

**IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN
METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM
KEAMANAN INFORMASI**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

BINAR PRIHADMANTYO NIM. 1341180029



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2017

**IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN
METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM
KEAMANAN INFORMASI**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

BINAR PRIHADMANTYO NIM. 1341180029



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2017**

HALAMAN PENGESAHAN

IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI

Disusun oleh :

BINAR PRIHADMANTYO NIM. 1341180029

Skripsi ini telah diuji pada tanggal 17 Juli 2017

Disetujui oleh :

1. Penguji I : Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001
2. Penguji II : Luqman Affandi, S.Kom., MMSI.
NIP. 19821130 201404 1 001
3. Pembimbing I : Ely Setyo Astuti, ST., MT.
NIP. 19760515 200912 2 001
4. Pembimbing II : Meyti Eka Apriyani, ST., MT.
.....

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika

Rudy Ariyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa Laporan Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, Agustus 2017

Binar Prihadmantyo

NIM. 1341180029

ABSTRAK

Prihadmantyo, Binar. “Implementasi Algoritma Kriptografi RC4 dan Metode Steganografi Audio 2LSB pada Sistem Keamanan Informasi”. **Pembimbing :**
(1) Ely Setyo Astuti, ST., MT. (2) Meyti Eka Apriyani, ST., MT.

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Memberikan keamanan dan kerahasiaan terhadap informasi sangat diperlukan ketika melakukan pertukaran informasi melalui jaringan komunikasi. Hal tersebut bertujuan agar informasi yang dikirimkan oleh pengirim dapat diterima secara utuh oleh penerima tanpa ada campur tangan dari pihak yang tidak berkepentingan terhadap informasi.

Teknik kriptografi dan steganografi dapat digunakan untuk pengamanan pesan rahasia. Dengan membangun aplikasi yang mengombinasikan antara kedua teknik ini dapat memberikan keamanan terhadap pesan rahasia dengan baik. Teknik pengamanan yang dapat digunakan yaitu teknik kriptografi dengan menggunakan algoritma RC4 untuk mengamankan pesan rahasia berupa teks atau citra, dan penyisipan pesan rahasia dengan metode steganografi 2LSB ke dalam media audio.

Analisa yang dilakukan adalah tingkat keberhasilan proses enkripsi dan penyisipan, ekstraksi dan dekripsi, kecepatan proses, serangan stego audio, dan kualitas audio. Hasil dari 18 kali pengujian proses enkripsi dan penyisipan serta ekstraksi dan dekripsi pesan, didapatkan persentase keberhasilan sebesar 100% dengan waktu proses yang berbeda-beda bergantung pada ukuran pesan. Stego audio yang dihasilkan memiliki kualitas baik dan tidak menimbulkan *noise* yang dapat didengar oleh indra pendengaran manusia secara langsung, akan tetapi stego audio tidak tahan terhadap serangan yang menyebabkan perubahan terhadap nilai *byte* file stego. Sehingga dapat disimpulkan bahwa pengombinasi antara algoritma kriptografi RC4 dan metode steganografi 2LSB dapat mengamankan pesan dengan baik dan memberikan hasil dekripsi tanpa ada perubahan pada pesan yang disisipkan.

Kata kunci : kriptografi, steganografi, RC4, 2LSB

ABSTRACT

Prihadmantyo, Binar. “*The Implementation of RC4 Cryptography Algorithm and Audio Steganography 2 LSB Method on Information Security System*”. ***Advisor :*** (1) ***Ely Setyo Astuti, ST., MT.*** (2) ***Meyti Eka Apriyani, ST., MT.***

Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.

Providing security and confidentiality to information is essential when exchanging information through communication networks. It is intended that the information sent by the sender can be received completely by the recipient without any interference from other parties who have no authorities to access the information.

Cryptography and steganography techniques can be used to secure secret messages. Creating an application that combines these techniques can provide security to a secret message well. The security techniques that can be used are cryptographic techniques using RC4 algorithm to secure secret messages in the form of text or image, and the insertion of secret messages with 2LSB steganography method into the audio media.

The analysis performed are the success rate of encryption and insertion process, extraction and decryption process, process speed, audio stego attack, and audio quality. From 18 tests of encryption, insertion, extraction and decryption of messages, the percentage of success was 100% with different processing time depends on the message size. The audio stego has good quality and did not cause any noise that can be heard by human directly, but it is not resistant to attacks that cause changes to the byte value of the stego file. Therefore, it can be concluded that the combination between RC4 cryptographic algorithm and 2LSB steganography method can secure the message well and provide the results of decryption without any changes to the message.

Keywords: *cryptography, steganography, RC4, 2LSB*

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan skripsi ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terimakasih kepada:

1. Tuhan Yang Maha Esa yang telah memberikan petunjuk dan hidayah dalam pembuatan skripsi dan penyusunan laporan sehingga dapat berjalan dengan baik dari awal hingga akhir.
2. Kedua orangtua kami yang telah memberikan doa dan dukungannya.
3. Bapak Rudy Ariyanto, S.T.,MCs selaku ketua jurusan Teknologi Informasi.
4. Bapak Ir. Deddy Kusbianto P. A., MMKom selaku ketua program studi Teknik Informatika.
5. Ibu Ely Setyo Astuti, ST.,MT dan ibu Meyti Eka Apriyani, ST., MT. selaku pembimbing skripsi.
6. Seluruh dosen dan karyawan program studi Teknik Informatika, jurusan Teknologi Informasi yang membantu pembuatan skripsi.
7. Teman-teman dari Program Studi Teknik Informatika angkatan 2013 yang selalu memberikan semangat dan dukungan selama penyelesaian skripsi ini,
8. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan skripsi ini. Semoga skripsi ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, Agustus 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
PERNYATAAN	iii
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	3
BAB II. LANDASAN TEORI.....	5
2.1 Penelitian Sebelumnya	5
2.2 Kriptografi	5
2.2 Steganografi.....	6
2.3 Audio	7
2.4 Algoritma Kriptografi RC4	8
2.5 Metode <i>Least Significant Bit</i> (LSB)	9
BAB III. METODOLOGI.....	10
3.1 Studi Literatur.....	10
3.2 Analisa.....	10
3.3 Implementasi	11
3.4 Pengkodean Sistem.....	13
3.5 Pengujian Sistem	13
3.6 Evaluasi Sistem	13
BAB IV. ANALISA DAN PERANCANGAN.....	14
4.1 Dekripsi Sistem	14

4.2	Analisis Kebutuhan Non Fungsional.....	14
4.3	Analisis Kebutuhan Fungsional.....	15
4.4	Batasan Sistem	16
4.5	Desain Sistem	16
4.6	Rancangan <i>User Interface</i>	47
4.7	Perancangan Perhitungan	50
BAB V. IMPLEMENTASI.....		56
5.1	Pembuatan Aplikasi.....	56
5.2	Pembuatan Jendela <i>Home</i>	56
5.3	Pembuatan Jendela <i>Text Encryption</i>	56
5.4	Pembuatan Jendela <i>Image Encryption</i>	61
5.5	Pembuatan Jendela <i>Text Decryption</i>	65
5.6	Pembuatan Jendela <i>Image Decryption</i>	69
BAB VI. PENGUJIAN DAN PEMBAHASAN.....		73
6.1	Pengujian Sistem	73
6.2	Pengujian Hasil.....	76
BAB VII. KESIMPULAN		92
7.1	Kesimpulan.....	92
7.2	Saran	93
DAFTAR PUSTAKA		95
LAMPIRAN.....		96

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Diagram Alur Sistem	11
Gambar 3.2 Siklus Pengembangan <i>Prototype</i>	12
Gambar 4.1 <i>Use Case Diagram</i> Sistem	16
Gambar 4.2 <i>Sequence Diagram</i> Enkripsi dan Penyisipan Pesan Teks	20
Gambar 4.3 <i>Sequence Diagram</i> Enkripsi dan Penyisipan Citra	21
Gambar 4.4 <i>Sequence Diagram</i> Ekstraksi dan Dekripsi Pesan Teks.....	22
Gambar 4.5 <i>Sequence Diagram</i> Ekstraksi dan Dekripsi Citra'	22
Gambar 4.6 <i>Flowchart Diagram</i> Enkripsi dan Penyisipan Pesan Teks.....	23
Gambar 4.7 <i>Flowchart Diagram</i> Enkripsi Pesan Teks Menggunakan Algoritma RC4	24
Gambar 4.8 <i>Flowchart Diagram</i> Konversi Cipherteks ke Susunan Byte Pesan....	25
Gambar 4.9 <i>Flowchart Diagram</i> Konversi File Audio ke Susunan Byte Audio ...	26
Gambar 4.10 <i>Flowchart Diagram</i> Penyisipan Byte Pesan ke dalam Byte Audio Menggunakan Metode 2LSB	27
Gambar 4.11 <i>Flowchart Diagram</i> Konversi Byte Stego Audio ke File Stego Audio	28
Gambar 4.12 <i>Flowchart Diagram</i> Enkripsi dan Penyisipan Citra.....	29
Gambar 4.13 <i>Flowchart Diagram</i> Mengubah Citra ke bentuk Citra Heksadesimal	30
Gambar 4.14 <i>Flowchart Diagram</i> Enkripsi Citra Heksadesimal Menggunakan Algoritma RC4	31
Gambar 4.15 <i>Flowchart Diagram</i> Konversi Cipher Citra ke Byte Citra	32
Gambar 4.16 <i>Flowchart Diagram</i> Konversi File Audio ke Byte Audio.....	33
Gambar 4.17 <i>Flowchart Diagram</i> Penyisipan Byte Citra kedalam Byte Audio Menggunakan Metode 2LSB	34
Gambar 4.18 <i>Flowchart Diagram</i> Konversi Byte Stego Audio ke File Stego Audio	35
Gambar 4.19 <i>Flowchart Diagram</i> Dekripsi dan Ekstraksi Pesan Teks	36
Gambar 4.20 <i>Flowchart Diagram</i> Mengubah File Stego Audio ke Susunan Byte Stego.....	37
Gambar 4.21 <i>Flowchart Diagram</i> Ekstraksi Byte Pesan dari Byte Stego Menggunakan Metode 2LSB	38
Gambar 4.22 <i>Flowchart Diagram</i> Konversi Byte Pesan menjadi Cipherteks	39
Gambar 4.23 <i>Flowchart Diagram</i> Dekripsi Cipherteks Menggunakan Algoritma RC4	40
Gambar 4.24 <i>Flowchart Diagram</i> Dekripsi dan Ekstraksi Citra	41
Gambar 4.25 <i>Flowchart Diagram</i> Mengubah File Stego ke Susunan Byte Stego.	42
Gambar 4.26 <i>Flowchart Diagram</i> Ekstraksi Byte Citra dari Byte Stego Menggunakan Metode 2LSB	43
Gambar 4.27 <i>Flowchart Diagram</i> Konversi Byte Citra Menjadi Cipher Citra.....	44
Gambar 4.28 <i>Flowchart Diagram</i> Dekripsi Cipher Citra Menggunakan Algoritma RC4	45
Gambar 4.29 <i>Flowchart Diagram</i> Konversi Citra Heksadesimal ke File Citra	46
Gambar 4.30 <i>Class Diagram</i> Sistem	47

Gambar 4.31 Rancangan Jendela <i>Home</i>	48
Gambar 4.32 Rancangan Jendela <i>Text Encryption</i>	48
Gambar 4.33 Rancangan Jendela <i>Image Encryption</i>	49
Gambar 4.34 Rancangan Jendela <i>Text Decryption</i>	49
Gambar 4.35 Rancangan Jendela <i>Image Decryption</i>	50
Gambar 5.1 Tampilan Jendela <i>Home</i>	56
Gambar 5.2 Tampilan Jendela <i>Text Encryption</i>	57
Gambar 5.3 Masukan Enkripsi Pesan Teks	57
Gambar 5.4 Cipherteks Hasil Enkripsi	58
Gambar 5.5 Masukan Penyisipan Pesan Teks	59
Gambar 5.6 Perbandingan File Audio Sebelum dan Setelah Disisisipi Pesan Teks	60
Gambar 5.7 Tampilan Jendela <i>Image Encryption</i>	61
Gambar 5.8 Masukan Enkripsi Citra	61
Gambar 5.9 Cipher Citra Hasil Enkripsi.....	63
Gambar 5.10 Masukan Penyisipan Citra	63
Gambar 5.11 Perbandingan File Audio Sebelum dan Setelah Disisisipi Citra	65
Gambar 5.12 Tampilan Jendela <i>Text Decryption</i>	66
Gambar 5.13 Masukan Ekstraksi Pesan Teks	66
Gambar 5.14 Cipherteks Hasil <i>Decoding</i>	67
Gambar 5.15 Masukan Dekripsi Pesan Teks	67
Gambar 5.16 Pesan Teks Terdekripsi	68
Gambar 5.17 Tampilan Jendela <i>Image Decryption</i>	69
Gambar 5.18 Masukan Ekstraksi Citra	69
Gambar 5.19 Cipher Citra Hasil <i>Decoding</i>	70
Gambar 5.20 Masukan Dekripsi Citra	71
Gambar 5.21 Citra Terdekripsi	72
Gambar 6.1 Grafik Kecepatan Proses Enkripsi dan Penyisipan Pesan Teks	86
Gambar 6.2 Grafik Kecepatan Proses Enkripsi dan Penyisipan Citra	86
Gambar 6.3 Grafik Kecepatan Proses Ekstraksi dan Dekripsi Pesan Teks	87
Gambar 6.4 Grafik Kecepatan Proses Ekstraksi dan Dekripsi Citra	87

DAFTAR TABEL

	Halaman
Tabel 4.1 Deskripsi <i>Use Case Diagram</i>	17
Tabel 6.1 Pengujian Sistem.....	73
Tabel 6.2 Pengujian Notifikasi <i>Error</i>	75
Tabel 6.3 File Uji Audio	76
Tabel 6.4 File Uji Citra	76
Tabel 6.5 Sampel Uji Pesan Teks	77
Tabel 6.6 Enkripsi Pesan Teks dan Citra pada File Audio	78
Tabel 6.7 Dekripsi Pesan Teks dan Citra pada Stego Audio	79
Tabel 6.8 Hasil Keluaran Dekripsi Pesan Teks pada File Stego Audio.....	80
Tabel 6.9 Hasil Keluaran Dekripsi Citra pada File Stego Audio.....	84
Tabel 6.10 Pengujian Serangan Stego Audio	88
Tabel 6.11 Pengujian PSNR	90

DAFTAR LAMPIRAN

- Lampiran 1. Kode Program
- Lampiran 2. Identitas Penulis
- Lampiran 3. Lembar Bimbingan Pembimbing 1
- Lampiran 4. Lembar Bimbingan Pembimbing 2
- Lampiran 5. Lembar Persetujuan Maju Ujian
- Lampiran 6. Form Revisi Pengaji 1
- Lampiran 7. Form Revisi Pengaji 2
- Lampiran 8. Form Verifikasi Abstrak Bahasa Inggris dan Tata Tulis Buku Skripsi

BAB I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang semakin pesat, membuat semua kegiatan manusia menjadi lebih mudah dan cepat. Semakin berkembangnya teknologi dan layanan internet, setiap orang dapat dengan mudah untuk melakukan pertukaran data atau informasi kepada orang lain tanpa dibatasi oleh batas-batas jarak dan waktu. Hal tersebut juga turut mempengaruhi berkembangnya kejahatan yang memanfaatkan teknologi informasi dan komunikasi. Informasi yang ditransmisikan pada jaringan komunikasi dari pengirim ke penerima, rentan untuk diakses oleh pihak lain yang tidak berkepentingan. Dengan demikian, keamanan dan kerahasiaan menjadi suatu kebutuhan penting dalam melakukan pertukaran informasi yang ditransmisikan melalui jaringan komunikasi. Dengan demikian diperlukan suatu sistem keamanan informasi yang berguna untuk mengamankan informasi berupa file atau pesan teks yang akan ditransimiskan melalui jaringan komunikasi oleh pengirim kepada penerima informasi.

Dalam bidang keamanan informasi, ada dua teknik yang digunakan untuk mengamankan informasi, yaitu teknik kriptografi dan steganografi. Kriptografi merupakan teknik untuk menyamarkan informasi dalam bentuk pesan bermakna menjadi pesan yang tidak bermakna. Sedangkan steganografi ialah teknik untuk menyembunyikan informasi ke dalam suatu media atau wadah pembawa informasi.

Kriptografi memiliki manfaat yaitu untuk menjaga atau mengamankan informasi dari pihak-pihak yang tidak berkepentingan, dengan cara menyandikan informasi tersebut. Informasi dienkripsi dengan algoritma tertentu agar tidak dapat dibaca dan dimengerti oleh orang lain. Namun karena informasi yang telah dienkripsi memiliki struktur acak dan sulit dimengerti maknanya, sangatlah mungkin menimbulkan kecurigaan orang lain, sebab informasi yang seperti demikian pasti sudah diolah dan menunjukkan bahwa informasi tersebut bersifat penting dan rahasia. Hal ini dapat memancing orang lain untuk memecahkan informasi rahasia tersebut. Untuk menghindari permasalahan tersebut, dapat diatasi dengan menggunakan teknik steganografi.

Steganografi memiliki manfaat yaitu untuk menyembunyikan informasi rahasia kedalam suatu media pembawa informasi, sehingga keberadaan informasi yang dikirimkan tidak dapat diketahui oleh orang lain. Aspek terpenting dari steganografi biasanya terletak pada penyembunyian informasi kedalam media pembawa informasi, dengan tingkat perubahan yang tidak signifikan pada media pembawa informasi sebelum dan setelah disisipi pesan.

Dengan mengombinasikan antara teknik kriptografi dan steganografi, akan memberikan keamanan yang baik dalam mengamankan informasi rahasia. Sehingga dapat memenuhi kriteria keamanan terhadap informasi, yaitu informasi yang diamankan tidak dapat diketahui maksud dan keberadaannya oleh pihak yang tidak berkepentingan dan informasi yang diamankan dapat diambil kembali secara utuh tanpa ada perubahan pada informasi [1].

Dalam penelitian ini menerapkan penggabungan antara teknik steganografi dan kriptografi, sehingga dapat mengamankan informasi rahasia dengan lebih baik. Implementasi enkripsi dan dekripsi pesan dengan menggunakan algoritma kriptografi RC4. Informasi yang telah dienkripsi kemudian disembunyikan pada media audio menggunakan metode *2 Least Significant Bit* (2LBS). Aplikasi yang dibangun dapat digunakan oleh pengguna individu ataupun instansi yang membutuhkan pengamanan terhadap informasi berupa file citra dan pesan teks yang dikirimkan melalui jaringan komunikasi kepada penerima informasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang dijelaskan diatas, rumusan masalah yang didapat antara lain :

1. Bagaimana cara menerapkan teknik keamanan informasi yang dapat memberikan keamanan yang baik pada informasi rahasia?
2. Bagaimana cara mengetahui parameter yang memengaruhi algoritma kriptografi RC4 dan metode steganografi 2LSB?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka tujuan dari penelitian ini adalah

1. Membuat sistem yang berguna untuk mengamankan informasi rahasia, dengan mengombinasikan algoritma kriptografi RC4 dan metode steganografi 2LSB.
2. Menganalisis parameter keamanan informasi dengan menggunakan algoritma kriptografi RC4 dan metode steganografi 2LSB.

1.4 Batasan Masalah

Agar aplikasi ini dapat berjalan sesuai dengan tujuan yang direncanakan, maka diperlukan batasan-batasan masalah, yaitu :

1. Aplikasi berbasis desktop.
2. Informasi yang dapat disispkan adalah berbentuk teks (*unformatted text*) dan file citra dengan ekstensi .jpg.
3. Penyisipan informasi dilakukan pada file audio dengan ekstensi .wav.
4. Metode pengamanan informasi menggunakan algoritma kriptografi RC4 dan metode steganografi 2LSB.

1.5 Sistematika Penulisan

- | | |
|---------|---|
| BAB I | Pendahuluan berisikan latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan. |
| BAB II | Landasan teori berisikan teori-teori yang melengkapi latar belakang. |
| BAB III | Metodologi berisikan langkah-langkah memilih metode yang tepat sehingga setiap tahap penelitian dilakukan dengan tepat. |
| BAB IV | Analisis dan Perancangan berisikan uraian sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan non fungsional. |
| BAB V | Implementasi berisikan uraian sistem sesuai rancangan dan bahasa pemrograman yang dipakai. |

- BAB VI Pengujian dan pembahasan berisikan proses untuk menentukan apakah hasil dari tugas akhir sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Pembahasan merupakan argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian.
- BAB VII Kesimpulan berisikan uraian singkat dan jelas tentang hasil tugas akhir yang diperoleh sesuai dengan tujuan penelitian.

BAB II. LANDASAN TEORI

Bab ini berisikan tentang teori yang digunakan sebagai dasar melakukan penelitian. Teori tersebut kemudian dipakai untuk mendukung pembuatan aplikasi, rancangan metode, serta pengujian yang dilakukan dalam penelitian.

2.1 Penelitian Sebelumnya

Penelitian sebelumnya yang berjudul “Perancangan dan Implementasi Aplikasi Steganografi Citra Digital dengan Metode 2LSB”, yang ditulis oleh Eko Krist Setyono dan M.A. Ineke Pakareng pada tahun 2014, membahas tentang penggunaan teknik steganografi 2LSB dengan media citra digital sebagai media pembawa informasi [2]. Penggunaan teknik 2LSB memberikan kapasitas penyimpanan yang lebih banyak, yaitu 2 bit terakhir dari tiap 1 *byte* atau 8 bit citra (bit ke 7 dan bit ke 8). Dari hasil pengujian menunjukkan bahwa media pembawa informasi sebelum dan sesudah disisipi pesan, secara visual tidak menampakkan perbedaan yang signifikan, serta pesan yang disisipkan tidak mengalami perubahan ketika diekstraksi.

Penelitian lainnya yang berjudul “Analisa Algoritma Kriptografi RC4 Pada Enkripsi Citra Digital” yang ditulis oleh Galuh Adjeng Sekarsari pada tahun 2015, menunjukkan bahwa algoritma RC4 dapat melakukan enkripsi dan dekripsi citra dalam waktu singkat, dengan selisih waktu yang tidak terpaut jauh ketika mengenkripsi dan mendekripsi citra yang berbeda ekstensi (BMP, JPEG, PNG) [3].

