atas, isi pada *file* teks yang telah diekstraksi pada *file* audio dapat dikembalikan seperti semula. Adapun perbandingan hasil data *file* teks sebelum dan sesudah ekstraksi sebagai berikut :

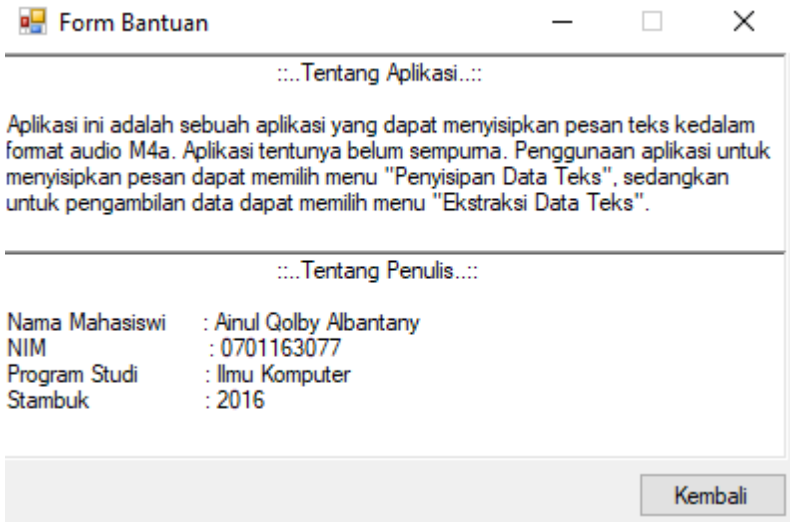


Gambar 4.33 Perbandingan Data Teks

Berdasarkan pada gambar 4.33 di atas, tidak terdapat perbedaan data sebelum dan sesudah disisipkan serta diekstraksi, hal ini menandakan aplikasi berhasil dalam menyisipkan pesan teks kedalam *file* audio m4a menggunakan metode *Least Significant Bit*.

3. Tampilan Menu Bantuan

Adapun tampilan menu bantuan dapat dilihat pada gambar di bawah ini:



Gambar 4.34 Tampilan Menu Bantuan

Berdasarkan gambar di atas, menu bantuan berisikan informasi tentang aplikasi dan penulis. Adapun *button* "Kembali" digunakan untuk kembali ke menu utama.

4. Hasil Pengujian Sistem

Berdasarkan uji steganografi pada aplikasi, ditemukan bahwa suara m4a yang menjadi objek *file* data teks yang disisipkan tidak mengalami perubahan kualitas suara yang signifikan. Hal ini tentunya membuat data *file* teks dalam *file* audio m4a tetap aman, karena tidak ada yang mencurigakan dari suara dalam audio m4a steganografi. Oleh karena itu, kerahasiaan *file* teks yang akan dikirim ke penerima beserta objek audio m4a tidak akan bocor. Hasil pengujian dengan 3 sampel audio dan data teks adalah sebagai berikut:

Tabel 4.8 Hasil Pengujian Penyisipan

No	Nama File		Ukuran Data		Ukuran Teks	Keterangan
	Audio	Audio Stego	Audio	Audio Stego		
1	Sample1.m4a	Stego1.m4a	1.94 MB	1.94 MB	5 Byte	Tidak Berubah
2	Sample2.m4a	Stego2.m4a	3.40 MB	3.40 MB	32 Byte	Tidak Berubah
3	Sample2.m4a	Stego2.m4a	1.66 MB	1.66 MB	150 Byte	Tidak Berubah

Berdasarkan pada tabel 4.8, audio setgano yang disisipkan oleh *file* teks, tidak mengalami perubahan ukuran yang menandakan tidak ada data yang hilang. Adapun hasil pengujian dari proses ekstraksi adalah sebagai berikut :

Tabel 4.9 Hasil Pengujian Ekstraksi

No	Nama File		Ukuran Data		Ukuran Audio	Keterangan
	File Teks	File Teks Ekstraksi	File Teks	File Teks Ekstraksi		
1	Ainul.txt	Text1.txt	5 Byte	5 Byte	1.94 MB	Berhasil Dikembalikan
2	Tes1.txt	Text2.txt	32 Byte	32 Byte	3.40 MB	Berhasil Dikembalikan
3	Tes2.txt	Text3.txt	150 Byte	150 Byte	1.66 MB	Berhasil Dikembalikan

Berdasarkan pada tabel 4.9, data *file* teks hasil ekstraksi kembali seperti data *file* teks awal sebelum disisipkan baik dari segi ukuran dan banyaknya *byte* didalam *file*.

4.2.2 Penerapan

Penerapan pada sistem ini ialah penyembunyian atau penyisipan pesan rahasia berupa teks (.txt) ke dalam wadah penampung berupa audio (.m4a) dimana bukan hanya penyisipan tetapi juga dapat mengambil kembali suatu informasi yang telah disisipkan. Dengan menggunakan sistem aplikasi keamanan data dengan menggunakan metode *least significant bit* bermaanfaat untuk membantu menyembunyikan pesan penting sehingga meminimalisir pencurian digital oleh pihak yang tidak bertanggung jawab.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian penggunaan metode least significant bit untuk melindungi *file* teks tersembunyi sebagai *file* audio berformat m4a, dapat diambil beberapa kesimpulan berdasarkan uraian yang telah diuraikan pada bab-bab sebelumnya. Kesimpulan dari hasil penelitian ini adalah sebagai berikut:

1. *File* teks dalam format .txt dapat dilindungi oleh steganografi, yang disembunyikan di media dalam bentuk audio dalam format .m4a.
2. Proses penyisipan dengan metode *Least Significant Bit* menghasilkan audio steganografi yang tidak berpengaruh banyak terhadap kualitas suara audio, sehingga penyampaian pesan rahasia lebih aman.
3. Aplikasi yang dirancang dengan *Microsoft Visual Studio* 2012 ini didasarkan pada metode *bit* paling tidak signifikan pada setiap tahap proses penyisipan dan ekstraksi, yang dapat menyederhanakan proses perlindungan dan penyembunyian pesan teks ke dalam *file* audio m4a.

5.2 Saran

Adapun saran-saran yang usulkan adalah sebagai berikut:

1. Perancangan aplikasi ini diharapkan dapat memudahkan keamanan pesan *file* teks rahasia dan meminimalkan jumlah kerusakan atau kehilangan pesan *file* teks rahasia.
2. Diharapkan akan ada lebih banyak pengembangan aplikasi yang dirancang untuk operasi keamanan pesan teks ganda.
3. Dapat dikembangkan menjadi aplikasi yang berbasis *web* atau *android*.

DAFTAR PUSTAKA

Al-Qur'an.

Abdul, M., Hi, R., & Files, W. A. V. (2017). *Implementasi Mekanisme Keamanan Data dalam Bentuk Steganografi dengan Metode Least Significant Bit (LSB) Pada File Audio Wav*. Juristek, *5*(2), 202–211.

Aminul, jumiran. F. (2014). *Penyisipan text pada gambar menggunakan steganografi*. Jurnal IPSIKOM, *2*(1), 1–12.

Anti, U. A., Kridalaksana, A. H., & Khairina, D. M. (2017). *Steganografi Pada Video Menggunakan Metode Least Significant Bit (LSB) Dan End Of File (EOF)*. Jurnal Ilmiah Ilmu Komputer, *12*(2), 104. https://doi.org/10.30872/jim.v12i2.658

Assyahid, M. M., Rihartanto, R., & Utomo, D. S. B. (2018). *Implementasi Steganografi Pesan Text ke Dalam Audio Dengan Metode Spread Spectrum*. Juristek, *3*(2), 27–34.

Furqan, M. (2020). *Implementasi Steganografi Menggunan Metode Spread Spectrum dalam Pengamanan Data Teks pada Citra Digital*. Jurnal Sains Komputer & Informatika.

Kuniadi, B., Puspitaningrum, D., & Coastera, F. F. (2017). *Perancangan Dan Pembuatan Aplikasi Steganografi Pesan Teks Pada Audio Digital Dengan Metode Least Significant Bit*. Jurnal Rekursif, *5*(3), 285–297.

Mukhtar, Harun. 2018. *Kriptografi untuk Keamanan Data*. Yogyakarta : Budi Utama.

Narayana, S., & Prasad, G. (2010). *Two New Approaches for Secured Image Steganography Using Cryptographic Techniques and Type Conversions*. Signal & Image Processing : An International Journal, *1*(2), 60–73. https://doi.org/10.5121/sipij.2010.1206

Nasution, Y. R., Furqan. M., & Sinaga. Meri. (2020). *Implementasi Steganografi Menggunakan Metode Spread Spectrum Dalam Pengamanan Data Teks Pada Citra Digital*. Jurnal Sains Komputer & Informatika, *4(2), 351-358.*

Oktaviani.J. (2015). *Dynamic Cell Spreading Technique to Hidden Secret Message and Authentication*. JTRISTE, *2*(2).