2.2 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, yaitu “*cyrptos*” yang artinya “rahasia” dan “*graphein*” artinya “tulisan”, sehingga kriptografi berarti “tulisan rahasia”. Kriptografi didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya [1].

Kriptografi memiliki dua konsep utama, yaitu enkripsi (*encryption*) dan dekripsi (*decryption*) [4]. Enkripsi adalah sebuah proses menyandikan plainteks menjadi cipherteks. Sedangkan dekripsi adalah proses mengembalikan cipherteks

menjadi plainteks semula. Dalam melakukan enkripsi dan dekripsi pesan, dibutuhkan kunci sebagai parameter yang digunakan untuk transformasi.

Kriptografi terbagi menjadi dua, yaitu :

- a. Kriptografi klasik, penyandian dilakukan dengan memanipulasi karakter tradisional berupa huruf dan angka secara langsung. Sistem kriptografi klasik terdiri dari dua macam, yaitu :
 - Cipher substitusi, sistem kriptografi ini melakukan penyandian dengan cara mensubstitusi huruf dengan huruf yang lain sesuai dengan yang ditetapkan.
 - Cipher transposisi, penyandian pesan dilakukan dengan cara mengubah letak (posisi) dari pesan teks yang akan disandikan.
- b. Kriptografi modern, penyandian beroperasi dalam mode bit. Sistem kriptografi modern terdiri dari dua macam, yaitu :
 - Kriptografi kunci simetris, merupakan sistem kriptografi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi pesan. Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya. Sistem kriptografi kunci simetris mengasumsikan pengirim dan penerima pesan sudah melakukan transaksi kunci sebelum bertukar pesan. Sistem kriptografi kunci simetris terbagi menjadi dua yaitu chiper aliran (*stream cipher*) yang melakukan penyandian pesan tiap karakter (misal : RC4) dan cipher blok (*block cipher*) yang melakukan penyandian pesan pada tiap blok bit (misal : AES dan DES).
 - Kriptografi kunci asimetris, merupakan sistem kriptografi yang menggunakan kunci untuk enkripsi tidak rahasia (diumumkan ke publik), sedangkan kunci untuk dekripsi pesan hanya diketahui oleh penerima pesan (karena bersifat rahasia). Pada kriptografi asimetris, setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Contoh algoritma kriptografi asimetris diantaranya, RSA, Elgamal, dan DSA.

2.2 Steganografi

Steganografi (*steganography*) berasal dari bahasa Yunani, yaitu “*steganos*” yang artinya “tersembunyi” dan “*graphein*” yang berarti “tulisan”, sehingga

steganografi dapat diartikan “tulisan tersembunyi”. Steganografi adalah ilmu dan seni untuk menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia [5].

Steganografi membutuhkan dua properti utama, yaitu wadah penampung dan pesan/ data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, seperti citra, audio, teks dan video. Begitu juga data rahasia yang disembunyikan juga dapat berupa citra, audio, teks dan video.

Proses penyisipan pesan rahasia ke dalam media dinamakan *encoding*, sedangkan ekstrasi pesan dari *stego object* dinamakan *decoding*. Kedua proses ini biasanya memerlukan kunci rahasia (*stegokey*) agar pihak yang berkepentingan saja yang dapat melakukan penyisipan pesan dan ekstrasi.

2.3 Audio

Audio adalah suara atau bunyi yang dihasilkan dari getaran suatu benda. Di butuhkan getaran minimal 20 kali/detik, agar audio dapat didengar oleh telinga manusia. Sinyal audio dibagi menjadi dua macam, yaitu analog dan digital [6]. audio analog memproduksi variasi suara dengan membuat atau membaca variasi sinyal listrik. Sedangkan audio digital memproduksi suara dengan mengambil sampel tekanan suara atau level sinyal pada rate tertentu dan mengubahnya menjadi angka. Audio digital memiliki berbagai macam ekstensi atau format, beberapa diantaranya sebagai berikut.

a. WAV

WAV merupakan standar format file audio yang digunakan oleh Windows. WAV umumnya digunakan untuk menyimpan audio tak terkompresi, file suara berkualitas audio CD dengan ukurannya relatif besar (sekitar 10 MB per menit).

b. MP3

MP3 merupakan format audio yang populer saat ini untuk pengunduhan dan penyimpanan lagu. Dengan mengeliminasi sebagian dari audio yang tidak terdengar, file audio .mp3 dikompresi secara signifikan sampai 1/10 dari file PCM, akan tetapi tetap mempertahankan kualitas audio.

c. VOX

Format audio VOX banyak digunakan untuk codec ADPCM (*Adaptive Differential Pulse Code Modulation*). VOX mirip dengan WAV, namun tidak memuat informasi tentang file itu sendiri, sehingga *rate* codec dan jumlah *channel* harus diberikan terlebih dahulu untuk memainkannya.

2.4 Algoritma Kriptografi RC4

RC4 merupakan jenis dari aliran kode yang berarti operasi enkripsinya dilakukan setiap karakter 1 *byte* (8 bit) untuk sekali operasi. Algoritma ini ditemukan pada tahun 1978 oleh Ronald Rivest dan menjadi simbol keamanan RSA. RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte* [7]. Tabel tersebut digunakan untuk generasi berikut dari *pseudo random* yang melakukan XOR dengan plainteks untuk menghasilkan keluaran berupa cipherteks. Setiap elemen dalam tabel saling ditukarkan minimal sekali penukaran.

RC4 menggunakan dua buah kotak substitusi atau S-Box. S-Box pertama berupa array 256 *byte* yang berisikan permutasi dari bilangan 0 sampai 255, dan S-Box kedua berisikan permutasi fungsi dari kunci dengan panjang variabel. Cara kerja algoritma RC4 yaitu menginisialisasi S-box pertama, $S[0]$, $S[1]$ hingga $S[255]$, dengan bilangan 0 sampai 255. Pertama mengisi secara berurutan $S[0] = 0$, $S[1] = 1$ hingga $S[255] = 255$. Selanjutnya menginisialisasi S-Box kedua, misal array K dengan panjang 256. Mengisi array K dengan kunci yang diulangi hingga seluruh array $K[0]$, $K[1]$ hingga $K[255]$ terisi seluruhnya.

- Proses inisialisasi array S

For r = 0 to 255

$S[r] = r$

- Proses inisialisasi *state* array K

Array Kunci // panjang kunci"length"

for i = 0 to 255

$K[i] = \text{Kunci } [i \bmod \text{length}]$

- Kemudian dilakukan pengacakan S-Box dengan langkah sebagai berikut :

$j = 0$

For i = 0 to 255

$$j = (j + S[i] + K[i]) \bmod 256$$

isi $S[i]$ dan isi $S[j]$ ditukar

Proses berikutnya ialah membangkitkan kunci enkripsi, dilakukan proses sebagai berikut:

- a. $i = j = 0$
- b. $i = (i + 1) \bmod 256$
- c. $j = (j + S[i]) \bmod 256$
- d. isi $S[i]$ dan $S[j]$ ditukar
- e. $k = S[S[i] + S[j]] \bmod 256$

Perlu diperhatikan bahwa k kecil merupakan kunci yang langsung beroperasi terhadap plainteks, sedangkan K besar adalah kunci utama.

2.5 Metode Least Significant Bit (LSB)

Metode LSB merupakan metode steganografi yang paling sederhana dan mudah diimplementasikan [5]. Pada susunan bit di dalam sebuah *byte* ($1\text{ byte} = 8\text{ bit}$), ada bit yang paling berarti (*most significant bit* atau MSB) dan bit yang paling kurang berarti (*least significant bit* atau LSB). Sebagai contoh *byte* 11010110, angka bit 1 (pertama, digaris-bawahi) adalah bit MSB, dan angka bit 0 (terakhir, digaris-bawahi) adalah bit LSB. Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

Misalkan, susunan *byte* pada media digital yang belum disisipi adalah sebagai berikut.

00110011 10100010 11100010 01101111

Pesan rahasia yang telah diubah ke susunan biner misalkan, ‘0111’. Maka susunan *byte* media digital setelah pesan rahasia disembunyikan adalah sebagai berikut.

00110010 10100011 11100011 01101111

Jika penyisipan bit menggunakan teknik 2LSB, maka penyisipan pesan rahasia diletakkan pada 2 bit terakhir pada susunan *byte* media. Sehingga susunan *byte* media digital setelah pesan rahasia disembunyikan adalah sebagai berikut.

00110001 10100011 11100010 01101111

BAB III. METODOLOGI

Bab ini menjelaskan langkah-langkah yang dilakukan untuk implementasi algoritma kriptografi RC4 dan metode steganografi audio 2LSB pada sistem keamanan informasi.

3.1 Studi Literatur

Pada tahap ini penelitian dilakukan dengan mempelajari berbagai literature melalui pengumpulan dokumen-dokumen, referensi-referensi, buku-buku, sumber dari internet, atau sumber lain yang diperlukan untuk merancang dan mengimplementasikan sistem yang berkaitan dengan penulisan skripsi yang dilakukan.

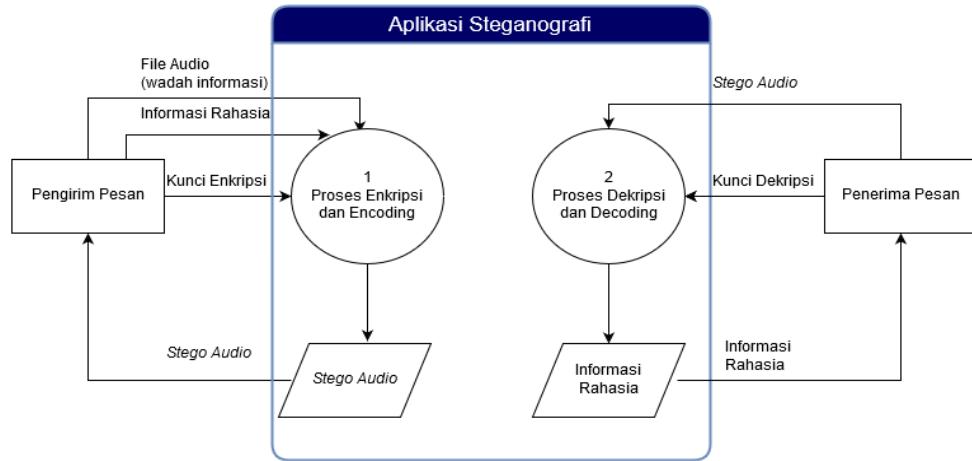
3.2 Analisa

Tujuan menganalisa antara lain menganalisa kebutuhan dan keperluan dasar yang digunakan dalam merancang dan mengimplementasikan sistem yang diinginkan. Hasil perancangan yang diperoleh adalah pembuatan aplikasi yang dapat mengamankan pesan/ data rahasia menggunakan algoritma kriptografi RC4 dan metode steganografi 2LSB.

Aplikasi yang dibangun menggunakan kunci simetris. Kunci simetris adalah penggunaan kunci yang sama untuk melakukan proses enkripsi dan dekripsi. Sehingga apabila kunci yang dimasukan untuk dekripsi berbeda dengan kunci pada saat enkripsi, maka pesan/ data tidak akan kembali ke bentuk awal sebelum terenkripsi.

Media yang digunakan sebagai wadah pembawa pesan rahasia berupa file audio dengan ekstensi .wav. Sedangkan media yang disisipkan pada wadah pembawa pesan berupa pesan teks dan file citra. Pesan rahasia yang disispkan kedalam media audio terlebih dahulu dienkripsi menggunakan algoritma kriptografi RC4, kemudian dilakukan penyisipan kedalam media pembawa pesan menggunakan metode steganografi 2LSB.

Adapun desain sistem yang dibuat digambarkan skema dibawah ini :



Gambar 3.1 Diagram Alur Sistem

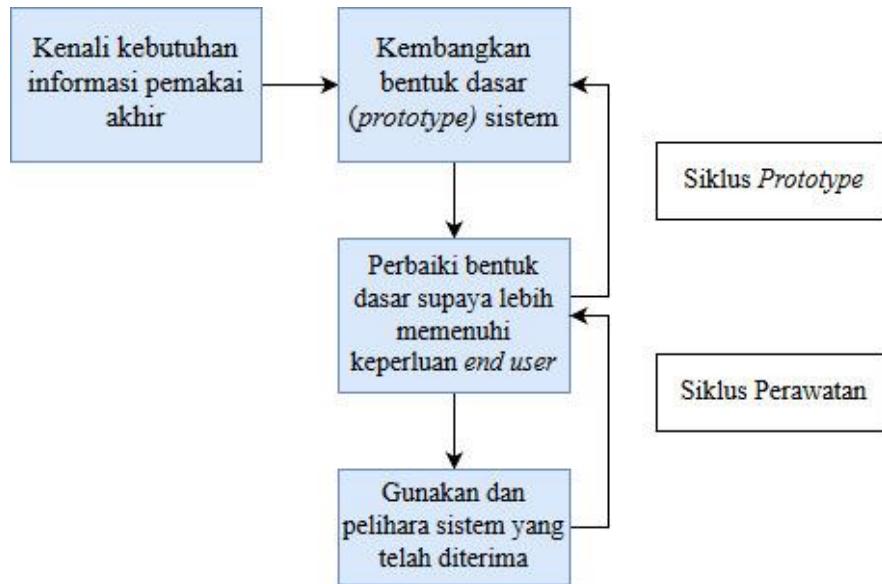
Pada gambar 3.1 dijelaskan bahwa pengirim pesan memerlukan *input* berupa informasi rahasia yang akan dikirim, file audio sebagai wadah informasi, dan kunci untuk enkripsi. Selanjutnya sistem melakukan proses enkripsi pesan dan *encoding* (penyembunyian informasi), kemudian menghasilkan sebuah *stego audio*.

Pada pihak penerima pesan memerlukan *input* berupa *stego audio* dan kunci dekripsi. Selanjutnya sistem melakukan proses *decoding* (ekstraksi) dan dekripsi pesan, agar informasi dapat dibaca oleh penerima pesan.

3.3 Implementasi

Implementasi algoritma kriptografi RC4 dan metode steganografi audio 2LSB pada sistem keamanan informasi mengacu pada metode pengembangan *Prototype*. Metode *prototype* merupakan salah satu jenis metode pengembangan sistem yang cepat dan dapat menghemat waktu. *Prototype* disebut juga dengan desain aplikasi cepat (*rapid application design/ RAD*) karena menyederhanakan dan mempercepat desain sistem [8].

Gambaran pengembangan aplikasi menggunakan metode *prototype* yang diberikan oleh O'Brien.



Gambar 3.2 Siklus Pengembangan *Prototype*

Pada gambar 3.2 dijelaskan siklus pengembangan dengan menggunakan model *prototype*. Berikut ini merupakan penjelasan dalam pengembangan yang dilakukan.

- Penyelidikan/ Analisa : *End user* mengenali kebutuhan informasi mereka dan menjelaskan beberapa kemungkinan pemecahan sistem informasi cadangan.
- Analisa/ Rancangan : *End user* dan *system analyst* menggunakan paket pengembangan penggunaan (*application development packages*) untuk rancangan yang menarik dan menguji bentuk dasar dari bagian sistem informasi yang memenuhi kebutuhan informasi dari *end user*.
- Rancangan/ Penerapan : Bentuk dasar sistem informasi diuji berulang kali hingga *end user* mendapatkan sistem dapat diterima.
- Penerapan/ Pemeliharaan : Sistem inforamasi yang diterima dapat diubah dengan mudah setelah kebanyakan dokumentasi sistem disimpan dalam penyimpanan.

3.4 Pengkodean Sistem

Pada tahap ini memulai membangun sistem yang sesuai dengan perancangan sebelumnya. Pembangunan Pembangunan sistem dimulai dengan :

a. Enkripsi pesan/ data rahasia

Tahap ini dilakukan mengubah informasi yang dapat dimengerti menjadi barisan kode yang tidak bermakna dan tidak dapat dibaca. Tahapan ini dilakukan untuk merahasiakan isi informasi.

b. Penyisipan informasi ke media audio

Tahap menyisipkan informasi rahasia ke dalam file audio dengan metode 2LSB. Penyisipan dilakukan dengan mengubah 2 bit terakhir pada setiap 1 *byte* dari file audio agar perubahan file tidak terlalu signifikan.

c. Penyusunan bit-bit

Tahap ini menyusun kembali susunan bit dari informasi rahasia yang disisipkan pada file audio sebagai pembawa informasi rahasia, untuk dapat mengambil kembali informasi rahasia tersebut.

d. Dekripsi pesan/ data rahasia

Tahap untuk merubah barisan kode yang tidak bermakna dan tidak dapat dimengerti, menjadi informasi yang bermakna dan dapat dimengerti. Tahap ini dilakukan untuk mengetahui informasi rahasia yang disampaikan.

3.5 Pengujian Sistem

Tahap pengujian sistem bertujuan untuk memastikan bahwa hasil dari sistem sesuai dengan hasil dari perencanaan sebelumnya. Pengujian sistem terhadap tingkat perubahan file audio yang disisipi pesan dengan cara melakukan perbandingan antara file audio sesudah dan sebelum disisipi pesan rahasia. Dan melakukan pengujian proses enkripsi dan dekripsi pesan beserta kecepatan proses.

3.6 Evaluasi Sistem

Melakukan evaluasi dan perbaikan sistem apabila sistem belum sesuai dengan apa yang diharapkan atau belum sesuai dengan konsep steganografi dan algoritma kriptografi.

BAB IV. ANALISA DAN PERANCANGAN

Bab ini berisikan uraian sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan non fungsional.

4.1 Dekripsi Sistem

Aplikasi pengamanan informasi rahasia ini dibangun untuk mengamankan informasi lebih baik, dengan menggabungkan teknik kriptografi menggunakan algoritma RC4 dan steganografi menggunakan metode 2LSB kedalam media audio. Aplikasi ini menekankan pada kecepatan proses enkripsi dan dekripsi, sehingga dapat memberikan efisiensi waktu pada pengguna ketika menggunakannya.

Aplikasi ini membutuhkan masukan berupa pesan teks dan file citra sebagai informasi rahasia yang diamankan, serta masukan berupa file audio dengan ekstensi .wav yang digunakan sebagai media pembawa pesan rahasia. Untuk kunci yang digunakan berupa kunci simetris, yang artinya kunci yang dimasukan untuk enkripsi harus sama dengan kunci untuk mendekripsi pesan rahasia.

Aplikasi ini berbasis desktop, dibangun dengan menggunakan bahasa pemrogramman VB.NET. Dengan adanya aplikasi ini, diharapkan pengguna dapat mendapatkan keamanan yang lebih untuk mengamankan informasi rahasia.

4.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan sistem yang dibangun. Spesifikasi kebutuhan meliputi analisis perangkat lunak, perangkat keras, dan pengguna.

a. Analisis kebutuhan perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

- Microsoft Windows 10

Sistem Operasi yang bertugas untuk melakukan kontrol dan manajemen perangkat keras serta operasi-operasi dasar sistem, termasuk untuk menjalankan aplikasi.

- Microsoft Visual Studio 2012

Aplikasi Editor utama untuk pembangunan aplikasi.

- Microsoft .NET Framework 4.5
Komponen tambahan yang diperlukan untuk Microsoft Visual Studio 2012.
- HxD Hexaeditor
Aplikasi untuk memanipulasi data dari suatu file dengan menampilkan dalam bentuk hexadecimal.
- SpectraPLUS-SC
Aplikasi untuk menganalisis suatu sinyal audio. Terdapat fungsi-fungsi yang tersedia di dalamnya yang membantu memudahkan untuk mencari parameter sinyal yang di inginkan.
- Audacity
Aplikasi pengolah audio digital.

b. Analisis kebutuhan perangkat keras

Spesifikasi perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

- Processor, Intel(R) Core(TM) i5 CPU
- RAM, 4 GB
- Harddisk, 30 GB

c. Analisis kebutuhan pengguna

Aplikasi yang dibangun hanya digunakan oleh *user*. *User* pengirim dan penerima pesan harus memiliki aplikasi ini, agar pesan yang dikirim oleh pengirim dapat diketahui isinya oleh penerima pesan sesuai fungsi dari aplikasi.

4.3 Analisis Kebutuhan Fungsional

Analisa kebutuhan fungsional berisikan proses-proses yang dilakukan oleh sistem. Dalam hal ini *user* sebagai pengguna dapat menggunakan layanan-layanan pada sistem, antara lain :

- a. Melakukan proses enkripsi dan penyisipan pesan teks
- b. Melakukan proses enkripsi dan penyisipan citra
- c. Melakukan proses ekstraksi dan dekripsi pesan teks
- d. Melakukan proses ekstraksi dan dekripsi citra

4.4 Batasan Sistem

Aplikasi ini memiliki batasan pada pembuatannya yaitu sebagai berikut :

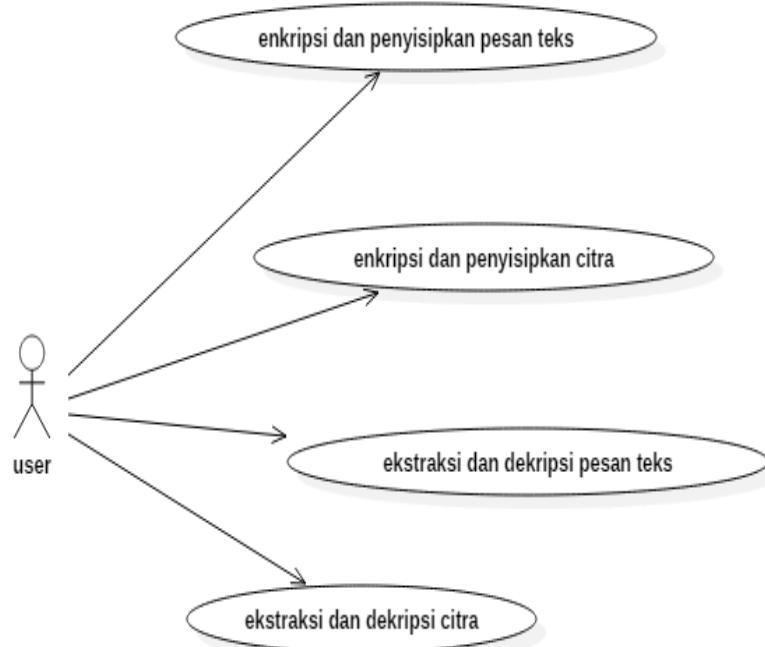
- a. Aplikasi berbasis dekstop.
- b. Masukan informasi rahasia berupa pesan teks dan file citra berekstensi .jpg.
- c. Masukan media pembawa pesan berupa file audio berekstensi .wav.
- d. Menggunakan algoritma kriptografi RC4 dan metode steganografi 2LSB.