Piarsa, I. N., & Dharmadi, I. M. A. (2010). *Implementasi Watermarking Pada Suara Digital Dengan Metode Data Echo Hiding*. Jurnal Teknologi Elektro, *9*(2). https://doi.org/10.24843/10.24843/MITE

Pratama, Firman Kurniawan. 2010. *Analisis perbandingan metode Redundant Pattern Encoding dan Discrete Cosine Transformation Sebagai metode steganografi pada citra digital*. Bandung : Universitas Komputer Indonesia.

Purwasito, Andrik. (2017). Analisis Pesan Message Analysis, *the Messenger*, 9

Sahid. 2006. *Panduan Praktis MATLAB.* Yogyakarta : Andi Offset.

Saragih, R. A., Teknik, J., Universitas, E., & Maranatha, K. (2006). *Metode Parity Coding Versus Metode Spread Spectrum Pada Audio Steganography*. JurnalnTeknologi Informasi, 71–76.

Sayed, L. (2020). *Optimalisasi Steganografi Audio Untuk Pengamanan Informasi*. Jurnal Sains Riset, *10*(1), 45–50.

Syafitri, Suri. (2013). *Analisis Perbandingan Metoe Low Bit Coding Dan Least Significant Bit Untuk Digital Watermarking Pada File Wma*, Jurnal Teknologi Informasi.

Tjolleng, Amir. 2017. *Pengantar Pemrograman MATLAB.* Jakarta : Elex Media Komputindo

Utami, E. (2009). *Pendekatan Metode Least bit Modification untuk Merancang Aplikasi Steganography pada File Audio Digital tidak Terkompresi.* Jurnal DASI, *Vol 10*(1).

LAMPIRAN-LAMPIRAN

Lampiran 1

Hasil pengujian steganografi pada aplikasi "*Text Steganography* Pada Media *Audio* Format M4a Dengan Menggunakan Metode *Least Significant Bit"*:

Hasil pengujian penyisipan

No	Nama File		Ukuran Data		Ukuran Teks	Keterangan
	Audio	Audio Stego	Audio	Audio Stego		
1	Sample1.m4a	Stego1.m4a	1.94 MB	1.94 MB	5 Byte	Tidak Berubah
2	Sample2.m4a	Stego2.m4a	3.40 MB	3.40 MB	32 Byte	Tidak Berubah
3	Sample2.m4a	Stego2.m4a	1.66 MB	1.66 MB	150 Byte	Tidak Berubah

Hasil pengujian ekstraksi

No	Nama File		Ukuran Data		Ukuran Audio	Keterangan
	File Teks	File Teks Ekstraksi	File Teks	File Teks Ekstraksi		
1	Ainul.txt	Text1.txt	5 Byte	5 Byte	1.94 MB	Berhasil Dikembalikan
2	Tes1.txt	Text2.txt	32 Byte	32 Byte	3.40 MB	Berhasil Dikembalikan
3	Tes2.txt	Text3.txt	150 Byte	150 Byte	1.66 MB	Berhasil Dikembalikan

Lampiran 2

Tools Visual Studio yang digunakan untuk "*Text Steganography* Pada Media *Audio* Format M4a Dengan Menggunakan Metode *Least Significant Bit"* :

Tools	Fungsi
FrmMenuUtama	Menampilkan menu utama
FrmEmbedding	Menampilkan manu penyisipan teks
FrmExtraction	Menampilkan form ekstraksi
groupBox1	Menampilkan sumber dan tujuan file
groupBox2	Menampilkan informasi file
groupBox3	Menampilkan output file steganografi
button1	Tombol untuk menampilkan menu penyisipan teks
button2	Tombol untuk menampilkan menu ekstraksi
button3	Tombol untuk menampilkan menu bantuan
btnOpenMessage	Mencari file teks
btnBrowseDestination	Mencari lokasi penyimpanan
btnOpenMedia	Mencari file audio m4a
btnEmbedding	Proses penyisipan
btnPlayM4a	Memutar audio m4a
btnPausedM4a	Menghentikan sementara audio m4a
btnStopM4a	Menghentikan audio m4a
btnPlaySteganо	Memutar stego audio m4a
btnPausedStegano	Menghentikan sementara stego audio m4a
btnStopStegano	Menghentikan stego audio m4a
btnRestart	Kembali ke menu utama
btnExit	keluar
btnExtraction	Proses ektraksi
btnPlaySteganoMP4	Memutar stego audio m4a
btnPausedSteganoMP4	Menghentikan sementara stego audio m4a
btnStopSteganoMP4	Menghentikan stego audio m4a

btnOpenSteganofile	Mencari file stego audio m4a
btnBrowseDestination	Menentukan lokasi output
btnOpenMessage	Membuka file pesan
label1	Menampilan maksimal bit pesan untuk proses penyisipan
label2	Menampilkan file teks untuk proses penyisipan
label3	Menampilkan lokasi file stegano untuk proses penyisipan
label4	Menampilkan file audio m4a untuk proses penyisipan
label5	Menampilkan file audio m4a stegano
label6	Menampilkan ukuran teks
label7	Menampilkan file audio m4a
label9	Menampilkan status
label10	Menampilkan kunci
lblLoading	Menampilkan loading
label1	Menampilkan stego audio untuk proses ekstraksi
label2	Menampilkan lokasi output untuk proses ekstraksi
label3	Menampilkan kunci untuk proses ekstraksi
label4	Menampilkan file output dalam proses ekstraksi
txtMessage	Menampilkan file teks
txtDestination	Menampilkan lokasi file stegano
txtMedia	Menampilkan file audio m4a
txtPassword	Menampilkan kunci
txtStegofilePath	Menampilkan file audio m4a stegano
txtSteganofile	Menampilkan stego audio yang akan diekstraksi
progressBar2	Menampilkan progress loading

Lampiran 3

Source code Visual Studio untuk proses *Text Steganography* Pada Media *Audio* Format M4A Dengan Menggunakan Metode *Least Significant Bit* :

Form Embedding

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using Revisi_Stegano_Audio.Lib;
using System.Diagnostics;
using System.Threading;

namespace Revisi_Stegano_Audio.UI
{
    public partial class FrmEmbedding : Form
    {

        private MediaPlayer mPlayer;
        private ReadSetting rs;
        private SteganoEmbedding stega;
        private FileSizeValidation fileValidation;
        private bool isImage;
        private bool isSuccess;
        private long messageSize;
        private long mediaSize;
        private Thread thread;

        private string cipherData;

        public FrmEmbedding()
        {
            InitializeComponent();
            rs = new ReadSetting();
        }

        #region method-method

        private void exit()
        {
            Application.ExitThread();
        }
```

```
        private void restart()
        {
            Application.Restart();
        }

        private void browseFile(TextBox tb)
        {
            openFileDialog1.FileName = string.Empty;
            if (tb == txtMessage)
                openFileDialog1.Filter = "Text File (*.txt)|*.txt";
            else
                openFileDialog1.Filter = "M4a File (*.m4a)|*.m4a";

            if (openFileDialog1.ShowDialog() != DialogResult.OK)
                return;

            tb.Text = openFileDialog1.FileName;
            if (tb.Name == "txtMessage")
            {
                if (openFileDialog1.FilterIndex != 1)
                    isImage = true;
                else
                    isImage = false;
            }
        }

        private void browsePathFolder(TextBox tb)
        {
            if (folderBrowserDialog1.ShowDialog() != DialogResult.OK)
                return;

            tb.Text = folderBrowserDialog1.SelectedPath;
        }

        private void infoFile(TextBox tb, Label lbl)
        {
            if (string.IsNullOrEmpty(tb.Text))
                return;
            FileInfo info = new FileInfo(tb.Text);
            lbl.Text = string.Format(info.Length.ToString() + " {0}",
"Bytes");

            if (tb.Name == "txtMessage")
                messageSize = info.Length;
            else
                mediaSize = info.Length;

            if (string.IsNullOrEmpty(lblMessageInf.Text) ||
string.IsNullOrEmpty(lblMediaInf.Text))
                return;

            fileValidation = new FileSizeValidation(messageSize,
mediaSize);
```

```
            lblStatus.Text = fileValidation.messageValid();
            if (lblStatus.Text.Contains("embedded"))
            {
                lblStatus.ForeColor = Color.Green;

                lblRequirement.Text =
string.Format(fileValidation.requirement() + "{0}", "Bytes");
            }
            else
            {
                lblStatus.ForeColor = Color.Red;
                lblRequirement.Text =
string.Format(fileValidation.requirement() + "{0}", "Bytes");
            }
        }

        private void playMedia(string path, string type)
        {
            if (string.IsNullOrEmpty(path))
                return;
            mPlayer.play(@path);
            if (type == "m4a")
            {
                btnPlayM4a.Enabled = false;
                btnPausedM4a.Enabled = true;
                btnStopM4a.Enabled = true;
            }
            else
            {
                btnPlaySteganо.Enabled = false;
                btnPausedSteganо.Enabled = true;
                btnStopSteganо.Enabled = true;
            }
        }