4.5 Desain Sistem

Desain sistem digunakan untuk melakukan perancangan sistem dari awal sampai akhir. Aplikasi yang dibangun harus sesuai dengan desain sistem dan teori pendukung yang ada, agar sesuai dengan kebutuhan.

4.5.1 Use Case Diagram

Kebutuhan fungsional sistem dapat digambarkan dengan menggunakan *use case diagram* sebagai berikut :



Gambar 4.1 *Use Case Diagram* Sistem

Pada gambar 4.1 menggambarkan *use case diagram* dengan satu aktor pengguna yaitu *user*. *User* dapat melakukan enkripsi dan penyisipan pesan teks,

enkripsi dan penyisipan citra, ekstraksi dan dekripsi pesan teks, serta ekstraksi dan dekripsi citra. Deskripsi *use case diagram* sistem dapat dilihat pada tabel 4.1.

Tabel 4.1 Deskripsi *Use Case Diagram*

<i>Use Case Name:</i> Enkripsi dan penyisipan pesan teks	<i>ID :</i> UC.01 Siklus 1	<i>Importance Level :</i> <i>High</i>		
<i>Primary Actor :</i> <i>User</i>	<i>Use Case Type :</i>			
<i>Stakeholder and Interest :</i> <i>user</i> melakukan enkripsi dan penyisipan pesan teks				
<i>Brief Description :</i> Menjelaskan tentang enkripsi dan penyisipan pesan teks				
<i>Normal flow events :</i> <ol style="list-style-type: none"> 1. <i>User</i> memasukkan pesan teks yang akan diamankan pada bagian <i>text box “message”</i>. 2. <i>User</i> memasukkan kunci untuk enkripsi pada bagian <i>text box “key”</i>. 3. <i>User</i> memasukkan file audio sebagai pembawa pesan rahasia pada bagian <i>text box “audio file”</i>. 4. <i>User</i> memasukkan nama file keluaran proses pada bagian <i>text box “stego file name”</i>. 5. <i>User</i> menekan tombol <i>“Encrypt”</i>. 6. Sistem melakukan proses enkripsi dan penyisipan pesan teks. 7. File keluaran disimpan pada folder <i>“output”</i>. 				
<i>Alternative flow :</i> <ol style="list-style-type: none"> 1a. Jika pesan teks tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 2a. Jika kunci tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 3a. Jika file audio tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 4a. Jika nama file keluaran tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 6a. Jika terjadi kesalahan dalam proses enkripsi dan penyisipan pesan teks, maka akan muncul notifikasi kesalahan proses. 				

<i>Use Case Name:</i> Enkripsi dan penyisipan citra	<i>ID :</i> UC.02 Siklus 2	<i>Importance Level :</i> <i>High</i>
<i>Primary Actor :</i> <i>User</i>	<i>Use Case Type :</i>	
<i>Stakeholder and Interest :</i> <i>user</i> melakukan enkripsi dan penyisipan citra		

<p><i>Brief Description :</i> Menjelaskan tentang enkripsi dan penyisipan citra</p> <p><i>Normal flow events :</i></p> <ol style="list-style-type: none"> 1. <i>User</i> memasukkan file citra yang akan diamankan pada bagian <i>text box</i> “<i>image file</i>”. 2. <i>User</i> memasukkan kunci untuk enkripsi pada bagian <i>text box</i> “<i>key</i>”. 3. <i>User</i> memasukkan file audio sebagai pembawa pesan rahasia pada bagian <i>text box</i> “<i>audio file</i>”. 4. <i>User</i> memasukkan nama file keluaran proses pada bagian <i>text box</i> “<i>stego file name</i>”. 5. <i>User</i> menekan tombol “<i>Encrypt</i>”. 6. Sistem melakukan proses enkripsi dan penyisipan citra. 7. File keluaran disimpan pada folder “<i>output</i>”. <p><i>Alternative flow :</i></p> <ol style="list-style-type: none"> 1a. Jika file citra tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 2a. Jika kunci tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 3a. Jika file audio tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 4a. Jika nama file keluaran tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 6a. Jika terjadi kesalahan dalam proses enkripsi dan penyisipan citra, maka akan muncul notifikasi kesalahan proses.

<i>Use Case Name:</i> Ekstraksi dan dekripsi pesan teks	<i>ID :</i> UC.03 Siklus 3	<i>Importance Level :</i> <i>High</i>		
<i>Primary Actor :</i> <i>User</i>	<i>Use Case Type :</i>			
<i>Stakeholder and Interest :</i> <i>user</i> melakukan ekstraksi dan dekripsi pesan teks				
<i>Brief Description :</i> Menjelaskan tentang ekstraksi dan dekripsi pesan teks				
<p><i>Normal flow events :</i></p> <ol style="list-style-type: none"> 1. <i>User</i> memasukkan file <i>stego audio</i> pembawa pesan pada bagian <i>text box</i> “<i>stego audio</i>”. 2. <i>User</i> memasukkan kunci untuk dekripsi pada bagian <i>text box</i> “<i>key</i>”. 3. <i>User</i> memasukkan nama file keluaran proses pada bagian <i>text box</i> “<i>output file name</i>”. 4. <i>User</i> menekan tombol “<i>Decrypt</i>”. 5. Sistem melakukan proses ekstraksi dan dekripsi pesan teks. 6. Keluaran ditampilkan pada <i>text box</i> “<i>message</i>” dan disimpan pada folder “<i>output</i>” dengan ekstensi .txt. 				

Alternative flow :

- 1a. Jika file *stego audio* tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan.
- 2a. Jika kunci tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan.
- 3a. Jika nama file keluaran tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan.
- 5a. Jika terjadi kesalahan dalam proses ekstraksi dan dekripsi pesan teks, maka akan muncul notifikasi kesalahan proses.

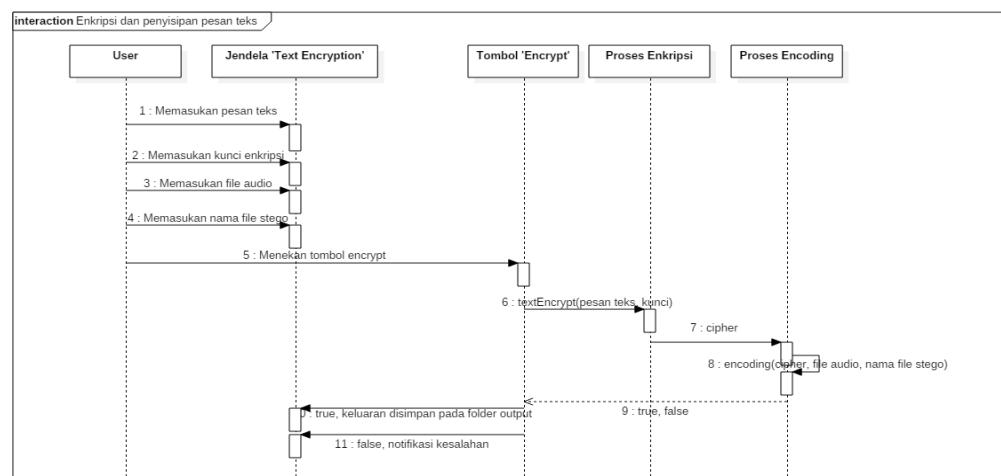
<i>Use Case Name:</i> Ekstraksi dan dekripsi citra	<i>ID :</i> UC.04 Siklus 4	<i>Importance Level :</i> <i>High</i>
<i>Primary Actor :</i> <i>User</i>	<i>Use Case Type :</i>	
<i>Stakeholder and Interest :</i> <i>user</i> melakukan ekstraksi dan dekripsi citra		
<i>Brief Description :</i> Menjelaskan tentang ekstraksi dan dekripsi citra		
<i>Normal flow events :</i>		
<ol style="list-style-type: none"> 1. <i>User</i> memasukkan file <i>stego audio</i> pembawa pesan pada bagian <i>text box</i> “<i>stego audio</i>”. 2. <i>User</i> memasukkan kunci untuk dekripsi pada bagian <i>text box</i> “<i>key</i>”. 3. <i>User</i> memasukkan nama file keluaran proses pada bagian <i>text box</i> “<i>output file name</i>”. 4. <i>User</i> menekan tombol “<i>Decrypt</i>”. 5. Sistem melakukan proses ekstraksi dan dekripsi citra. 6. Keluaran ditampilkan pada <i>picture box</i> “<i>output image</i>” dan disimpan pada folder “<i>output</i>” dengan ekstensi .jpg. 		
<i>Alternative flow :</i>		
<ol style="list-style-type: none"> 1a. Jika file <i>stego audio</i> tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 2a. Jika kunci tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 3a. Jika nama file keluaran tidak dimasukan, maka akan muncul notifikasi untuk melengkapi masukan. 5a. Jika terjadi kesalahan dalam proses ekstraksi dan dekripsi citra, maka akan muncul notifikasi kesalahan proses. 		

4.5.2 Sequence Diagram

Sequence diagram berfungsi untuk mendeskripsikan interaksi entitas dalam sistem. Pada perancangan aplikasi ini terbagi menjadi 4 *sequence diagram*, yaitu enkripsi dan penyisipan pesan teks, enkripsi dan penyisipan citra, ekstraksi dan dekripsi pesan teks, serta ekstraksi dan dekripsi citra.

4.5.2.1 Enkripsi dan penyisipan pesan teks

Gambar 4.2 menunjukkan untuk melakukan enkripsi dan penyisipan pesan teks, *user* harus membuka jendela *Text Encryption*. *User* memerlukan masukan berupa pesan teks, kunci untuk enkripsi, file audio, dan nama file stego. Untuk memulai proses enkripsi dan penyisipan pesan teks, *user* harus menekan tombol *encrypt*. Proses enkripsi pesan teks dilakukan terlebih dahulu, kemudian diikuti proses penyisipan (*encoding*) kedalam file audio. Apabila proses enkripsi dan penyisipan pesan teks berhasil dilakukan, maka keluaran file stego disimpan pada folder *output*. Jika terjadi kesalahan pada proses enkripsi atau *encoding*, maka akan muncul notifikasi kesalahan proses.

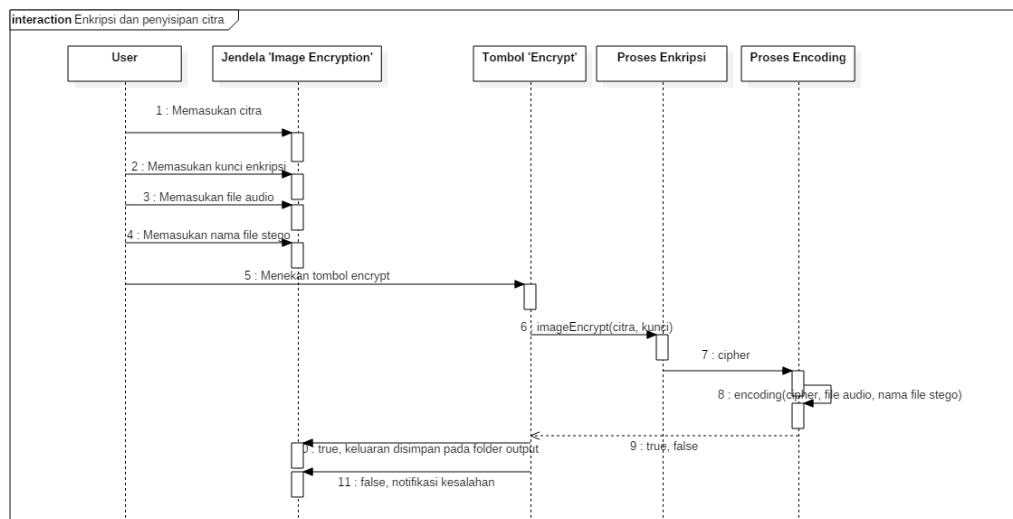


Gambar 4.2 *Sequence Diagram* Enkripsi dan Penyisipan Pesan Teks

4.5.2.2 Enkripsi dan penyisipan citra

Gambar 4.3 menunjukkan untuk melakukan enkripsi dan penyisipan citra, *user* harus membuka jendela *Image Encryption*. *User* memerlukan masukan berupa citra, kunci untuk enkripsi, file audio, dan nama file stego. Untuk memulai proses enkripsi dan penyisipan citra, *user* harus menekan tombol *encrypt*. Proses enkripsi citra dilakukan terlebih dahulu, kemudian diikuti proses penyisipan (*encoding*) kedalam file audio. Apabila proses enkripsi dan penyisipan citra berhasil dilakukan,

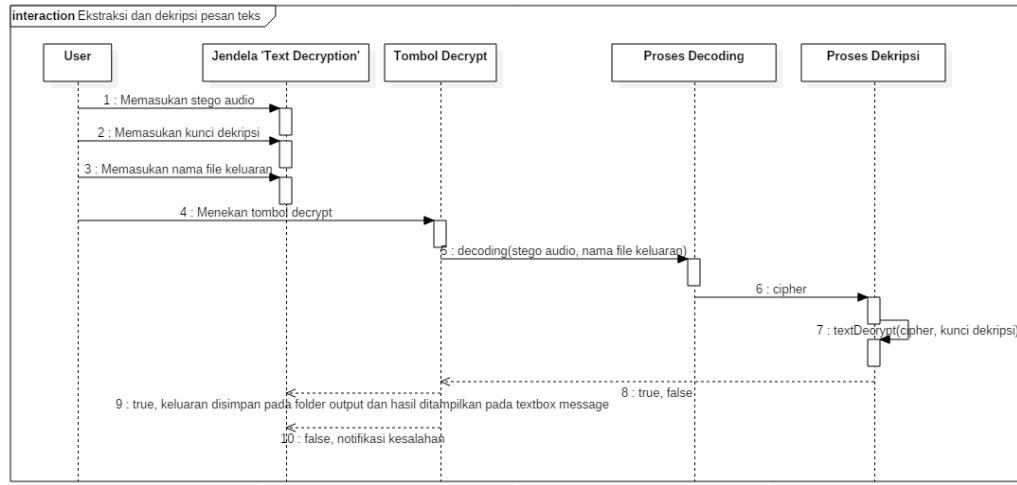
maka keluaran file stego disimpan pada folder *output*. Jika terjadi kesalahan pada proses enkripsi atau *encoding*, maka akan muncul notifikasi kesalahan proses.



Gambar 4.3 Sequence Diagram Enkripsi dan Penyisipan Citra

4.5.2.3 Ekstraksi dan dekripsi pesan teks

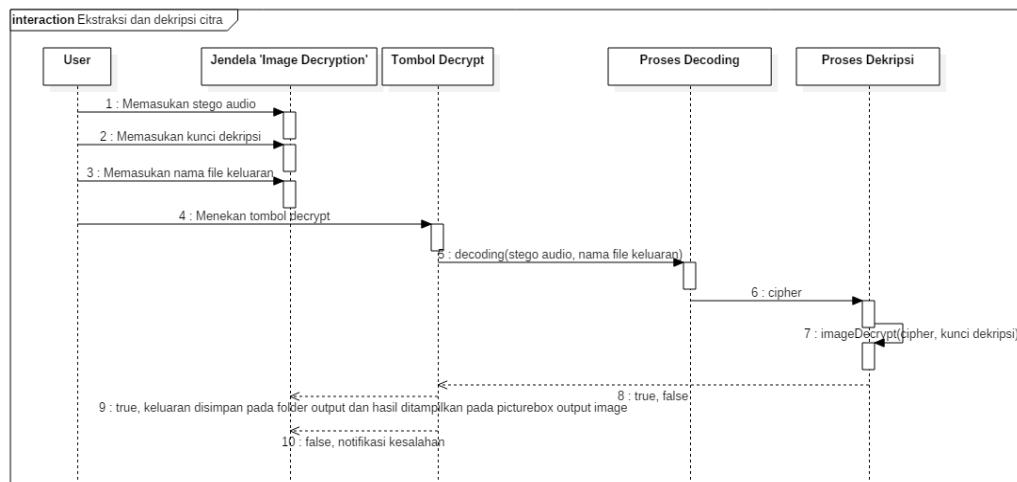
Gambar 4.4 menunjukkan untuk melakukan ekstraksi dan dekripsi pesan teks, *user* harus membuka jendela *Text Decryption*. *User* memerlukan masukan berupa stego audio, kunci dekripsi, dan nama file keluaran. Untuk memulai proses ekstraksi dan dekripsi pesan teks, *user* harus menekan tombol *decrypt*. Proses ekstraksi (*decoding*) dilakukan terlebih dahulu, kemudian diikuti proses dekripsi pesan teks. Apabila proses ekstraksi dan dekripsi pesan teks berhasil dilakukan, maka keluaran ditampilkan pada *textbox message* dan file keluaran disimpan pada folder *output* dengan ekstensi .txt. Jika terjadi kesalahan pada proses ekstraksi dan dekripsi pesan teks, maka akan muncul notifikasi kesalahan proses.



Gambar 4.4 Sequence Diagram Ekstraksi dan Dekripsi Pesan Teks

4.5.2.4 Ekstraksi dan dekripsi citra

Gambar 4.5 menunjukkan untuk melakukan ekstraksi dan dekripsi citra, *user* harus membuka jendela *Image Decryption*. *User* memerlukan masukan berupa stego audio, kunci dekripsi, dan nama file keluaran. Untuk memulai proses ekstraksi dan dekripsi citra, *user* harus menekan tombol *decrypt*. Proses ekstraksi (*decoding*) dilakukan terlebih dahulu, kemudian diikuti proses dekripsi citra. Apabila proses ekstraksi dan dekripsi citra berhasil dilakukan, maka keluaran ditampilkan pada *picturebox output image* dan file keluaran disimpan pada folder *output* dengan ekstensi .jpg. Jika terjadi kesalahan pada proses ekstraksi dan dekripsi citra, maka akan muncul notifikasi kesalahan proses.



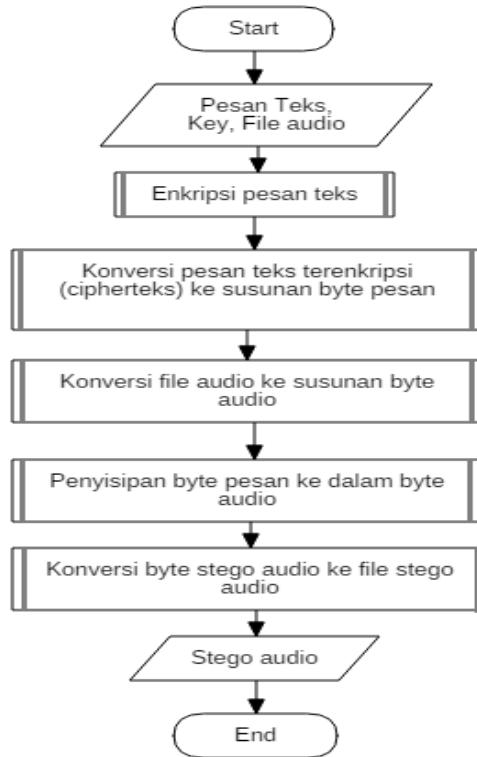
Gambar 4.5 Sequence Diagram Ekstraksi dan Dekripsi Citra'

4.5.3 Flowchart Diagram

Flowchart diagram berguna untuk menunjukkan langkah-langkah atau prosedur-prosedur suatu sistem yang dibangun.

4.5.3.1 Enkripsi dan penyisipan pesan teks

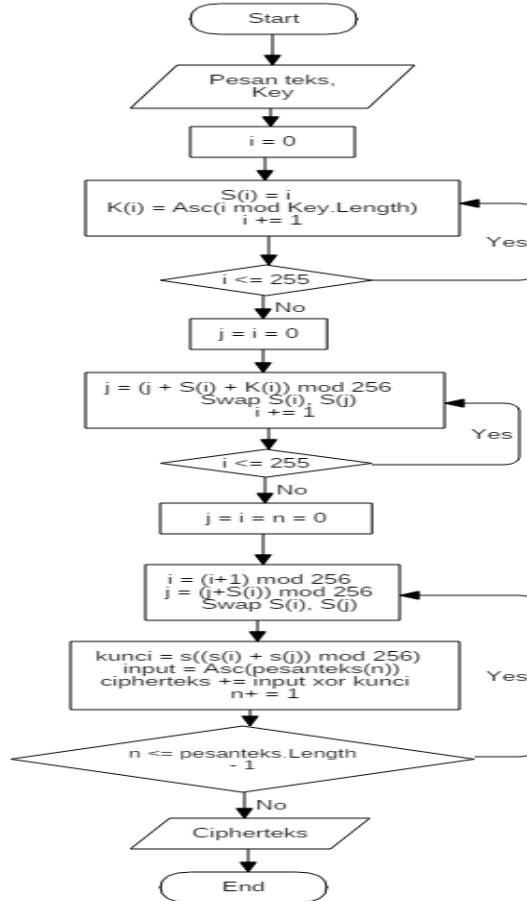
Berikut adalah *flowchart diagram* proses enkripsi dan penyisipan pesan teks.



Gambar 4.6 *Flowchart Diagram* Enkripsi dan Penyisipan Pesan Teks

Pada gambar 4.6 menjelaskan tentang enkripsi dan penyisipan pesan teks, memerlukan masukan berupa pesan teks yang diamankan, key, dan file audio pembawa pesan. Proses pertama yang dilakukan yaitu, enkripsi pesan teks. Proses ini membutuhkan pesan teks dan key sebagai masukan proses, yang kemudian menghasilkan nilai keluaran berupa cipherteks. Proses berikutnya cipherteks dikonversi ke dalam susunan *byte* pesan yang selanjutnya disisipkan kedalam susunan *byte* audio pembawa pesan, sehingga menghasilkan nilai keluaran berupa *byte* stego. *Byte* stego yang didapatkan dari proses penyisipan dikonversi ke bentuk file audio sehingga menghasilkan keluaran berupa file stego audio.