        private void pauseMedia(string type)
        {
            mPlayer.pause();
            if (type == "m4a")
            {
                btnPlayM4a.Enabled = true;
                btnStopM4a.Enabled = true;
                btnPausedM4a.Enabled = false;
            }
            else
            {
                btnPlaySteganо.Enabled = true;
                btnStopSteganо.Enabled = true;
                btnPausedSteganо.Enabled = false;
            }
        }

        private void stopMedia(string type)
        {
```

```
            mPlayer.stop();
            if (type == "m4a")
            {
                btnPausedM4a.Enabled = false;
                btnStopM4a.Enabled = false;
                btnPlayM4a.Enabled = true;
            }
            else
            {
                btnPausedStegano.Enabled = false;
                btnStopStegano.Enabled = false;
                btnPlaySteganо.Enabled = true;
            }
        }

        private bool validation()
        {
            if (string.IsNullOrEmpty(txtMessage.Text))
            {
                MessageBox.Show("File Teks  belum dipilih!", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }

            if (string.IsNullOrEmpty(txtDestination.Text))
            {
                MessageBox.Show("Lokasi penyimpanan belum dipilih!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }

            if (string.IsNullOrEmpty(txtMedia.Text))
            {
                MessageBox.Show("File Audio m4a  belum dipilih!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }

            if (string.IsNullOrEmpty(txtPassword.Text) ||
txtPassword.Text.Length < 6)
            {
                MessageBox.Show("Ukuran kunci minimal 6 karakter!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }

            return true;
        }

        private bool encrypt()
        {
            cipherData = string.Empty;
            if (isImage == true)
                cipherData = save.encrypt(
```

```
Convert.ToBase64String(DocCommonFunction.readDocumentBytes(txtMessage.Te
xt)),
                        txtPassword.Text,
                        rs.keySize(),
                        rs.keySalt(),
                        rs.padding(),
                        rs.cipher());
            else
                cipherData = save.encrypt(

DocCommonFunction.readDocumentString(txtMessage.Text),
                        txtPassword.Text,
                        rs.keySize(),
                        rs.keySalt(),
                        rs.padding(),
                        rs.cipher());

            if (string.IsNullOrEmpty(cipherData))
                return false;
            return true;
        }

        private void embedding()
        {
            stega = new SteganoEmbedding();
            stega.fileMedia = txtMedia.Text;
            stega.data = cipherData;
            if (stega.embeddingData())
            {

DocCommonFunction.createDocumentBytes(@txtDestination.Text +
@"\SteganoFile.m4a", stega.stegofilem4a);
                MessageBox.Show("Penyisipan data berhasil!",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                isSuccess = true;
            }
            else
            {
                MessageBox.Show("Penyisipan data gagal!", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                isSuccess = false;
            }
        }

       private void openFile(string path)
        {
            if(string.IsNullOrEmpty(path))
                return;
            Process.Start(@path);
        }

        private void reset()
        {
```

```
            txtMessage.Text = string.Empty;
            txtMedia.Text = string.Empty;
            txtDestination.Text = string.Empty;
            txtPassword.Text = string.Empty;

            lblMessageInf.Text = string.Empty;
            lblMediaInf.Text = string.Empty;

                        lblStatus.Text = string.Empty;
            lblRequirement.Text = string.Empty;

            lblLoading.Text = "Loading: 0%";
            progressBar2.Value = 0;
        }

        private void initMPlayer(string path)
        {
            if (string.IsNullOrEmpty(@path))
                return;
            mPlayer = new MediaPlayer();
        }

        private void enableComponents(bool bol)
        {
            fileToolStripMenuItem.Enabled = bol;
            settingToolStripMenuItem.Enabled = bol;
            txtMessage.Enabled = bol;
            txtDestination.Enabled = bol;
            txtMedia.Enabled = bol;
            txtPassword.Enabled = bol;
            txtStegofilePath.Enabled = bol;
            btnOpenMessage.Enabled = bol;
            btnBrowseDestination.Enabled = bol;
            btnOpenMedia.Enabled = bol;
            btnPlayM4a.Enabled = bol;
            btnPlayStegano.Enabled = bol;
            btnEmbedding.Enabled = bol;
            btnRestart.Enabled = bol;
            btnExit.Enabled = bol;
            ControlBox = bol;
        }

        #endregion

        private void FrmEmbedding_FormClosed(object sender,
FormClosedEventArgs e)
        {
            exit();
        }

        private void btnOpenMessage_Click(object sender, EventArgs e)
        {
            browseFile(txtMessage);
            infoFile(txtMessage, lblMessageInf);
```

```
        }

        private void btnBrowseDestination_Click(object sender, EventArgs
e)
        {
            browsePathFolder(txtDestination);
        }

        private void btnOpenMedia_Click(object sender, EventArgs e)
        {
            browseFile(txtMedia);
            infoFile(txtMedia, lblMediaInf);
        }

        private void btnPlayM4a_Click(object sender, EventArgs e)
        {
            playMedia(txtMedia.Text, "m4a");
        }

        private void btnPausedM4a_Click(object sender, EventArgs e)
        {
            pauseMedia("m4a");
        }

        private void btnStopM4a_Click(object sender, EventArgs e)
        {
            stopMedia("m4a");
        }

        private void btnEmbedding_Click(object sender, EventArgs e)
        {
            if (!validation())
                return;
            if (!encrypt())
                return;

            thread = new Thread(new ThreadStart(embedding));
            thread.IsBackground = true;
            thread.Start();
            timer1.Enabled = true;
        }
        private void btnOpenFile_Click(object sender, EventArgs e)
        {
            openFile(txtStegofilePath.Text);
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            exit();
        }

        private void btnRestart_Click(object sender, EventArgs e)
        {
            restart();
```

```
        }

        private void btnPlayStegano_Click(object sender, EventArgs e)
        {
            playMedia(txtStegofilePath.Text, "stegano");
        }

        private void btnPausedStegano_Click(object sender, EventArgs e)
        {
            pauseMedia("stegano");
        }

        private void btnStopStegano_Click(object sender, EventArgs e)
        {
            stopMedia("stegano");
        }

        private void txtMedia_TextChanged(object sender, EventArgs e)
        {
            initMPlayer(txtMedia.Text);
        }

        private void txtStegofilePath_TextChanged(object sender,
EventArgs e)
        {
            initMPlayer(txtStegofilePath.Text);
        }

        private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            exit();
        }

        private void txtMessage_DoubleClick(object sender, EventArgs e)
        {
            btnOpenMessage_Click(sender, e);
        }

     private void txtDestination_DoubleClick(object sender, EventArgs e)
        {
            btnBrowseDestination_Click(sender, e);
        }

        private void txtMedia_DoubleClick(object sender, EventArgs e)
        {
            btnOpenMedia_Click(sender, e);
        }

        private void txtStegofilePath_DoubleClick(object sender,
EventArgs e)
        {
            btnOpenFile_Click(sender, e);
        }
```

```
        private void timer1_Tick(object sender, EventArgs e)
        {
            if (thread.IsAlive)
            {
                enableComponents(false);
                lblLoading.Text = "Loading: " +
stega.persenLoading.ToString() + "%";
                progressBar2.Value =
Convert.ToInt32(stega.persenLoading);
            }
            else
            {
                timer1.Enabled = false;
                enableComponents(true);
                if (isSuccess)
                    txtStegofilePath.Text = @txtDestination.Text +
@"\SteganoFile.m4a";
                else
                    txtStegofilePath.Text = string.Empty;
                reset();
            }
        }

        private void exitToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            exit();
        }