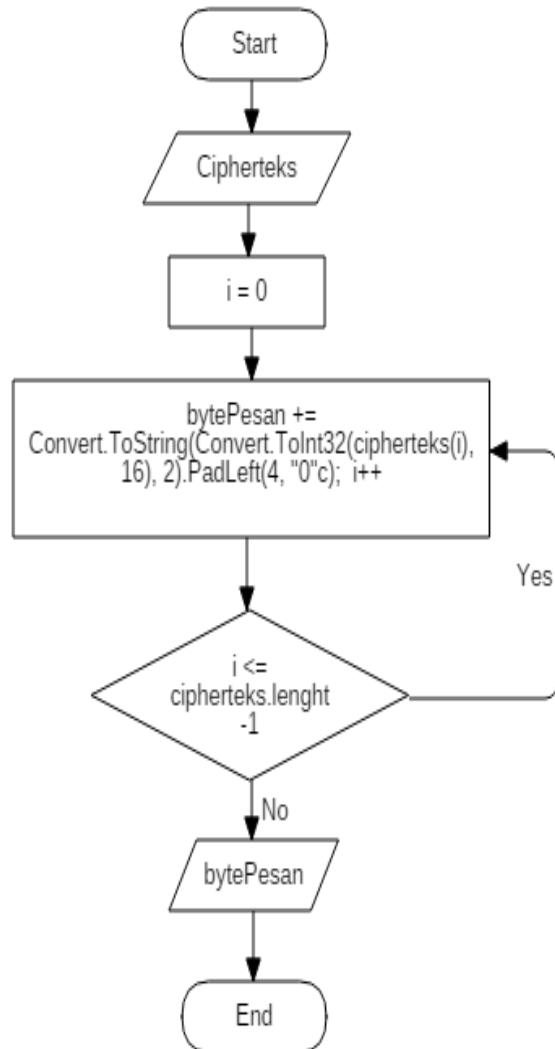
a. Enkripsi pesan teks menggunakan algoritma RC4



Gambar 4.7 Flowchart Diagram Enkripsi Pesan Teks Menggunakan Algoritma RC4

Pada gambar 4.7 menjelaskan tentang proses enkripsi pesan teks menggunakan algoritma RC4. Dibutuhkan masukan berupa pesan teks yang diamankan dan key enkripsi. Tahapan enkripsi pesan teks dimulai dari menginisialisasi array Sbox dan state-array K sampai iterasi ke 256. Selanjutnya dilakukan KSA (*key-scheduling algorithm*) yaitu pengacakan nilai array Sbox sampai iterasi ke 256. Setelah melakukan KSA, tahapan enkripsi pesan selanjutnya ialah PRGA (*Pseudo Random Generation Algorithm*) untuk menghasilkan kunci enkripsi yang akan di XOR kan pada pesan teks masukan, sehingga menghasilkan keluaran berupa pesan teks yang terenkripsi (cipherteks). PRGA dilakukan iterasi sebanyak panjangnya karakter teks masuk.

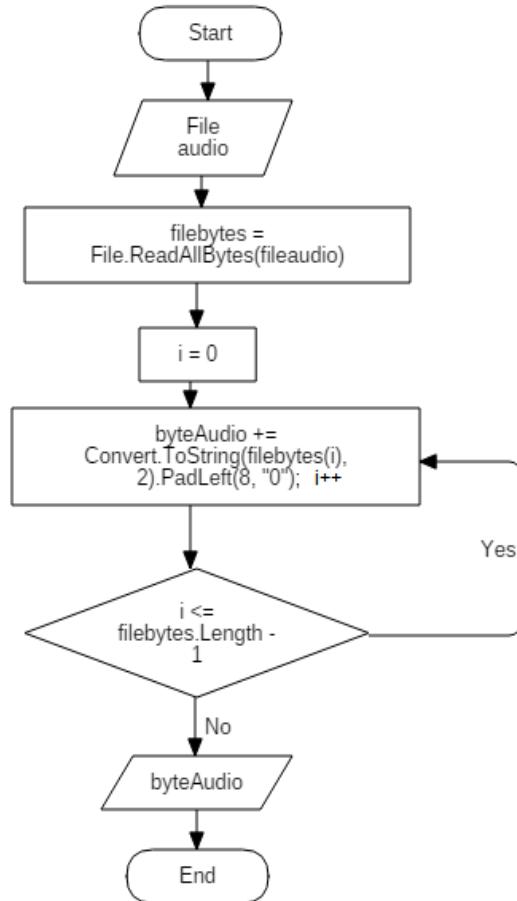
b. Konversi cipherteks ke susunan *byte* pesan



Gambar 4.8 *Flowchart Diagram* Konversi Cipherteks ke Susunan *Byte Pesan*

Pada gambar 4.8 menjelaskan tentang proses konversi cipherteks menjadi *byte* pesan , dengan nilai chiperteks yang diproses tiap karakter pada nilai masukkan dengan perulangan sesuai panjang chiperteks. Nilai kembalian dari proses ini berupa *byte* pesan.

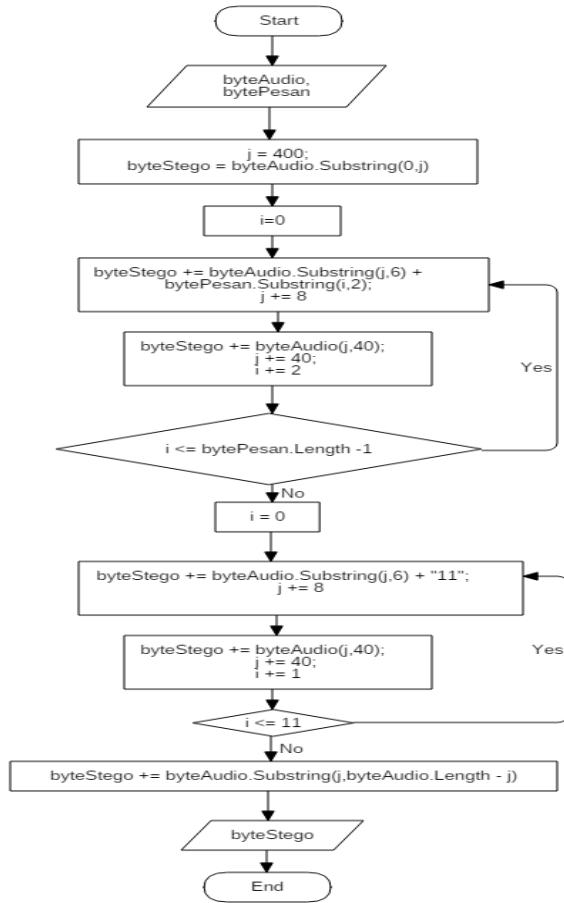
c. Konversi file audio ke susunan *byte* audio



Gambar 4.9 *Flowchart Diagram* Konversi File Audio ke Susunan *Byte* Audio

Pada gambar 4.9 menjelaskan proses konversi file audio ke susunan *byte* audio, menerima masukan berupa file audio yang kemudian dilakukan pembacaan tiap *byte* file audio. Hasil dari pembacaan *byte* tersebut kemudian disimpan kedalam variabel array dengan panjang dari file audio yang di proses. Selanjutnya dikonversi kedalam bilangan biner sehingga didapatkan kembalian dari proses berupa susunan *byte* audio.

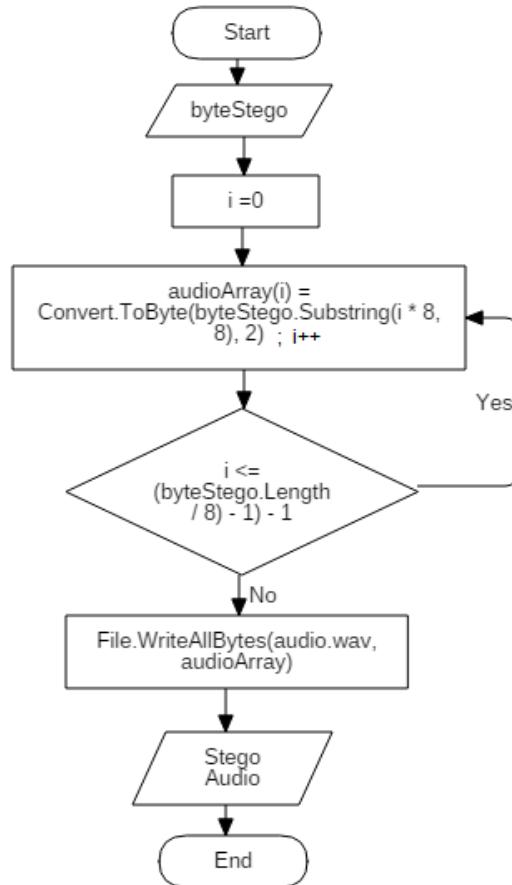
d. Penyisipan *byte* pesan ke dalam *byte* audio menggunakan metode 2LSB



Gambar 4.10 Flowchart Diagram Penyisipan Byte Pesan ke dalam Byte Audio Menggunakan Metode 2LSB

Gambar 4.10 menjelaskan proses penyisipan *byte* pesan ke dalam *byte* audio menggunakan metode 2LSB membutuhkan nilai masukan berupa *byte* pesan dan *byte* audio. Proses penyisipan dimulai pada *byte* ke 51. Penyisipan pesan dilakukan dengan mengganti 2 bit terakhir setiap *byte* data audio dengan *byte* pesan. Setiap penyisipan pesan dilakukan dengan memberikan jarak 5 *byte* dengan penyisipan berikutnya. Banyaknya iterasi penyisipan bergantung pada panjang *byte* pesan yang disisipkan. Setelah penyisipan pesan selesai, maka dilakukan penyisipan bit penutup dengan nilai '11' dan dilakukan iterasi sebanyak 12 kali. *Byte* audio yang disisipi pesan seluruhnya disimpan ke dalam *byte* stego yang merupakan nilai kembalian dari proses ini.

e. Konversi *byte* stego audio ke file stego audio

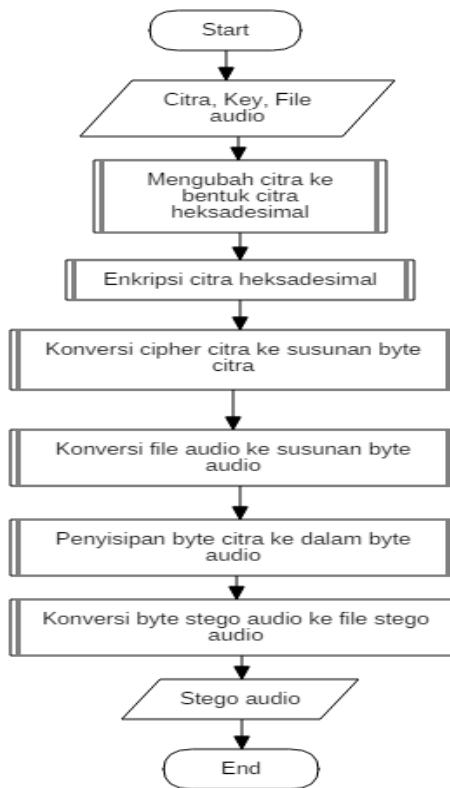


Gambar 4.11 *Flowchart Diagram* Konversi *Byte* Stego Audio ke File Stego Audio

Pada gambar 4.11 menjelaskan tentang konversi *byte* stego menghasilkan file stego audio, dengan masukan berupa *byte* dari stego. Proses yang dilakukan pertama yaitu dengan mengubah semua *byte* dari stego menjadi *audioArray* yang memiliki tipe data *byte*. Dari *audioArray* tersebut dijalankan method `file.writeAllBytes()`, dengan parameter berupa nama file keluaran dan variabel *audioArray*, sehingga dihasilkan kembalian berupa file audio yang telah dibentuk yang berada pada folder yang telah ditentukan.

4.5.3.2 Enkripsi dan penyisipan citra

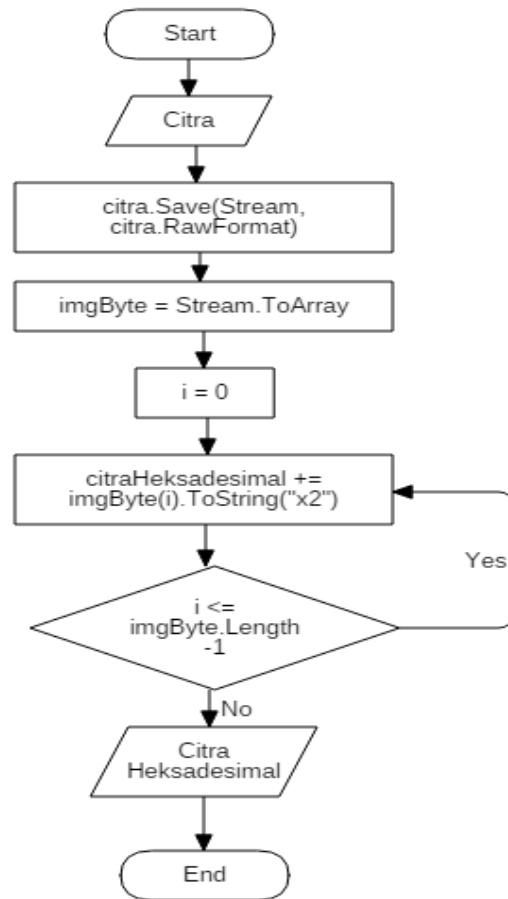
Berikut adalah *flowchart diagram* proses enkripsi dan penyisipan citra.



Gambar 4.12 *Flowchart Diagram* Enkripsi dan Penyisipan Citra

Pada gambar 4.12 menjelaskan tentang enkripsi dan penyisipan citra, memerlukan masukan berupa citra yang diamankan, key, dan file audio pembawa pesan. Proses pertama yang dilakukan yaitu, enkripsi citra. Proses ini membutuhkan citra heksadesimal dan key sebagai masukan proses, yang kemudian menghasilkan nilai keluaran berupa cipher citra. Proses berikutnya cipher citra dikonversi ke dalam susunan *byte* pesan yang selanjutnya disisipkan kedalam susunan *byte* audio pembawa pesan, sehingga menghasilkan nilai keluaran berupa *byte* stego. *Byte* stego yang didapatkan dari proses penyisipan dikonversi ke bentuk file audio sehingga menghasilkan keluaran berupa file stego audio.

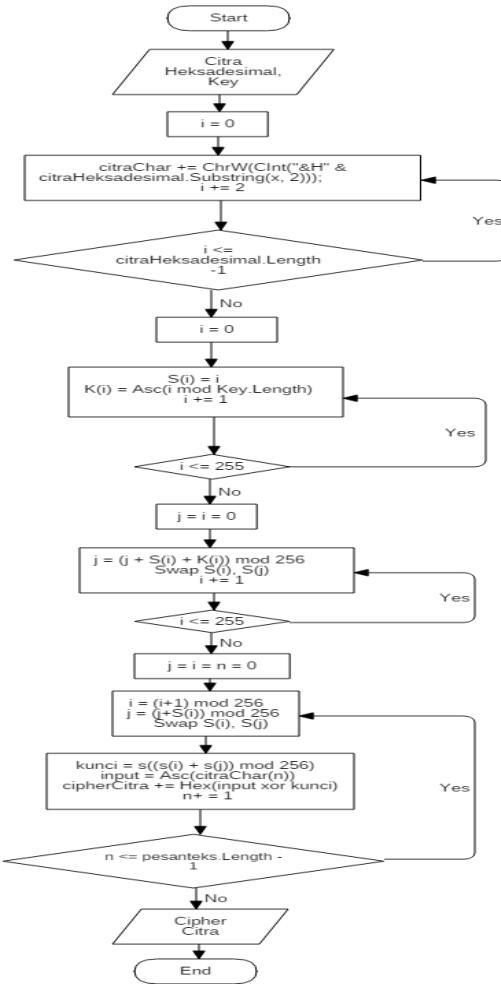
a. Mengubah citra ke bentuk citra heksadesimal



Gambar 4.13 *Flowchart Diagram* Mengubah Citra ke bentuk Citra Heksadesimal

Gambar 4.13 menunjukkan proses pengubahan dengan masukan berupa file citra menjadi bentuk citra heksadesimal, pada proses yang dilakukan adalah melakukan pengubahan citra menjadi *byte*, kemudian dilakukan pembacaan tiap indeks *byte* dan diubah menjadi heksadesimal sesuai dengan panjang dari *byte* tersebut. Sehingga didapatkan nilai kembalian berupa citra heksadesimal.

b. Enkripsi citra heksadesimal menggunakan algoritma RC4

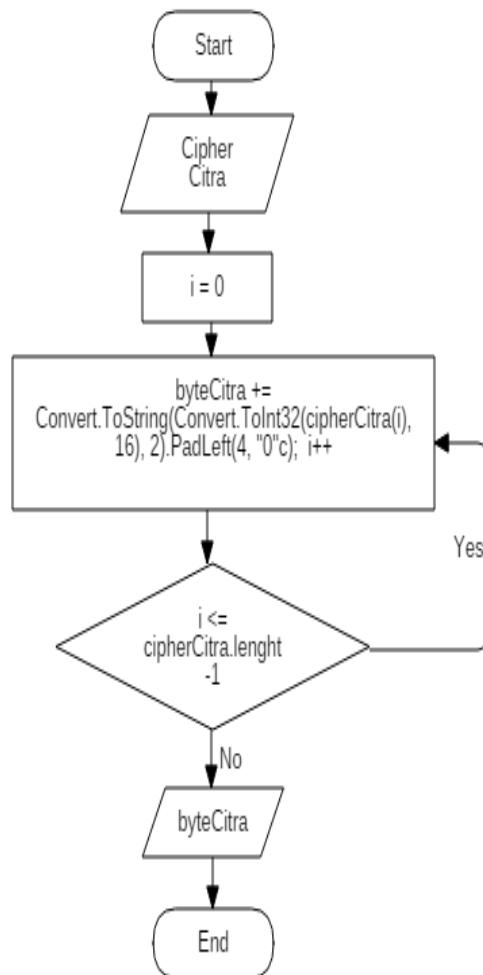


Gambar 4.14 Flowchart Diagram Enkripsi Citra Heksadesimal Menggunakan Algoritma RC4

Pada gambar 4.14 menjelaskan tentang proses enkripsi citra heksadesimal menggunakan algoritma RC4. Dibutuhkan masukan berupa citra heksadesimal dan key enkripsi. Citra heksadesimal masukan diubah terlebih dahulu ke bentuk karakter. Tahapan enkripsi dimulai dari menginisialisasi array Sbox dan state-array K sampai iterasi ke 256. Selanjutnya dilakukan KSA (*key-scheduling algorithm*) yaitu pengacakan nilai array Sbox sampai iterasi ke 256. Setelah melakukan KSA, tahapan selanjutnya ialah PRGA (*Pseudo Random Generation Algorithm*) untuk menghasilkan kunci enkripsi yang akan di XOR kan pada citra masukan.

Hasil keluaran diubah kembali ke bentuk heksadesimal dan menghasilkan keluaran berupa cipher citra. PRGA dilakukan iterasi sebanyak panjangnya karakter teks masuk.

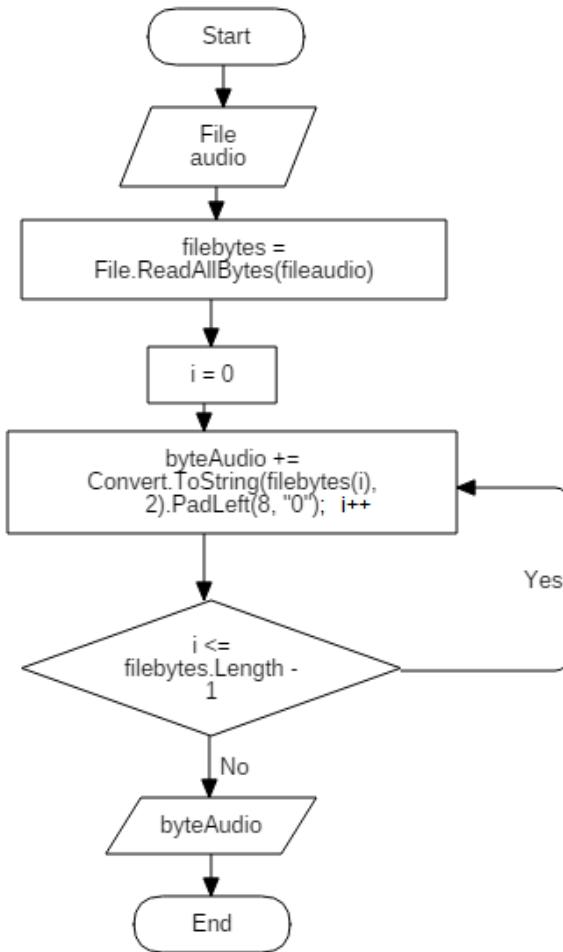
c. Konversi cipher citra ke susunan *byte* citra



Gambar 4.15 Flowchart Diagram Konversi Cipher Citra ke Byte Citra

Pada gambar 4.15 menunjukkan proses dari konversi cipher citra ke susunan *byte* citra. Masukan berupa chiper citra yang dilakukan proses konversi ke dalam bilangan biner sehingga menghasilkan nilai kembalian proses berupa susunan *byte* citra.