        private void FrmEmbedding_Load(object sender, EventArgs e)
        {

        }

        private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

        }

     private void txtDestination_TextChanged(object sender, EventArgs e)
        {
        }
    }
}
```

Form Extraction

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Revisi_Stegano_Audio.Lib;

using System.Diagnostics;
using System.Threading;

namespace Revisi_Stegano_Audio.UI
{
    public partial class FrmExtraction : Form
    {
        private MediaPlayer mPlayer;
        private ReadSetting rs;
        private SteganoExtraction stega;

        private bool istxt;
        private bool isSuccess;
        private string plainText;
        private Thread thread;

        public FrmExtraction()
        {
            InitializeComponent();
            rs = new ReadSetting();
        }

        #region method-method

        private void exit()
        {
            Application.ExitThread();
        }

        private void restart()
        {
            Application.Restart();
        }

        private void browseFile(TextBox tb)
        {
            openFileDialog1.FileName = string.Empty;
            if (tb == txtMessage)
                openFileDialog1.Filter = "Text File (*.txt)|*.txt";
            else
                openFileDialog1.Filter = "MP3 File (*.m4a)|*.m4a";
```

```
            if (openFileDialog1.ShowDialog() != DialogResult.OK)
                return;

            tb.Text = openFileDialog1.FileName;
        }

        private void browsePathFolder(TextBox tb)
        {
            if (folderBrowserDialog1.ShowDialog() != DialogResult.OK)
                return;

            tb.Text = folderBrowserDialog1.SelectedPath;
        }

        private void initMPlayer(string path)
        {
            if (string.IsNullOrEmpty(@path))
                return;
            mPlayer = new MediaPlayer();
        }

        private void playMedia(string path)
        {
            if (string.IsNullOrEmpty(path))
                return;
            mPlayer.play(@path);
            btnPlaySteganoMP3.Enabled = false;
            btnPausedSteganoMP3.Enabled = true;
            btnStopSteganoMP3.Enabled = true;
        }

        private void pauseMedia()
        {
            mPlayer.pause();
            btnPlaySteganoMP3.Enabled = true;
            btnStopSteganoMP3.Enabled = true;
            btnPausedSteganoMP3.Enabled = false;
        }

        private void stopMedia()
        {
            mPlayer.stop();
            btnPausedSteganoMP3.Enabled = false;
            btnStopSteganoMP3.Enabled = false;
            btnPlaySteganoMP3.Enabled = true;
        }


        private bool validation()
        {
            if (string.IsNullOrEmpty(txtSteganofile.Text))
            {
                MessageBox.Show("File Stegano belum dipilih!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
```

```
                return false;
            }

            if (string.IsNullOrEmpty(txtDestinaion.Text))
            {
                MessageBox.Show("Lokasi penyimpanan belum dipilih!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }

            if (string.IsNullOrEmpty(txtPassword.Text) ||
txtPassword.Text.Length < 6)
            {
                MessageBox.Show("Ukuran kunci minimal 6 karakter!",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }
            return true;
        }

        private bool decrypt(string cipherData)
        {
            plainText = string.Empty;
            plainText = save.decrypt(
                cipherData,
                txtPassword.Text,
                rs.keySize(),
                rs.keySalt(),
                rs.padding(),
                rs.cipher());
            if (!string.IsNullOrEmpty(plainText))
                return true;
            return false;
        }

        private void extraction()
        {
            stega = new SteganoExtraction();
            stega.steganoFilem4a = txtSteganofile.Text;

            if (stega.extractData())
            {
                if
(!decrypt(Encoding.UTF8.GetString(stega.bytesMessage)))
                    return;
                if (istxt)
                {
                    byte[] blob = Convert.FromBase64String(plainText);

DocCommonFunction.createDocumentBytes(@txtDestinaion.Text + @"\Image
Message.jpg", blob);
                }
                else
```

```
DocCommonFunction.createDocumentString(@txtDestinaion.Text + @"\Text
Message.txt", plainText);

                MessageBox.Show("Extraction data success!",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                isSuccess = true;
            }
            else
            {
                MessageBox.Show("Extraction data failed!", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                isSuccess = false;
            }
        }

        private void reset()
        {
            txtSteganofile.Text = string.Empty;
            txtPassword.Text = string.Empty;
            txtDestinaion.Text = string.Empty;

            lblLoading.Text = "Loading: 0%";
            progressBar1.Value = 0;
        }

        private void openFile(string path)
        {
            if (string.IsNullOrEmpty(path))
                return;
            Process.Start(@path);
        }

        private void enableComonents(bool bol)
        {
            fileToolStripMenuItem.Enabled = bol;
            settingToolStripMenuItem.Enabled = bol;
            txtSteganofile.Enabled = bol;
            txtMessage.Enabled = bol;
            txtDestinaion.Enabled = bol;
            txtPassword.Enabled = bol;
             btnPlaySteganoMP3.Enabled = bol;
            btnOpenSteganofile.Enabled = bol;
            btnOpenMessage.Enabled = bol;
            btnBrowseDestination.Enabled = bol;
            btnExtraction.Enabled = bol;
            btnExit.Enabled = bol;
            btnRestart.Enabled = bol;
            ControlBox = bol;
        }

        #endregion
```

```
        private void btnOpenSteganofile_Click(object sender, EventArgs
e)
        {
            browseFile(txtSteganofile);
        }

        private void btnBrowseDestination_Click(object sender, EventArgs
e)
        {
            browsePathFolder(txtDestinaion);
        }

        private void btnExtraction_Click(object sender, EventArgs e)
        {
            if (!validation())
                return;
            thread = new Thread(new ThreadStart(extraction));
            thread.IsBackground = true;
            thread.Start();
            timer1.Enabled = true;
        }

        private void btnOpenMessage_Click(object sender, EventArgs e)
        {
            openFile(txtMessage.Text);
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            exit();
        }

        private void btnRestart_Click(object sender, EventArgs e)
        {
            restart();
        }

        private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            exit();
        }

        private void FrmExtraction_FormClosed(object sender,
FormClosedEventArgs e)
        {
            exit();
        }

        private void btnPlaySteganoMP3_Click(object sender, EventArgs e)
        {
            playMedia(txtSteganofile.Text);
        }
```

```
        private void btnPausedSteganoMP3_Click(object sender, EventArgs
e)
        {
            pauseMedia();
        }

        private void btnStopSteganoMP3_Click(object sender, EventArgs e)
        {
            stopMedia();
        }

        private void txtSteganofile_DoubleClick(object sender, EventArgs
e)
        {
            browseFile(txtSteganofile);
        }

        private void txtDestinaion_DoubleClick(object sender, EventArgs
e)
        {
            btnBrowseDestination_Click(sender, e);
        }

        private void txtMessage_DoubleClick(object sender, EventArgs e)
        {
            btnOpenMessage_Click(sender, e);
        }

        private void txtSteganofile_TextChanged(object sender, EventArgs
e)
        {
            initMPlayer(txtSteganofile.Text);
        }


        int i = 0;
        private void timer1_Tick(object sender, EventArgs e)
        {
            if (thread.IsAlive)
            {
                enableComonents(false);
                lblLoading.Text = "Loading: " +
stega.persenLoading.ToString() + "%";
                progressBar1.Value =
Convert.ToInt32(stega.persenLoading);
                i++;
            }
            else
            {
                timer1.Enabled = false;
                enableComonents(true);
                if (isSuccess)
                {
                    if (istxt)
```

```
                        txtMessage.Text = @txtDestinaion.Text + @"\Text
Message.txt";
                    else
                        txtMessage.Text = @txtDestinaion.Text + @"\Text
Message.txt";
                }
                else

                    txtMessage.Text = string.Empty;
                reset();
            }
        }

        private void exitToolStripMenuItem1_Click(object sender,
EventArgs e)
        {
            exit();
        }

        private void FrmExtraction_Load(object sender, EventArgs e)
        {
        }
    }
}
```