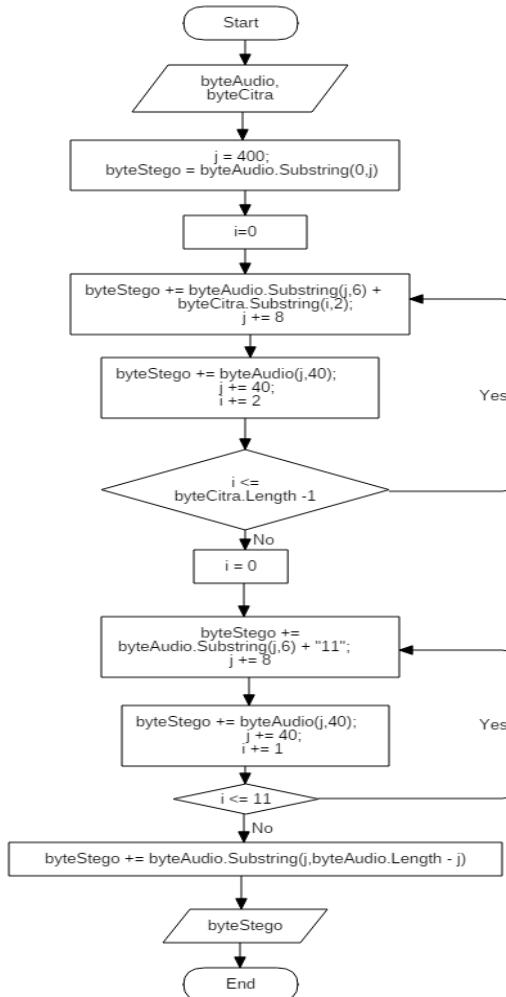
d. Konversi file audio ke susunan *byte* audio



Gambar 4.16 *Flowchart Diagram* Konversi File Audio ke *Byte* Audio

Gambar 4.16 menjelaskan proses konversi file audio ke susunan *byte* audio, menerima masukan berupa file audio yang kemudian dilakukan pembacaan tiap *byte* file audio. Hasil dari pembacaan *byte* tersebut kemudian disimpan kedalam variabel array dengan panjang dari file audio yang di proses. Selanjutnya dikonversi kedalam bilangan biner sehingga didapatkan kembalian dari proses berupa susunan *byte* audio.

e. Penyisipan *byte* citra ke dalam *byte* audio menggunakan metode 2LSB

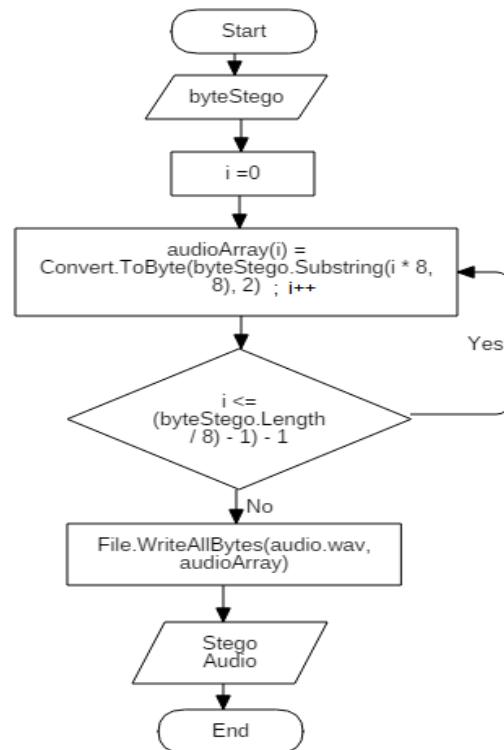


Gambar 4.17 *Flowchart Diagram* Penyisipan *Byte* Citra kedalam *Byte* Audio Menggunakan Metode 2LSB

Gambar 4.17 menjelaskan proses penyisipan *byte* citra ke dalam *byte* audio menggunakan metode 2LSB membutuhkan nilai masukan berupa *byte* citra dan *byte* audio. Proses penyisipan dimulai pada *byte* ke 51. Penyisipan dilakukan dengan mengganti 2 bit terakhir setiap *byte* data audio dengan *byte* pesan. Setiap penyisipan pesan dilakukan dengan memberikan jarak 5 *byte* dengan penyisipan berikutnya. Banyaknya iterasi penyisipan bergantung pada panjang *byte* citra yang disisipkan. Setelah penyisipan citra selesai, maka dilakukan penyisipan bit penutup dengan nilai ‘11’ dan dilakukan

iterasi sebanyak 12 kali. *Byte* audio yang disisipi citra seluruhnya disimpan ke dalam *byte stego* yang merupakan nilai kembalian dari proses ini.

f. Konversi *byte* stego audio ke file stego audio



Gambar 4.18 *Flowchart Diagram* Konversi *Byte* Stego Audio ke File Stego Audio

Pada gambar 4.18 menjelaskan tentang konversi *byte* stego menghasilkan file stego audio, dengan masukan berupa *byte* dari stego. Proses yang dilakukan pertama yaitu dengan mengubah semua *byte* dari stego menjadi *audioArray* yang memiliki tipe data *byte*. Dari *audioArray* tersebut dijalankan method *file.writeAllBytes()*, dengan parameter berupa nama file keluaran dan variabel *audioArray*, sehingga dihasilkan kembalian berupa file audio yang telah dibentuk yang berada pada folder yang telah ditentukan.

4.5.3.3 Dekripsi dan ekstraksi pesan teks

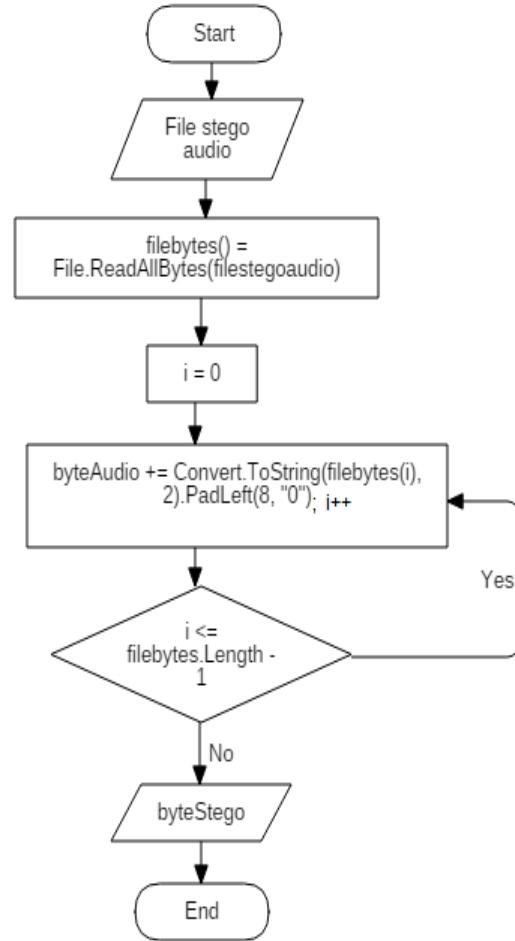
Berikut adalah *flowchart diagram* proses dekripsi dan ekstraksi pesan teks.



Gambar 4.19 *Flowchart Diagram* Dekripsi dan Ekstraksi Pesan Teks

Pada gambar 4.19 menjelaskan tentang dekripsi dan ekstraksi pesan teks, memerlukan masukan berupa file stego audio dan key dekripsi. Proses pertama yang dilakukan yaitu, mengubah file stego audio ke susunan *byte* stego. Selanjutnya melakukan ekstraksi *byte* pesan yang terdapat pada *byte* stego audio. *Byte* pesan tersebut kemudian dikonversi menjadi suatu cipherteks yang diperlukan proses dekripsi menggunakan key dekripsi tertentu, sehingga menghasilkan keluaran berupa pesan teks yang dapat dimengerti maksudnya.

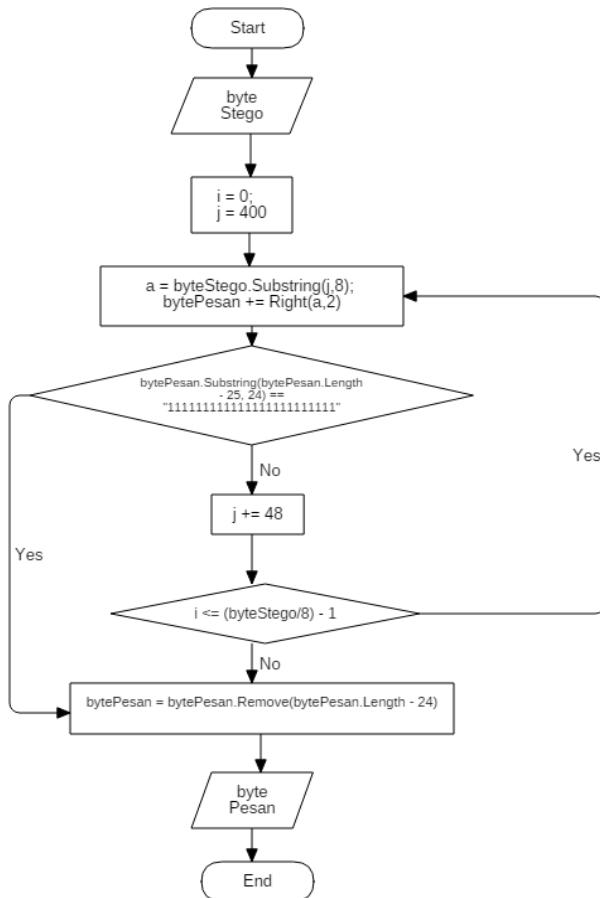
a. Mengubah file stego audio ke susunan *byte* stego



Gambar 4.20 *Flowchart Diagram* Mengubah File Stego Audio ke Susunan *Byte* Stego

Gambar 4.20 menjelaskan proses konversi file stego audio ke susunan *byte* stego, menerima masukan berupa file stego audio yang kemudian dilakukan pembacaan tiap *byte* file stego audio. Hasil dari pembacaan *byte* tersebut kemudian disimpan kedalam variabel array dengan panjang dari file stego audio yang di proses. Selanjutnya dikonversi kedalam bilangan biner sehingga didapatkan kembalian dari proses berupa susunan *byte* stego.

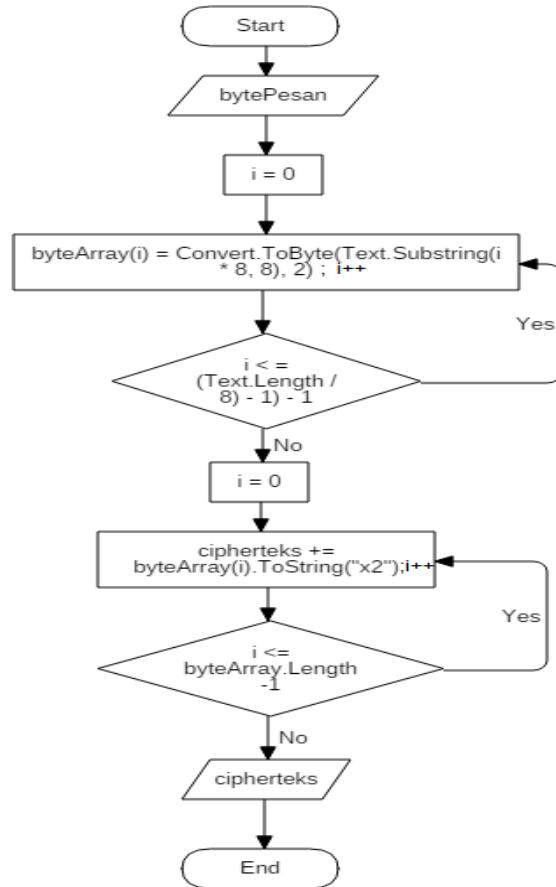
b. Ekstraksi *byte* pesan dari *byte* stego menggunakan metode 2LSB



Gambar 4.21 *Flowchart Diagram* Ekstraksi *Byte* Pesan dari *Byte* Stego Menggunakan Metode 2LSB

Gambar 4.21 menjelaskan untuk melakukan proses ekstraksi *byte* pesan dari *byte* stego menggunakan metode steganografi 2LSB, diperlukan masukan berupa *byte* stego. Langkah pertama menginisialisasi *byte* pesan dengan memasukkan 2 bit terakhir dari *byte* stego audio. Sama seperti proses penyisipan, pengambilan bit pesan dimulai pada *byte* stego ke 51 dan memberikan jarak 5 *byte* dengan pengambilan bit berikutnya. Pada saat melakukan pengambilan bit pesan, dilakukan seleksi kondisi apabila bit pesan yang diambil berupa susunan bit dengan panjang 24 bit memiliki nilai '111111111111111111111111', maka langkah pengambilan *byte* pesan telah selesai, sehingga didapatkan kembalian berupa *byte* pesan.

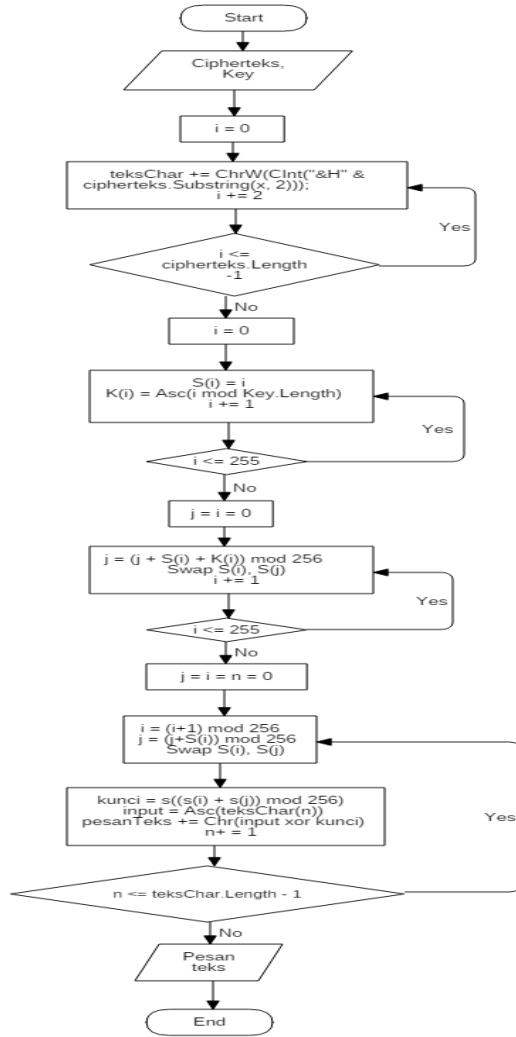
c. Konversi *byte* pesan menjadi cipherteks



Gambar 4.22 Flowchart Diagram Konversi *Byte* Pesan menjadi Cipherteks

Gambar 4.22 menjelaskan untuk melakukan proses konversi *byte* pesan menjadi cipherteks memerlukan masukan berupa *byte* pesan. *Byte* pesan selanjutnya dikonversi dan disimpan pada variabel byteArray yang memiliki tipe data *byte*. Langkah berikutnya mengubah byteArray kedalam bentuk heksadesimal dan didapatkan suatu cipherteks yang perlu di dekripsi untuk mengetahui maknanya.

d. Dekripsi cipherteks menggunakan algoritma RC4

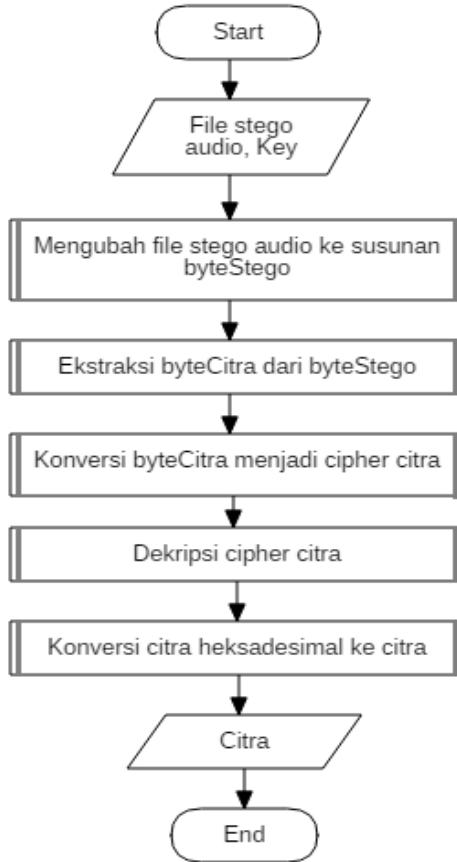


Gambar 4.23 *Flowchart Diagram Dekripsi Cipherteks Menggunakan Algoritma RC4*

Gambar 4.23 menjelaskan proses dekripsi cipherteks menggunakan algoritma RC4, memerlukan masukan berupa cipherteks dan key dekripsi. Cipherteks diubah terlebih dahulu kedalam bentuk karakter. Kemudian dilakukan tahap dekripsi yang sama seperti tahap yang dilakukan ketika enkripsi, yaitu inisialisasi Sbox dan state array K, KSA, dan PRGA. Dari tahap tersebut didapatkan pesan teks yang telah terdekripsi yang dapat dimengerti maknanya.

4.5.3.4 Dekripsi dan ekstraksi citra

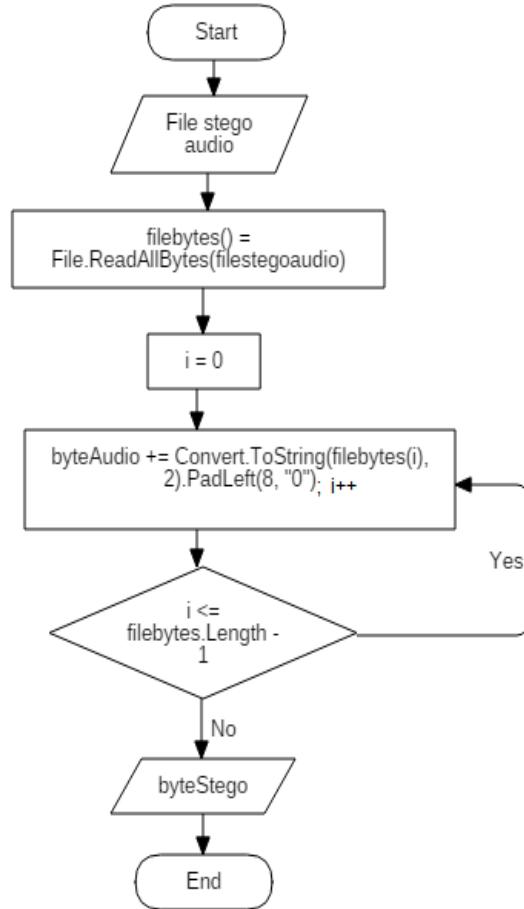
Berikut adalah *flowchart diagram* proses dekripsi dan ekstraksi citra.



Gambar 4.24 *Flowchart Diagram* Dekripsi dan Ekstraksi Citra

Pada gambar 4.24 menjelaskan tentang dekripsi dan ekstraksi citra, memerlukan masukan berupa file stego audio dan key dekripsi. Proses pertama yang dilakukan yaitu, mengubah file stego audio ke susunan *byte* stego. Selanjutnya melakukan ekstraksi *byte* citra yang terdapat pada *byte* stego audio. *Byte* citra tersebut kemudian dikonversi menjadi suatu *cipher* citra yang diperlukan proses dekripsi menggunakan key dekripsi tertentu, sehingga menghasilkan keluaran berupa citra heksadesimal yang selanjutnya dikonversi ke bentuk file citra kembali.

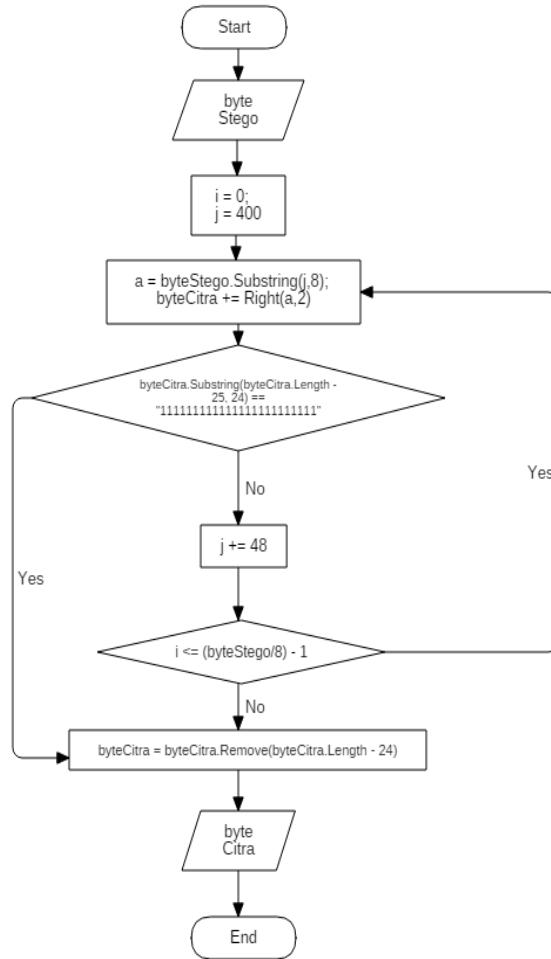
a. Mengubah file stego audio ke susunan *byte* stego



Gambar 4.25 *Flowchart Diagram* Mengubah File Stego ke Susunan *Byte Stego*

Gambar 4.25 menjelaskan proses konversi file stego audio ke susunan *byte* stego, menerima masukan berupa file stego audio yang kemudian dilakukan pembacaan tiap *byte* file stego audio. Hasil dari pembacaan *byte* tersebut kemudian disimpan kedalam variabel array dengan panjang dari file stego audio yang di proses. Selanjutnya dikonversi kedalam bilangan biner sehingga didapatkan kembalian dari proses berupa susunan *byte* stego.

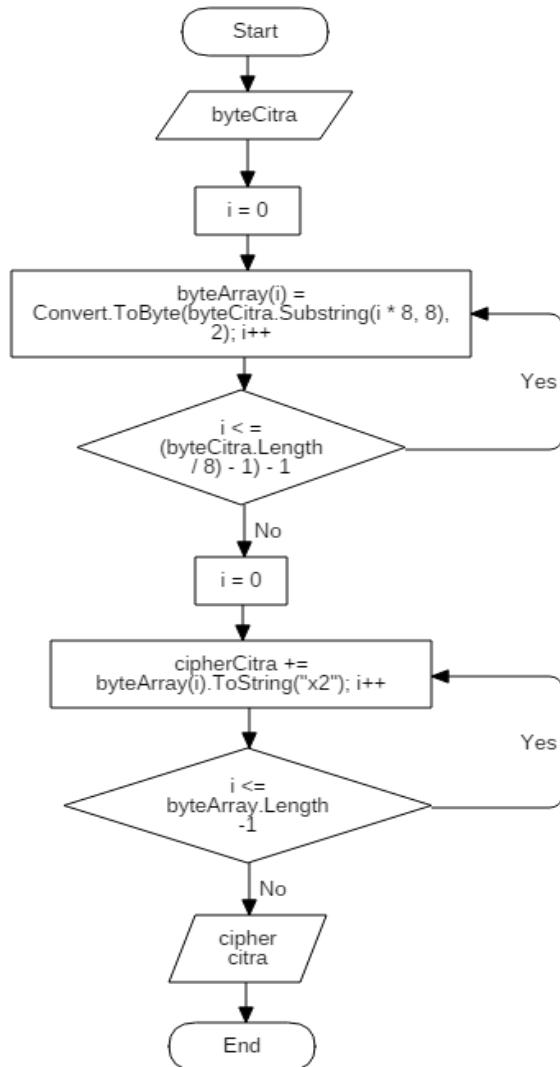
b. Ekstraksi *byte* citra dari *byte* stego menggunakan metode 2LSB



Gambar 4.26 *Flowchart Diagram* Ekstraksi *Byte* Citra dari *Byte* Stego
Menggunakan Metode 2LSB

Gambar 4.26 menjelaskan untuk melakukan proses ekstraksi *byte* citra dari *byte* stego menggunakan metode steganografi 2LSB, diperlukan masukan berupa *byte* stego. Langkah pertama menginisialisasi *byte* citra dengan memasukkan 2 bit terakhir dari *byte* stego audio. Sama seperti proses penyisipan, pengambilan bit citra dimulai pada *byte* stego ke 51 dan memberikan jarak 5 *byte* dengan pengambilan bit berikutnya. Pada saat melakukan pengambilan bit citra, dilakukan seleksi kondisi apabila bit citra yang diambil berupa susunan bit dengan panjang 24 bit memiliki nilai ‘111111111111111111111111’, maka langkah pengambilan *byte* citra telah selesai, sehingga didapatkan kembalian berupa *byte* citra.

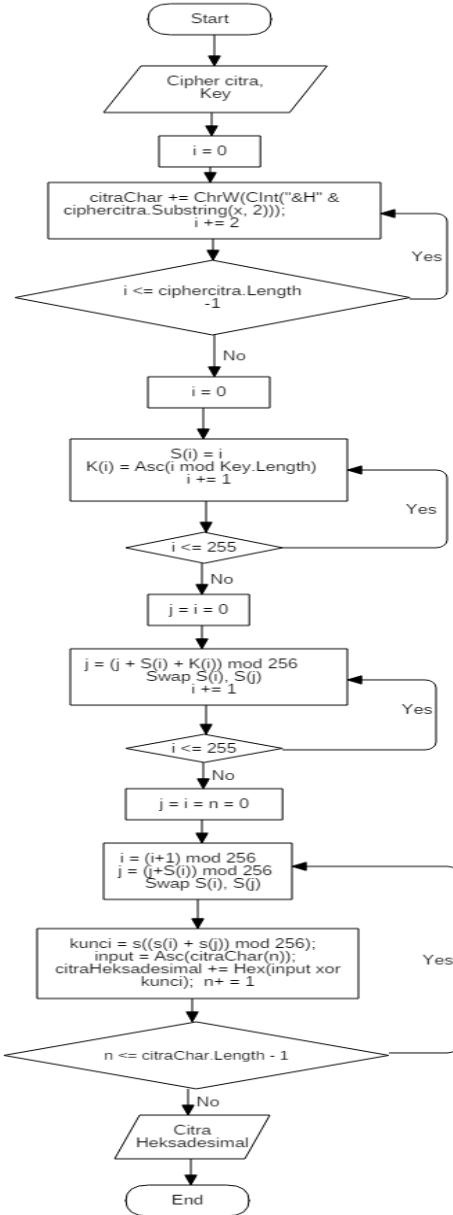
c. Konversi *byte* citra menjadi cipher citra



Gambar 4.27 *Flowchart Diagram* Konversi *Byte* Citra Menjadi *Cipher* Citra

Gambar 4.27 menjelaskan untuk melakukan proses konversi *byte* citra menjadi *cipher* citra memerlukan masukan berupa *byte* citra. *Byte* citra selanjutnya dikonversi dan disimpan pada variabel *byteArray* yang memiliki tipe data *byte*. Langkah berikutnya mengubah *byteArray* kedalam bentuk heksadesimal dan didapatkan suatu *cipher* citra yang diperlukan untuk proses dekripsi citra.

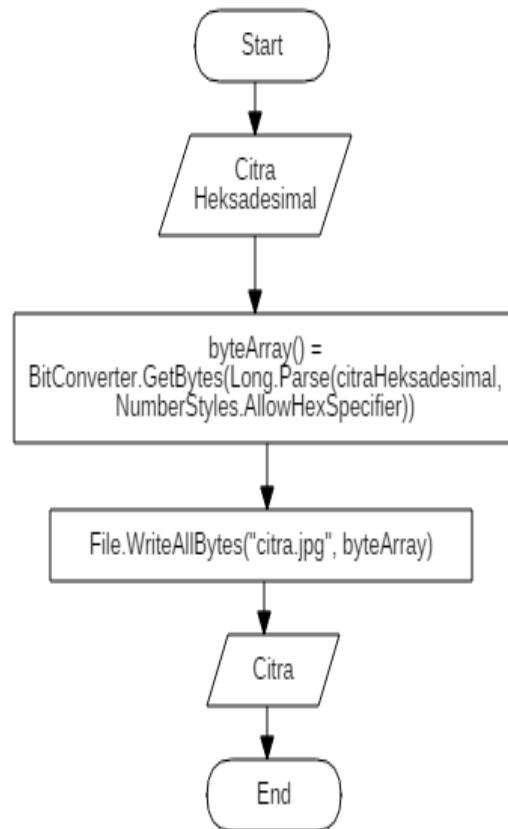
d. Dekripsi cipher citra menggunakan algoritma RC4



Gambar 4.28 *Flowchart Diagram* Dekripsi Cipher Citra Menggunakan Algoritma RC4

Gambar 4.28 menjelaskan proses dekripsi cipher citra menggunakan algoritma RC4, memerlukan masukan berupa cipher citra dan key dekripsi. Cipher citra diubah terlebih dahulu kedalam bentuk karakter. Kemudian dilakukan tahap dekripsi yang sama seperti tahap yang dilakukan ketika enkripsi, yaitu inisialisasi Sbox dan state array K, KSA, dan PRGA. Dari tahap tersebut didapatkan kembalian berupa citra heksadesimal.

e. Konversi citra heksadesimal ke file citra

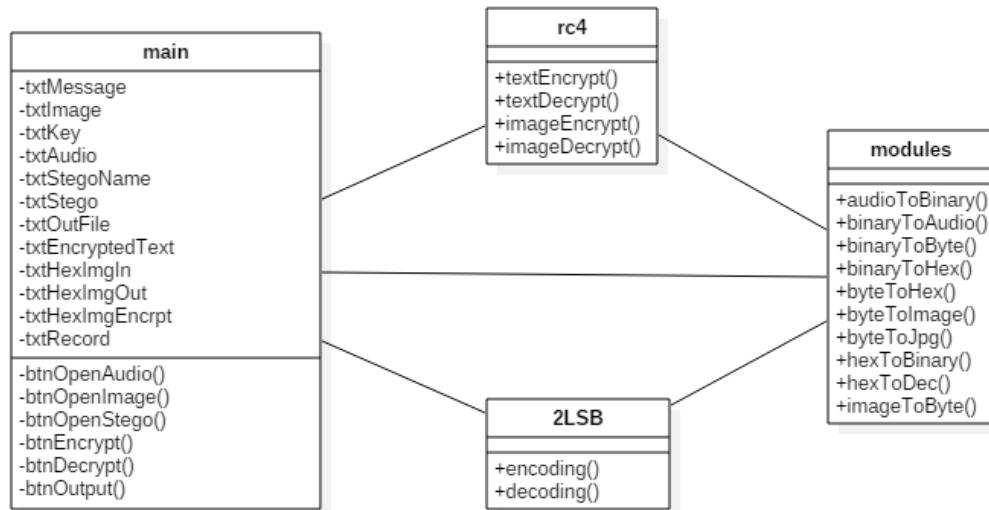


Gambar 4.29 *Flowchart Diagram* Konversi Citra Heksadesimal ke File Citra

Gambar 4.29 menjelaskan konversi citra heksadesimal ke file citra, memerlukan masukan berupa citra heksadesimal. Citra heksadesimal diubah dan disimpan pada variabel byteArray yang memiliki tipe data *byte*. Selanjutnya dilakukan proses membentuk file citra dengan menggunakan fungsi `File.WriteAllBytes` dengan memasukkan parameter nama file penyimpanan dan variabel byteArray. Sehingga didapatkan keluaran file citra yang berada pada folder yang telah ditentukan.

4.5.4 Class Diagram

Class diagram berfungsi untuk menampilkan kelas-kelas yang ada pada sistem dan hubungannya secara logika.



Gambar 4.30 *Class Diagram* Sistem

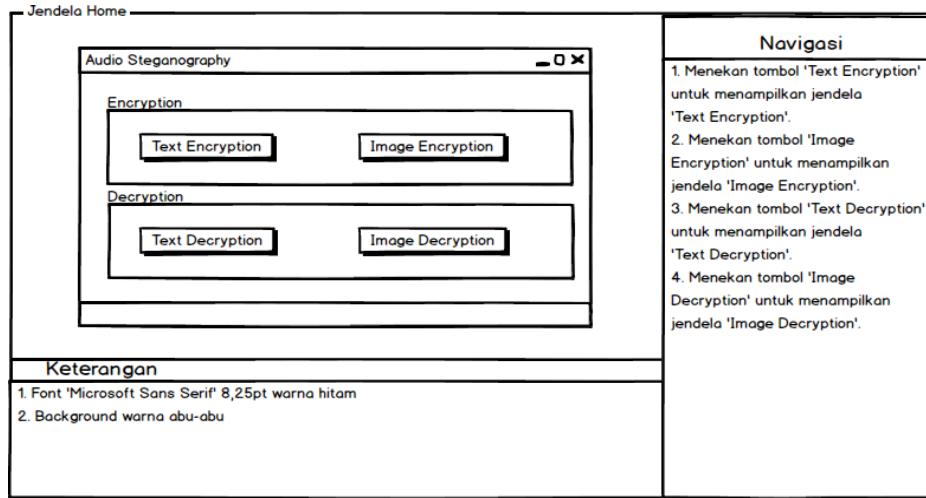
Gambar 4.30 menunjukkan aplikasi ini memiliki 4 kelas, yaitu kelas *main*, *modules*, *rc4*, dan *2 LSB*. Kelas *main* merupakan kelas utama pada aplikasi ini, yang digunakan sebagai media untuk berinteraksi antara sistem dengan *user*. Kelas *modules* berisikan fungsi-fungsi konversi yang dibutuhkan untuk proses enkripsi, dekripsi, penyisipan (*encoding*), dan ekstraksi (*decoding*). Sedangkan kelas *rc4* merupakan kelas yang menangani proses enkripsi dan dekripsi pesan teks atau citra. Dan kelas *2 LSB* merupakan kelas yang menangani proses *encoding* dan *decoding*.

4.6 Rancangan User Interface

Rancangan *user interface* merupakan rancangan tampilan sebagai media untuk berinteraksi antara pengguna dengan sistem.

a. Rancangan jendela *home*

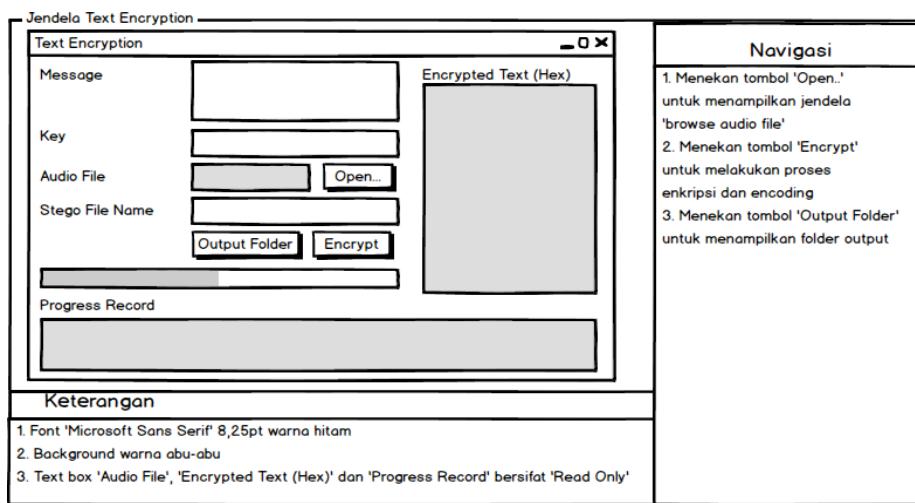
Ketika aplikasi pertama kali dijalankan maka jendela yang akan ditampilkan adalah jendela *home*. Rancangan *user interface* diperlihatkan pada gambar 4.31.



Gambar 4.31 Rancangan Jendela Home

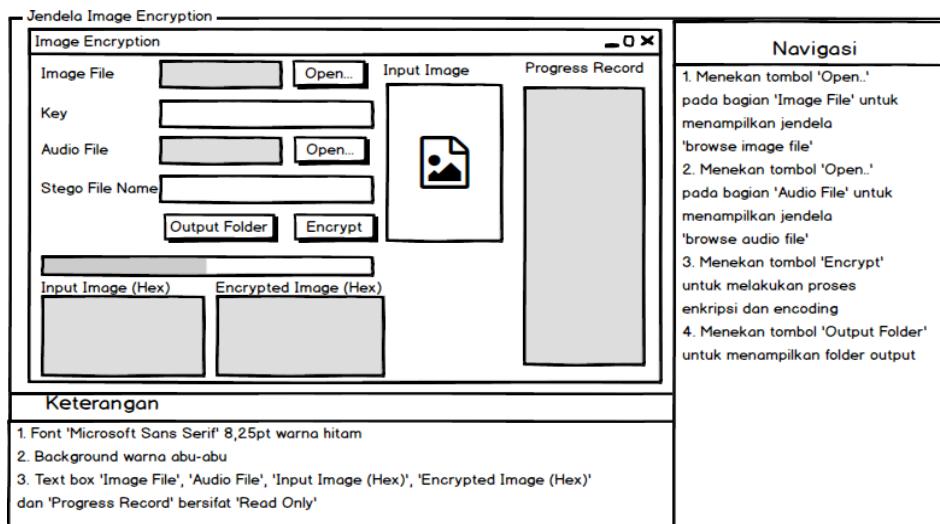
b. Rancangan jendela *text encryption*

Jendela *text encryption* merupakan jendela yang digunakan oleh *user* untuk mengeksekusi fitur enkripsi teks. Rancangan *user interface* diperlihatkan pada gambar 4.32.

Gambar 4.32 Rancangan Jendela *Text Encryption*

c. Rancangan jendela *image encryption*

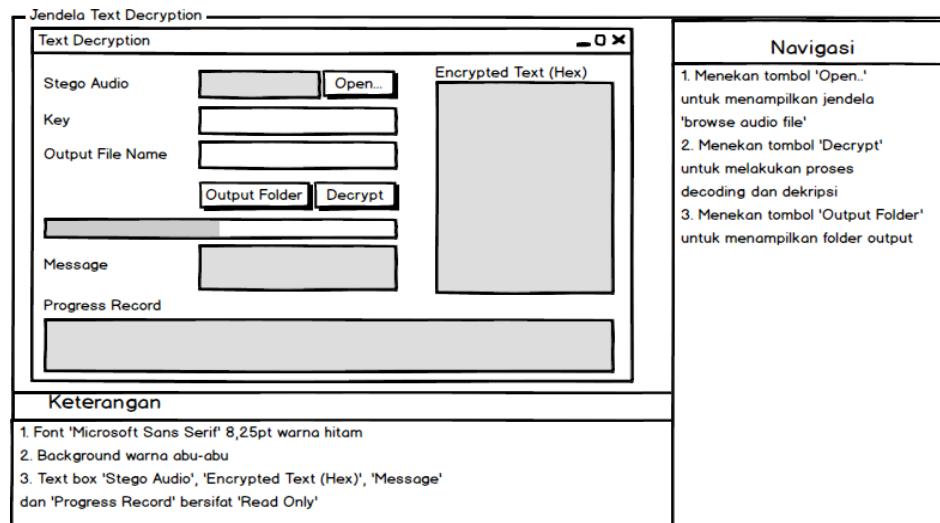
Jendela *image encryption* merupakan jendela yang digunakan oleh *user* untuk mengeksekusi fitur enkripsi gambar. Rancangan *user interface* diperlihatkan pada gambar 4.33.



Gambar 4.33 Rancangan Jendela *Image Encryption*

d. Rancangan jendela *text decryption*

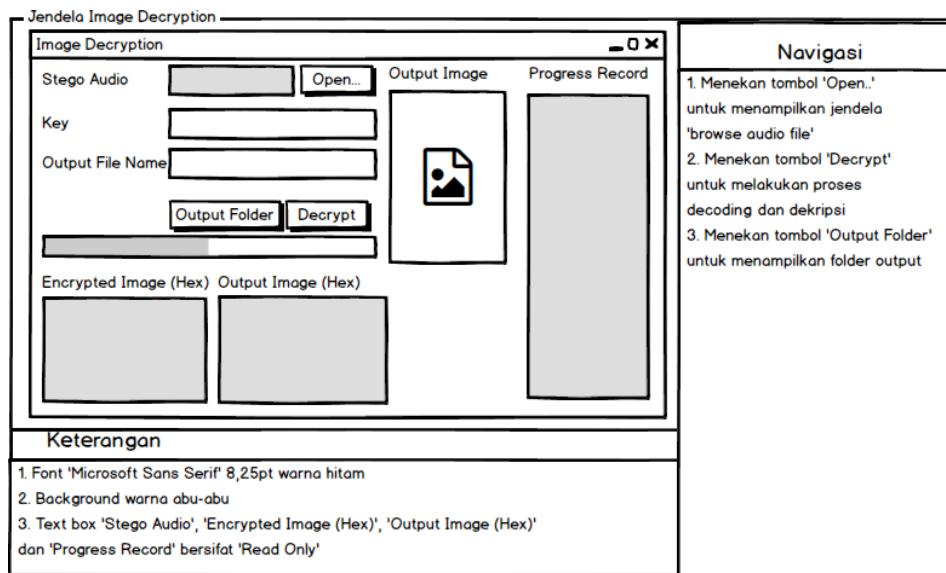
Jendela *text decryption* merupakan jendela yang digunakan oleh *user* untuk mengeksekusi fitur dekripsi teks. Rancangan *user interface* diperlihatkan pada gambar 4.34.



Gambar 4.34 Rancangan Jendela *Text Decryption*

e. Rancangan jendela *image decryption*

Jendela *image decryption* merupakan jendela yang digunakan oleh *user* untuk mengeksekusi fitur dekripsi gambar. Rancangan *user interface* diperlihatkan pada gambar 4.35.



Gambar 4.35 Rancangan Jendela *Image Decryption*

4.7 Perancangan Perhitungan

Perancangan perhitungan berfungsi untuk memudahkan dalam menerapkan algoritma dan metode yang digunakan pada sistem.

4.7.1 Perhitungan Algoritma Kriptografi RC4

Berikut ini adalah contoh perhitungan enkripsi dan dekripsi menggunakan algoritma kriptografi RC4. Langkah pertama yaitu menentukan plainteks dan kunci untuk enkripsi atau dekripsi.

Plainteks : H A

Kunci : 2 1

Kemudian menginisialisasi S-Box (array S) dan *state* array K (array K) dengan panjang 256 byte. Selanjutnya memberikan nilai array S dengan nilai 0-255 secara berurutan seperti berikut, $S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, \dots, S[255] = 255$. Setelah itu mengubah kunci yang dimasukan kedalam kode ASCII, angka 2 = 50 (kode ASCII), dan angka 1 = 49 (kode ASCII). Nilai kunci yang telah diubah kedalam kode ASCII selanjutnya dimasukan secara berulang-ulang kedalam array K seperti berikut, $K[0] = 50, K[1] = 49, K[2] = 50, K[3] = 49, \dots, K[255] = 49$.

Langkah berikutnya menginisialisasi i dan j dengan memberikan nilai 0, kemudian dilakukan KSA (*key-scheduling algorithm*) agar tercipta *state* array acak.

*Iterasi 1

$$\begin{aligned}
 i &= 0 \\
 j &= (j + S[i] + K[i]) \bmod 256 \\
 &= (0 + 0 + 50) \bmod 256 \\
 &= 50 \bmod 256 \\
 &= 50 \\
 \text{Swap}(S[0], S[50]) \\
 S[0] &= 50 \\
 S[50] &= 0
 \end{aligned}$$

*Iterasi 2

$$\begin{aligned}
 i &= 1 \\
 j &= (j + S[i] + K[i]) \bmod 256 \\
 &= (50 + 1 + 49) \bmod 256 \\
 &= 100 \bmod 256 \\
 &= 100 \\
 \text{Swap}(S[1], S[100]) \\
 S[1] &= 100 \\
 S[100] &= 1
 \end{aligned}$$

.

.

Perhitungan dilakukan hingga iterasi ke 256

$$\begin{aligned}
 i &= 255 \\
 j &= (j + S[i] + K[i]) \bmod 256 \\
 &= (233 + 167 + 49) \bmod 256 \\
 &= 449 \bmod 256 \\
 &= 193 \\
 \text{Swap}(S[255], S[193]) \\
 S[255] &= 246 \\
 S[193] &= 167
 \end{aligned}$$

Sehingga nilai array S yang didapatkan secara berurutan sebagai berikut.

$$\begin{aligned}
 S = 50, 216, 163, 93, 152, 56, 112, 168, 30, 28, 31, 148, 210, 195, 70, 144, 207, 17, \\
 85, 53, 26, 60, 109, 181, 59, 194, 32, 14, 9, 158, 186, 88, 92, 82, 4, 95, 172, 90, 74, \\
 166, 252, 67, 46, 84, 108, 232, 140, 114, 236, 154, 128, 12, 151, 89, 188, 0, 101, \\
 174, 141, 184, 103, 213, 69, 23, 162, 124, 47, 129, 250, 102, 222, 132, 227, 169, \\
 55, 61, 122, 81, 91, 83, 225, 142, 57, 20, 80, 110, 231, 111, 134, 176, 193, 86, 117, \\
 204, 147, 49, 10, 41, 79, 87, 6, 115, 7, 37, 175, 116, 2, 137, 146, 125, 106, 62, 251, \\
 72, 118, 211, 121, 45, 138, 25, 196, 13, 71, 99, 19, 239, 159, 5, 234, 179, 131, 145,
 \end{aligned}$$

97, 253, 52, 217, 64, 27, 16, 51, 189, 143, 205, 240, 206, 150, 126, 1, 173, 229, 212, 244, 243, 209, 192, 133, 235, 199, 160, 237, 18, 180, 178, 34, 156, 198, 136, 24, 191, 105, 242, 63, 21, 230, 36, 96, 33, 241, 215, 187, 233, 171, 76, 120, 190, 155, 226, 48, 218, 200, 185, 238, 78, 167, 73, 58, 113, 201, 127, 248, 11, 98, 119, 8, 224, 157, 249, 149, 29, 245, 228, 42, 130, 40, 165, 68, 100, 208, 255, 123, 221, 220, 43, 214, 94, 38, 107, 22, 219, 223, 197, 153, 202, 254, 104, 203, 35, 75, 177, 135, 66, 44, 77, 139, 164, 183, 170, 65, 182, 54, 247, 161, 3, 15, 39, 246

Setelah melakukan KSA, akan dilakukan PRGA. PRGA dilakukan sebanyak dua kali karena plainteks yang dienkripsi berjumlah dua karakter. Hal ini dikarenakan dibutuhkan satu kunci dan satu pengoperasian XOR untuk tiap-tiap karakter pada plainteks. Berikut adalah tahapan penghasilan kunci enkripsi dengan PRGA.