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik ...../.....

Nama : Ainul Qolby A	Pembimbing I : Dr. Mhd Furqan, S.Si., M.Comp. Sc
NIM : 0701163077	Pembimbing II : Yusuf Ramadhan Nasution,
Prog. Studi : ILMU KOMPUTER	SK Pembimbing :
Judul Skripsi : Text Steganography pada media Audio Format M4A dengan menggunakan metode least significant Bit	

PERT	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I		Pengecekan Proposal			Perbaiki BAB I	
II		Acc Seminar Proposal			Acc BAB 1	
III		Revisi Proposal Skripsi			Acc BAB 2	
IV		Revisi BAB III			Acc BAB 3	
V		Revisi BAB IV			Acc Seminar Proposal	

VI		Revisi BAB V			Revisi Proposal	
VII		Revisi Abstrak			Revisi ~~Abstrak~~ bab IV & V	
VIII		Revisi Kesimpulan			Revisi Abstrak Daftar Pustaka	
IX		Acc Semua bab			Acc Semua bab	
X		Acc sidang			Acc sidang	

Medan, 02 Des 2021
An. Dekan
Ketua Jurusan/Program Studi
ILMU KOMPUTER

ILKA ZUFRIA, M.KOM
NIP. 198506042015031006

Catatan: *Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing*

DAFTAR RIWAYAT HIDUP

(CURRICULUM VITAE)

Nama	: Ainul Qolby Albantany
Nim	: 0701163077
Tempat/Tanggal Lahir	: Medan, 03 Nopember 1998
Jenis Kelamin	: Perempuan
Alamat	: PASAR I LORONG II TIMUR NO. 6A
Kelurahan	: SAMPALI
Kecamatan	: Percut Sei Tuan
Kabupaten	: Deli Serdang
Agama	: Islam
Status Nikah	: Belum Menikah
No.hp	: 082310383813
Nama orang tua	
Ayah	: M. Salman Albantany
Ibu	: Arviani
Email	: ainul.qolby1@gmail.com
Pendidikan formal	
SD	: Sd Muhammadiyah 02 Medan
SMP	: Smp Muhammadiyah 57 Medan
SMK	: Smk Tritech Informatika Medan