Inisialisasi

$$i = 0$$

$$j = 0$$

*Iterasi 1

$$i = (i + 1) \bmod 256$$

$$= (0 + 1) \bmod 256$$

$$= 1 \bmod 256$$

$$= 1$$

$$j = (j + S[i]) \bmod 256$$

$$= (0 + 216) \bmod 256$$

$$= 216 \bmod 256$$

$$= 216$$

Swap(S[1], S[216])

$$S[1] = 100$$

$$S[216] = 216$$

$$K1 = S[(S[i] + S[j]) \bmod 256]$$

$$= S[(S[1] + S[216]) \bmod 256]$$

$$= S[(100 + 216) \bmod 256]$$

$$= S[316 \bmod 256]$$

$$= S[60]$$

$$= 103 \approx 01100111$$

*Iterasi 2

$$i = (i + 1) \bmod 256$$

$$= (1 + 1) \bmod 256$$

$$= 2 \bmod 256$$

$$= 2$$

$$j = (j + S[i]) \bmod 256$$

$$= (216 + 163) \bmod 256$$

$$= 379 \bmod 256$$

$$= 123$$

$$Swap(S[i], S[j])$$

$$S[2] = 99$$

$$S[123] = 163$$

$$K2 = S[(S[i] + S[j]) \bmod 256]$$

$$= S[(S[2] + S[123]) \bmod 256]$$

$$= S[(99 + 163) \bmod 256]$$

$$= S[262 \bmod 256]$$

$$= S[6]$$

$$= 112 \approx 01110000$$

Setelah menemukan kunci dari tiap-tiap karakter, maka dilakukan operasi XOR antara karakter plainteks dengan kunci yang dihasilkan. Berikut ini adalah hasil enkripsinya.

	H	A	
Plainteks	01001000	01000001	
Kunci	01100111	01110000	XOR
Cipherteks	00101111	00110001	
	(Hexadesimal : 2F)	(Hexadesimal : 31)	

Dari proses enkripsi diatas, dihasilkan karakter cipherteks ‘2F31’. Sedangkan untuk mendekripsinya dengan cara melakukan operasi XOR antara karakter cipherteks dengan kunci. Berikut ini adalah hasil dekripsinya.

	2F	31	
Cipherteks	00101111	00110001	
Kunci	01100111	01110000	XOR
Plainteks	01001000	01000001	
	(kode ascii 72 = H)	(kode ascii 65 = A)	

4.7.2 Penyembunyian Pesan Menggunakan Metode 2LSB

Berikut adalah contoh penyembunyian pesan menggunakan metode 2LSB. Misalkan susunan *byte* audio yang belum disisipi pesan adalah sebagai berikut.

10110111 10101010 11110110 00101111

Pesan rahasia yang telah diubah ke susunan biner misalkan ‘10101110’. Penyisipan dilakukan dengan mengganti 2 bit terakhir pada susunan *byte* audio. Sehingga diperoleh susunan *byte* audio setelah pesan disisipkan menggunakan teknik 2LSB sebagai berikut.

10110110 10101010 11110111 00101110

4.7.3 Kapasitas Penyisipan File Audio

Kapasitas penyisipan pesan rahasia kedalam file audio pada dasarnya mengikuti ekstensi dari file audio yang akan disisipi. Setiap ekstensi file audio memiliki panjang *byte header* yang berbeda-beda. *Byte header* merupakan *byte* yang penting dalam file audio, sehingga perubahan yang terjadi pada *byte header* dapat mengakibatkan korup atau gangguan pada file audio. Sehingga penyisipan pesan hanya dapat dilakukan pada bagian audio selain *byte header*. Pada file audio dengan ekstensi .wav memiliki panjang 44 *byte header* (deret *byte* ke 1 sampai 44).

Pada penelitian ini, penyisipan pesan dimulai pada deret *byte* ke 51. Setiap penyisipan diberikan jarak 5 *byte* dengan penyisipan berikutnya.

Berikut adalah perhitungan yang digunakan untuk menentukan *byte* audio yang dapat disisipi.

Berikut ini perhitungan yang digunakan untuk menentukan ukuran maksimal penyisipan pesan dan persentase ukuran pesan terhadap ukuran file audio.

$$Ukuran maksimal penyisipan = \frac{\left(\frac{audio yang dapat disisipi}{6}\right) * 2}{8}(4.2)$$

Persentase ukuran pesan terhadap audio = $\frac{\text{ukuran maksimal penyisipan}}{\text{byte audio}} * 100\% \dots (4.3)$

Misalkan, file audio berekstensi .wav memiliki ukuran 220.972 byte. Maka dari ukuran tersebut, audio yang dapat disisipi pesan ialah:

$$\text{Audio yang dapat disisipi} = 220.972 - (44 + 6)$$

$$= 220.922 \text{ byte}$$

Dari hasil perhitungan audio yang dapat disisipi, maka dapat digunakan untuk menentukan ukuran maksimal penyisipan pesan.

$$\text{Ukuran maksimal penyisipan} = \frac{\left(\frac{220.922}{6}\right) * 2}{8}$$

$$= 9.205,08 \approx 9.205 \text{ byte}$$

Berdasarkan hasil perhitungan ukuran maksimal penyisipan, maka persentase ukuran pesan yang dapat disisipkan pada file audio ialah:

$$\text{Persentase ukuran pesan terhadap audio} = \frac{9.205}{220.972} * 100 = 4,165686 \% \approx 4 \%$$

Sehingga ukuran maksimal pesan yang dapat disisipkan kedalam file audio ialah 4% dari ukuran file audio pembawa pesan. Apabila ukuran pesan melebihi 4% dari ukuran file audio, maka penyisipan pesan tidak dapat dilakukan.

4.7.4 Perhitungan *Peak Signal to Noise Ratio* (PSNR)

Pengukuran *noise* pada stego audio dilakukan dengan menggunakan PSNR *Peak Signal to Noise Ratio* (PSNR). Perhitungan PSNR dilakukan dengan memakai rumus persamaan :

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{P_1^2}{P_1^2 + P_0^2 - 2P_1P_0} \right) \quad [9]$$

Dimana P_1 adalah kekuatan sinyal audio setelah disisipi pesan dan P_0 adalah kekuatan sinyal audio awal. Contoh perhitungan PSNR adalah sebagai berikut :

$$P_0 = 65,64$$

$$P_1 = 64,23$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{64,23^2}{64,23^2 + 65,64^2 - 2 \cdot 64,23 \cdot 65,64} \right)$$

$$10 \cdot \log_{10}(2075,093255) = 33,17 \text{ dB}$$

Dari contoh perhitungan PSNR, didapatkan nilai PSNR audio setelah disisipi pesan sebesar 33,17 dB.

BAB V. IMPLEMENTASI

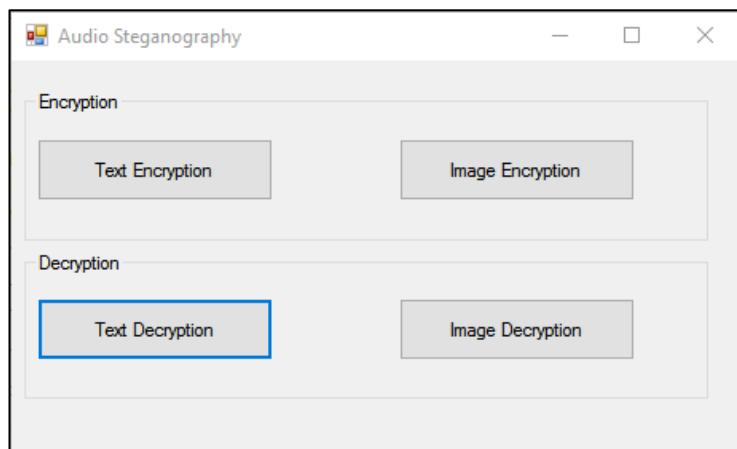
Bab implementasi menjelaskan tentang pembuatan aplikasi berdasarkan analisa dan perancangan desain sistem yang telah disusun pada bab sebelumnya.

5.1 Pembuatan Aplikasi

Pembuatan aplikasi mengacu pada desain yang dijelaskan pada bab sebelumnya. Pada bab implementasi dilakukan penulisan kode program sesuai dengan apa yang direncanakan. Aplikasi memiliki fungsi utama yaitu melakukan proses enkripsi dan penyisipan (*encoding*), serta ekstraksi (*decoding*) dan dekripsi informasi rahasia.

5.2 Pembuatan Jendela *Home*

Jendela *home* aplikasi terdiri dari tombol-tombol yang memberikan akses menuju jendela *text encryption*, *image encryption*, *text decryption*, dan *image decryption*. Tampilan jendela *home* dapat dilihat pada gambar 5.1.

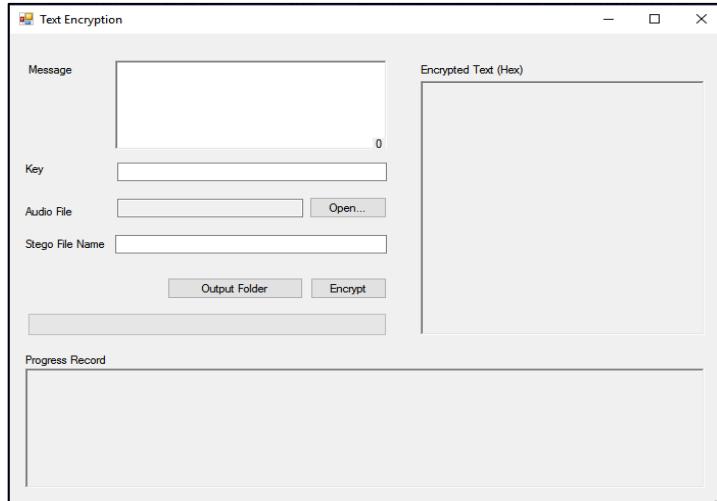


Gambar 5.1 Tampilan Jendela *Home*

5.3 Pembuatan Jendela *Text Encryption*

Jendela *text encryption* berfungsi sebagai antarmuka untuk melakukan fungsi enkripsi dan penyisipan pesan teks kedalam file audio. Untuk menjalankan proses enkripsi dan penyisipan pesan teks, diperlukan masukan berupa pesan teks yang akan disisipkan (*message*), kunci enkripsi (*key*), file audio pembawa pesan rahasia (*audio file*), dan nama keluaran file stego (*stego file name*). Proses enkripsi dan penyisipan pesan teks berjalan ketika tombol ‘*encrypt*’ di tekan. Setelah proses

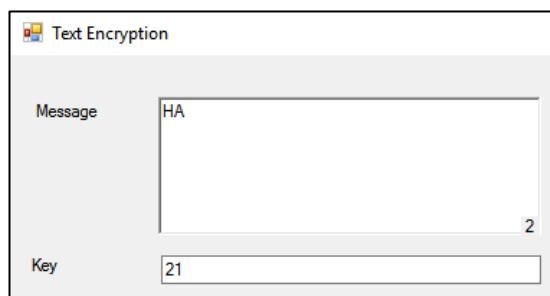
enkripsi dan penyisipan pesan teks dilakukan maka keluarannya akan disimpan pada folder *output*, yang dapat diakses dengan menekan tombol *output folder*. Tampilan jendela *text encryption* dapat dilihat pada gambar 5.2.



Gambar 5.2 Tampilan Jendela *Text Encryption*

5.3.1 Enkripsi Pesan Teks

Proses pertama yang dilakukan adalah pengenkripsi terhadap pesan teks sebelum disisipkan kedalam file audio. Kebutuhan yang diperlukan untuk melakukan proses enkripsi adalah pesan teks (*message*) dan kunci enkripsi (*key*). Pada gambar 5.3 menunjukkan masukan pesan teks berupa karakter ‘HA’ dengan panjang 2 karakter (2 byte) dan kunci enkripsi berupa karakter ‘21’.



Gambar 5.3 Masukan Enkripsi Pesan Teks

Kode program yang digunakan untuk melakukan proses enkripsi pesan teks adalah sebagai berikut.

```
For z = 0 To 255
    s(z) = z
    k(z) = Asc(key(z Mod key.Length))
```

```

Next
' State Array K
Dim j As Integer = 0
Dim tmp As Integer = 0
For n = 0 To 255
    j = (j + s(n) + k(n)) Mod 256
    tmp = s(n)
    s(n) = s(j)
    s(j) = tmp
Next
' Penghasilan Kunci Enkripsi PRGA
Dim i As Integer = 0
j = 0
tmp = 0
For n = 1 To inputText.Length
    i = (i + 1) Mod 256
    j = (j + s(i)) Mod 256
    tmp = s(i)
    s(i) = s(j)
    s(j) = tmp

    kE = s((s(i) + s(j)) Mod 256)
    input = Asc(inputText(n - 1))
    output = input Xor kE

    If Hex(output).Length = 1 Then
        outputText += "0" & Hex(output).ToLower
    Else
        outputText += Hex(output).ToLower
    End If
Next

```

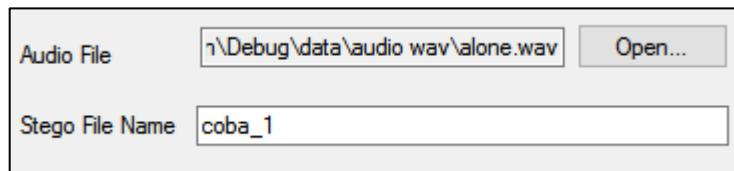
Dari masukan pesan teks dan kunci enkripsi yang ditunjukkan pada gambar 5.3, dihasilkan keluaran berupa cipherteks yang dapat dilihat pada gambar 5.4.

Encrypted Text (Hex)
2F31

Gambar 5.4 Cipherteks Hasil Enkripsi

5.3.2 Penyisipan Pesan Teks

Proses yang dilakukan setelah enkripsi adalah penyisipan (*encoding*) pesan teks kedalam file audio. Kebutuhan yang diperlukan untuk proses penyisipan ialah teks yang telah dienkripsi (cipherteks), file audio (*audio file*), dan nama keluaran file stego (*stego file name*). Cipherteks yang disisipkan adalah keluaran proses enkripsi yang ditunjukkan pada gambar 5.4. Masukan yang digunakan untuk proses penyisipan dapat dilihat pada gambar 5.5.



Gambar 5.5 Masukan Penyisipan Pesan Teks

Kode program yang digunakan untuk melakukan proses penyisipan pesan teks adalah sebagai berikut.

```

fileReader = Text_audio.Substring(0, 400)
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
"_stego.txt", fileReader, True)

    Dim j As Integer = 400
    Dim prog As Integer
    For i As Integer = 0 To Text_string.Length - 1 Step 2
        fileReader = Text_audio.Substring(j, 8)
        fileReader = fileReader.Remove(6, 2) +
Text_string.Substring(i, 2)
        j = j + 8
        fileReader = fileReader + Text_audio.Substring(j, 40)
        j = j + 40
        prog = i
        oReturn.Append(fileReader)
        Console.WriteLine(fileReader)
    Next
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
"_stego.txt", oReturn.ToString, True)

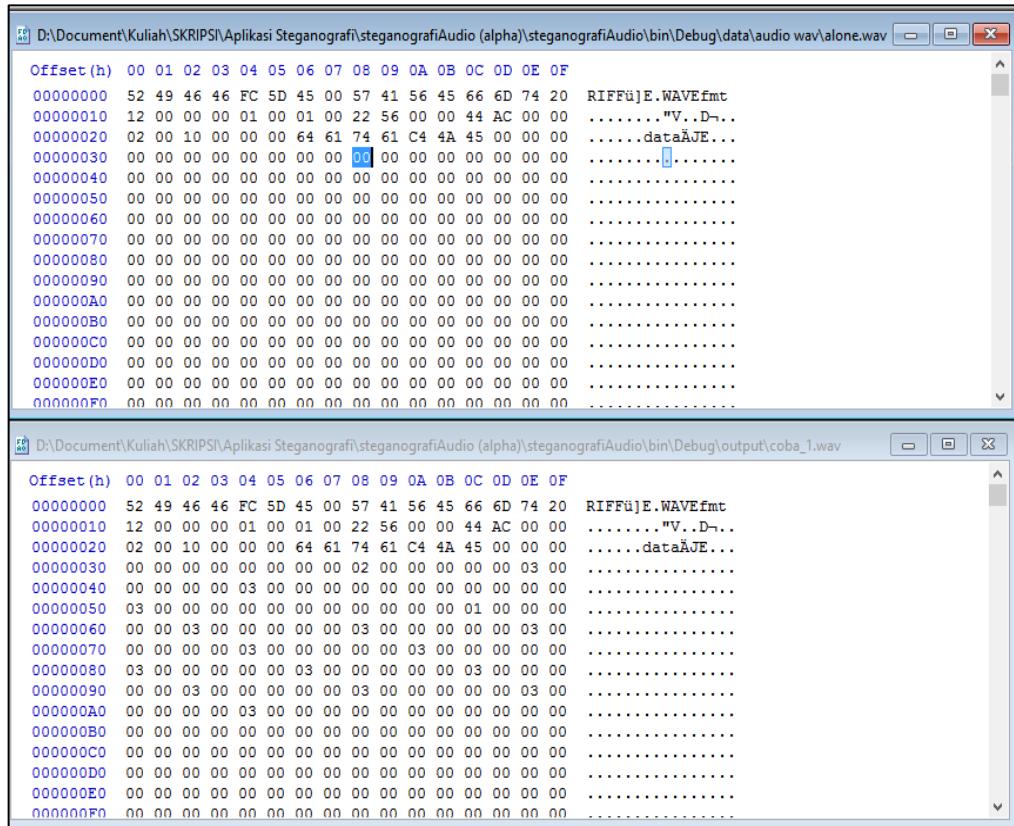
    For z = 0 To 11
        prog += 1
        fileReader = Text_audio.Substring(j, 8)
        fileReader = fileReader.Remove(6, 2) + "11"
    
```

```

j = j + 8
fileReader = fileReader + Text_audio.Substring(j, 40)
j = j + 40
My.Computer.FileSystem.WriteAllText("temp\" + savefile
+ "_stego.txt", fileReader, True)
Next
caps = (j / Text_audio.Length) * 100
Dim k As Integer = Text_audio.Length - j
fileReader = Text_audio.Substring(j, k)
My.Computer.FileSystem.WriteAllText("temp\" + savefile
+ "_stego.txt", fileReader, True)

```

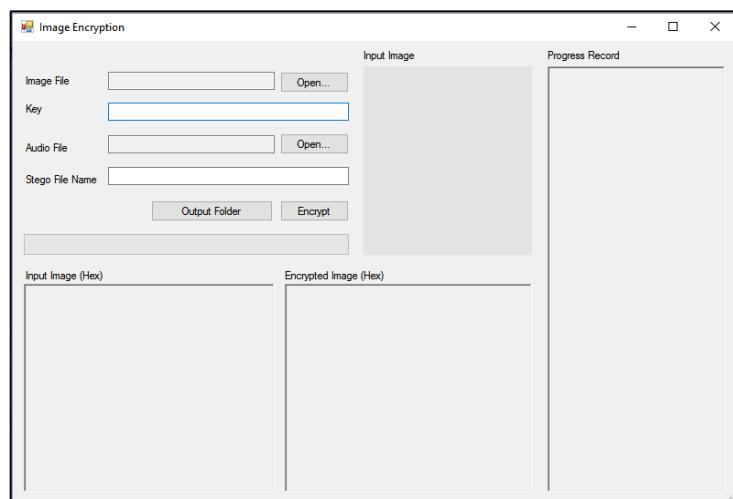
Dari masukan parameter proses penyisipan pesan teks yang ditunjukkan pada gambar 5.5, dihasilkan keluaran berupa file stego audio dengan nama ‘coba_1.wav’. Perbandingan file audio sebelum dan setelah disisipi pesan teks dapat dilihat pada gambar 5.6.



Gambar 5.6 Perbandingan File Audio Sebelum dan Setelah Disisipi Pesan Teks

5.4 Pembuatan Jendela *Image Encryption*

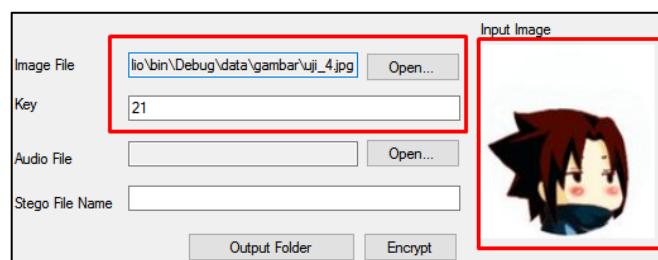
Jendela *image encryption* berfungsi sebagai antarmuka untuk melakukan fungsi enkripsi dan penyisipan citra kedalam file audio. Untuk menjalankan proses enkripsi dan penyisipan citra, diperlukan masukan berupa citra yang akan disisipkan (*image file*), kunci enkripsi (*key*), file audio pembawa pesan rahasia (*audio file*), dan nama keluaran file stego (*stego file name*). Proses enkripsi dan penyisipan citra berjalan ketika tombol ‘*encrypt*’ di tekan. Setelah proses enkripsi dan penyisipan citra dilakukan maka keluarannya akan disimpan pada folder *output*, yang dapat diakses dengan menekan tombol *output folder*. Tampilan jendela *image encryption* dapat dilihat pada gambar 5.7.



Gambar 5.7 Tampilan Jendela *Image Encryption*

5.4.1 Enkripsi Citra

Proses pertama yang dilakukan adalah pengenkripsi terhadap citra sebelum disisipkan kedalam file audio. Kebutuhan yang diperlukan untuk melakukan proses enkripsi adalah file citra (*image file*) dan kunci enkripsi (*key*). Masukan parameter proses enkripsi citra dapat dilihat pada gambar 5.8.



Gambar 5.8 Masukan Enkripsi Citra

Kode program yang digunakan untuk melakukan proses enkripsi citra adalah sebagai berikut.

```

For z = 1 To inputText.Length Step 2
    inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
Next
For z = 0 To 255
    s(z) = z
    k(z) = Asc(key(z Mod key.Length))
Next
'State Array K
Dim j As Integer = 0
Dim tmp As Integer = 0
For n = 0 To 255
    j = (j + s(n) + k(n)) Mod 256
    tmp = s(n)
    s(n) = s(j)
    s(j) = tmp
Next
'Penghasilan Kunci Enkripsi PRGA
Dim i As Integer = 0
j = 0
tmp = 0
For n = 1 To inputText_.Length
    i = (i + 1) Mod 256
    j = (j + s(i)) Mod 256
    tmp = s(i)
    s(i) = s(j)
    s(j) = tmp
    kE = s((s(i) + s(j)) Mod 256)
    input = Asc(inputText_(n - 1))
    output = input Xor kE
    If Hex(output).Length = 1 Then
        outputText += "0" & Hex(output).ToLower
    Else
        outputText += Hex(output).ToLower
    End If
Next

```

Dari masukan file citra dan kunci enkripsi yang ditunjukkan pada gambar 5.8, dihasilkan keluaran berupa cipher citra dalam bentuk heksadesimal yang dapat dilihat pada gambar 5.9.

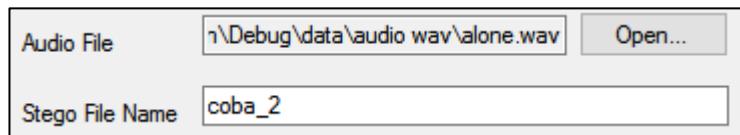


```
Encrypted Image (Hex)
98a8a6e1412fe0ef7c6047454c3532aff888bebf2ef1 ^
f55146333d20d35d4ed4f2fd8563718cc805daaff5d
506d412a66553d5286891148d1ec4cf5306f79d6fa
a225cfba64fc9b3e103806b9c2fe90cc0e5967a5da
e9f05fc274e5985b6ac26ff56516ee128d7884434fd
2c21ddcacf7b100e333a0bae47a8708601f3363997
eed91223b678b86979a9c232cd681175312ddfec3
4970c6ba7a7492ea604716cb4c942070038447071
a8d8f8a8055d7b247dfbb9eeee823b75849c1348b6
d25dea265bcb3fe7de9a82bcc25118a99a5694b34
98b93119691bbe16ffc47018f5c1e134f1dc1ffc3c5e
df0dad347b08badcc4d1b5656df3677b50c3dd0796
937ebfe08cb1463da57a09fb279253cc7c2f47ad20
da2e25aefe50e2cafe523e19e13be379e24ce5fce4
8787ca9cee050ea026049e939ebaa7cebf396c38
7c4327042981cbfd20ba2f8cd80867c84a5c0f210a
ba27b10a432de094b28f5c8cf750c7c2a42cc3d60
```

Gambar 5.9 Cipher Citra Hasil Enkripsi

5.4.2 Penyisipan Citra

Proses yang dilakukan setelah enkripsi adalah penyisipan (*encoding*) citra kedalam file audio. Kebutuhan yang diperlukan untuk proses penyisipan ialah citra yang telah dienkripsi (cipher citra), file audio (*audio file*), dan nama keluaran file stego (*stego file name*). Cipher citra yang disisipkan adalah keluaran proses enkripsi yang ditunjukkan pada gambar 5.9. Masukan yang digunakan untuk proses penyisipan dapat dilihat pada gambar 5.10.



Gambar 5.10 Masukan Penyisipan Citra

Kode program yang digunakan untuk melakukan proses penyisipan citra adalah sebagai berikut.

```
fileReader = Text_audio.Substring(0, 400)
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
"_stego.txt", fileReader, True)
    Dim j As Integer = 400
    Dim prog As Integer
    For i As Integer = 0 To Text_string.Length - 1 Step 2
        fileReader = Text_audio.Substring(j, 8)
```

```

        fileReader      =      fileReader.Remove(6,      2)      +
Text_string.Substring(i, 2)
        j = j + 8
        fileReader = fileReader + Text_audio.Substring(j, 40)
        j = j + 40
        prog = i
        oReturn.Append(fileReader)
        Console.WriteLine(fileReader)

    Next
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
"_stego.txt", oReturn.ToString, True)

    For z = 0 To 11
        prog += 1
        fileReader = Text_audio.Substring(j, 8)
        fileReader = fileReader.Remove(6, 2) + "11"
        j = j + 8
        fileReader = fileReader + Text_audio.Substring(j, 40)
        j = j + 40
        My.Computer.FileSystem.WriteAllText("temp\" + savefile
+ "_stego.txt", fileReader, True)
    Next
    caps = (j / Text_audio.Length) * 100
    Dim k As Integer = Text_audio.Length - j
    fileReader = Text_audio.Substring(j, k)
    My.Computer.FileSystem.WriteAllText("temp\" +
savefile + "_stego.txt", fileReader, True)

```

Dari masukan parameter proses penyisipan citra yang ditunjukkan pada gambar 5.10, dihasilkan keluaran berupa file stego audio dengan nama ‘coba_2.wav’. Perbandingan file audio sebelum dan setelah disisipi citra dapat dilihat pada gambar 5.11.

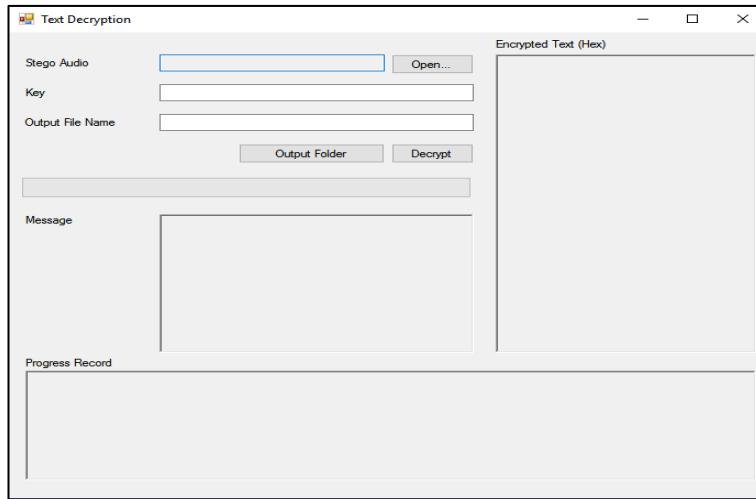
D:\Document\Kuliah\SKRIPSI\Aplikasi Steganografi\steganografiAudio (alpha)\steganografiAudio\bin\Debug\data\audio wav\alone.wav																
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	52	49	46	46	FC	5D	45	00	57	41	56	45	66	6D	74	20
00000010	12	00	00	00	01	00	01	00	22	56	00	00	44	AC	00	00
00000020	02	00	10	00	00	64	61	74	61	C4	4A	45	00	00	00	..dataÂE..
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

D:\Document\Kuliah\SKRIPSI\Aplikasi Steganografi\steganografiAudio (alpha)\steganografiAudio\bin\Debug\output\coba_2.wav																
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	52	49	46	46	FC	5D	45	00	57	41	56	45	66	6D	74	20
00000010	12	00	00	00	01	00	01	00	22	56	00	00	44	AC	00	00
00000020	02	00	10	00	00	64	61	74	61	C4	4A	45	00	00	00	..dataÂE..
00000030	00	00	02	00	00	00	00	01	00	00	00	00	02	00	00	00
00000040	00	00	00	00	00	00	00	00	00	02	00	00	00	00	00	00
00000050	02	00	00	00	00	00	02	00	00	00	00	00	00	00	00	00
00000060	00	00	02	00	00	00	00	02	00	00	00	00	00	01	00	00
00000070	00	00	00	02	00	00	00	00	03	00	00	00	00	00	00	00
00000080	02	00	00	00	00	00	00	00	00	00	00	01	00	00	00	00
00000090	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00
000000B0	02	00	00	00	00	03	00	00	00	00	00	03	00	00	00	00
000000C0	00	00	03	00	00	00	00	02	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	03	00	00	00	00	00	00	00
000000E0	02	00	00	00	00	00	03	00	00	00	00	03	00	00	00	00
000000F0	00	00	01	00	00	00	00	03	00	00	00	00	03	00	00	00

Gambar 5.11 Perbandingan File Audio Sebelum dan Setelah Disisipi Citra

5.5 Pembuatan Jendela *Text Decryption*

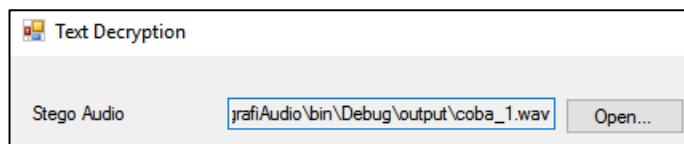
Jendela *text decryption* berfungsi sebagai antarmuka untuk melakukan fungsi ekstraksi dan dekripsi pesan teks. Untuk menjalankan proses ekstraksi dan dekripsi pesan teks, diperlukan masukan berupa file stego audio (*Stego Audio*), kunci dekripsi (*key*), dan nama file keluaran (*output file name*). Proses ekstraksi dan dekripsi pesan teks berjalan ketika tombol ‘*decrypt*’ di tekan. Setelah proses ekstraksi dan dekripsi pesan teks dilakukan maka keluarannya akan disimpan pada folder *output*, yang dapat diakses dengan menekan tombol *output folder*. Tampilan jendela *text decryption* dapat dilihat pada gambar 5.12.



Gambar 5.12 Tampilan Jendela *Text Decryption*

5.5.1 Ekstraksi Pesan Teks

Proses pertama yang dilakukan adalah melakukan ekstraksi (*decoding*) pesan teks dari file stego audio. Kebutuhan yang diperlukan untuk melakukan proses ekstraksi adalah file stego audio (*stego audio*) seperti yang ditunjukkan pada gambar 5.13.



Gambar 5.13 Masukan Ekstraksi Pesan Teks

Kode program yang digunakan untuk melakukan proses ekstraksi pesan teks adalah sebagai berikut.

```

For i As Integer = 0 To Text_stego.Length / 8 - 1
    a = Text_stego.Substring(j, 8)
    filereader.Append(Strings.Right(a, 2))
    filereader.Append(Separator)
    If filereader.Length Mod 8 = 0 And filereader.Length > 16 Then
        b = filereader.ToString.Substring(k - 16, 24)
        If b = "111111111111111111111111" Then
            i = Text_stego.Length / 8 - 1
        End If
        k += 8
        Console.WriteLine(b)
    End If
End If

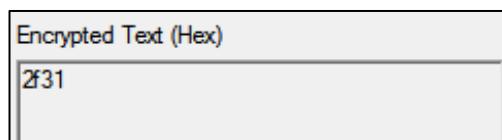
```

```

j = j + 48
Next
Dim str As String
str = filereader.ToString
str = str.Remove(str.Length - 24)
File.WriteAllText("temp2\" + savefile + "_2lsb.txt", str)
Return "temp2\" + savefile + "_2lsb.txt"
Return str

```

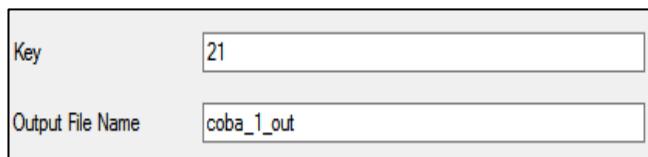
Dari masukan file stego audio yang ditunjukkan pada gambar 5.13, dihasilkan keluaran berupa cipherteks yang dapat dilihat pada gambar 5.14.



Gambar 5.14 Cipherteks Hasil *Decoding*

5.5.2 Dekripsi Pesan Teks

Proses yang dilakukan setelah ekstraksi adalah dekripsi pesan teks agar pesan dapat dibaca oleh penerima pesan. Kebutuhan yang diperlukan untuk proses dekripsi ialah cipherteks hasil *decoding*, kunci dekripsi (*key*), dan nama file keluaran (*output file name*). Masukan yang digunakan untuk proses dekripsi dapat dilihat pada gambar 5.15.



Gambar 5.15 Masukan Dekripsi Pesan Teks

Kode program yang digunakan untuk melakukan proses dekripsi pesan teks adalah sebagai berikut.

```

For z = 1 To inputText.Length Step 2
    inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
Next
For z = 0 To 255
    s(z) = z
    k(z) = Asc(key(z Mod key.Length))
Next

```

```

For n = 0 To 255
    j = (j + s(n) + k(n)) Mod 256
    tmp = s(n)
    s(n) = s(j)
    s(j) = tmp
Next
Dim i As Integer = 0
j = 0
tmp = 0
For n = 1 To inputText_.Length
    i = (i + 1) Mod 256
    j = (j + s(i)) Mod 256
    tmp = s(i)
    s(i) = s(j)
    s(j) = tmp
    kE = s((s(i) + s(j)) Mod 256)
    input = Asc(inputText_(n - 1))
    output = input Xor kE
    outputText += Chr(output)
Next

```

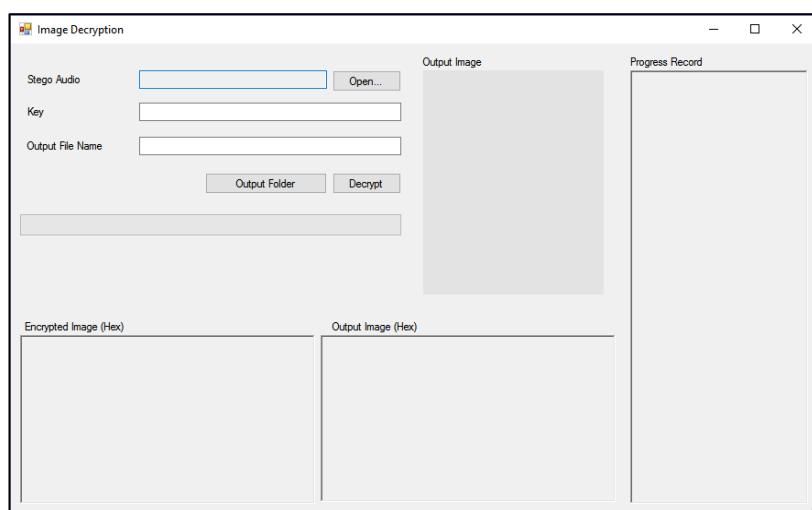
Dari masukan parameter proses dekripsi pesan teks yang ditunjukkan pada gambar 5.15, dihasilkan keluaran berupa pesan teks yang telah terdekripsi. Pesan teks yang terdekripsi ditampilkan pada jendela *text decryption* dan disimpan pada folder *output* dengan ekstensi .txt. Gambar 5.16 menunjukkan pesan teks terdekripsi yang ditampilkan pada jendela *text decryption*.



Gambar 5.16 Pesan Teks Terdekripsi

5.6 Pembuatan Jendela *Image Decryption*

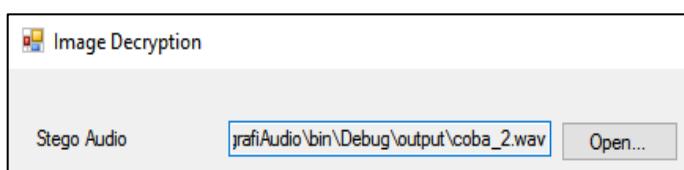
Jendela *image decryption* berfungsi sebagai antarmuka untuk melakukan fungsi ekstraksi dan dekripsi citra. Untuk menjalankan proses ekstraksi dan dekripsi citra, diperlukan masukan berupa file stego audio (*Stego Audio*), kunci dekripsi (*key*), dan nama file keluaran (*output file name*). Proses ekstraksi dan dekripsi citra berjalan ketika tombol ‘*decrypt*’ di tekan. Setelah proses ekstraksi dan dekripsi citra dilakukan maka keluarannya akan disimpan pada folder *output*, yang dapat diakses dengan menekan tombol *output folder*. Tampilan jendela *image decryption* dapat dilihat pada gambar 5.17.



Gambar 5.17 Tampilan Jendela *Image Decryption*

5.6.1 Ekstraksi Citra

Proses pertama yang dilakukan adalah melakukan ekstraksi (*decoding*) citra dari file stego audio. Kebutuhan yang diperlukan untuk melakukan proses ekstraksi adalah file stego audio (*stego audio*) seperti yang ditunjukkan pada gambar 5.18.



Gambar 5.18 Masukan Ekstraksi Citra

Kode program yang digunakan untuk melakukan proses ekstraksi citra adalah sebagai berikut.

```

For i As Integer = 0 To Text_stego.Length / 8 - 1
    a = Text_stego.Substring(j, 8)
    filereader.Append(Strings.Right(a, 2))
    filereader.Append(Separator)
    If filereader.Length Mod 8 = 0 And filereader.Length > 16 Then
        b = filereader.ToString.Substring(k - 16, 24)
        If b = "111111111111111111111111" Then
            i = Text_stego.Length / 8 - 1
        End If
        k += 8
        Console.WriteLine(b)
    End If
    j = j + 48
Next
Dim str As String
str = filereader.ToString
str = str.Remove(str.Length - 24)
File.WriteAllText("temp2\" + savefile + "_2lsb.txt", str)
Return "temp2\" + savefile + "_2lsb.txt"
Return str

```

Dari masukan file stego audio yang ditunjukkan pada gambar 5.18, dihasilkan keluaran berupa cipher citra yang dapat dilihat pada gambar 5.19.



The screenshot shows a hex editor window titled "Encrypted Image (Hex)". The content is a large block of hexadecimal data, starting with 98a8a6e1412fe0ef7c6047454c3532aff888bef2ef1f55146333 and ending with c3c5edf0dad347b08badcc4d1b5656df3677b50c3dd0796937. The window has scroll bars on the right and bottom.

Gambar 5.19 Cipher Citra Hasil *Decoding*

5.6.2 Dekripsi Citra

Proses yang dilakukan setelah ekstraksi adalah dekripsi citra agar dapat dimengerti oleh penerima pesan. Kebutuhan yang diperlukan untuk proses dekripsi ialah cipher citra hasil *decoding*, kunci dekripsi (*key*), dan nama file keluaran (*output file name*). Masukan yang digunakan untuk proses dekripsi dapat dilihat pada gambar 5.20.

Key	21
Output File Name	coba_2_out

Gambar 5.20 Masukan Dekripsi Citra

Kode program yang digunakan untuk melakukan proses dekripsi citra adalah sebagai berikut.

```

For z = 1 To inputText.Length Step 2
    inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
Next

For z = 0 To 255
    s(z) = z
    k(z) = Asc(key(z Mod key.Length))
Next

For n = 0 To 255
    j = (j + s(n) + k(n)) Mod 256
    tmp = s(n)
    s(n) = s(j)
    s(j) = tmp
Next

Dim i As Integer = 0
j = 0
tmp = 0

For n = 1 To inputText_.Length
    i = (i + 1) Mod 256
    j = (j + s(i)) Mod 256

```

```

tmp = s(i)
s(i) = s(j)
s(j) = tmp
kE = s((s(i) + s(j)) Mod 256)
input = Asc(inputText_(n - 1))
output = input Xor kE

If Hex(output).Length = 1 Then
    outputText += "0" & Hex(output).ToLower
Else
    outputText += Hex(output).ToLower
End If
Next

```

Dari masukan parameter proses dekripsi citra yang ditunjukkan pada gambar 5.20, dihasilkan keluaran berupa citra yang telah terdekripsi. Citra yang terdekripsi ditampilkan pada jendela *image decryption* dan disimpan pada folder *output* dengan ekstensi .jpg. Gambar 5.21 menunjukkan citra terdekripsi yang ditampilkan pada jendela *image decryption*.



Gambar 5.21 Citra Terdekripsi

BAB VI. PENGUJIAN DAN PEMBAHASAN

Pada bab ini dilakukan pengujian setelah implementasi sistem. Pengujian dilakukan untuk mengetahui kesesuaian fungsional aplikasi dengan apa yang direncanakan. Pengujian dibagi menjadi dua yaitu pengujian sistem dan pengujian hasil. Pengujian sistem berfungsi untuk menguji fitur-fitur yang ada pada aplikasi. Sedangkan pengujian hasil adalah melakukan analisa untuk mengetahui kesesuaian dengan metode yang digunakan serta kelayakan penggunaan metode pada aplikasi yang serupa.

6.1 Pengujian Sistem

Pengujian sistem menggunakan metode *blackbox*. Metode ini memungkinkan adanya pengembangan untuk melatih seluruh fungsi sistem. Dengan menggunakan metode ini dapat dinilai apakah masukan dan keluaran yang diterima sudah tepat atau belum. Berikut adalah metode *blackbox* untuk melakukan pengujian aplikasi.

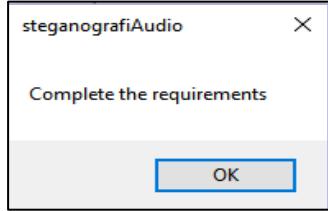
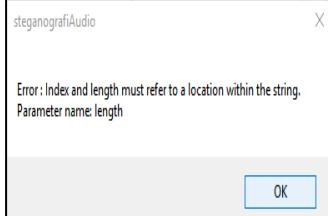
Tabel 6.1 Pengujian Sistem

Aplikasi				
No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
1	Memilih dan menampilkan citra	Citra yang dipilih ditampilkan pada <i>picturbox</i> dan <i>path file</i> ditampilkan pada <i>textbox</i> .	Citra yang dipilih dapat ditampilkan pada <i>picturbox</i> dan <i>path file</i> ditampilkan pada <i>textbox</i> .	Berhasil
2	Memilih dan menampilkan file audio	<i>Path file audio</i> ditampilkan pada <i>textbox</i> .	<i>Path file audio</i> dapat ditampilkan pada <i>textbox</i> .	Berhasil

No	Skenario	Hasil yang diharapkan	Hasil yang terjadi	Keterangan
3	Melakukan proses enkripsi pesan teks	Ketika tombol <i>encrypt</i> ditekan, maka dilakukan proses enkripsi pada pesan teks, serta dilanjutkan proses penyisipan kedalam file audio	Pesan teks dapat terenkripsi dan tersisipkan kedalam file audio	Berhasil
4	Melakukan proses enkripsi citra	Ketika tombol <i>encrypt</i> ditekan, maka dilakukan proses enkripsi pada citra, serta dilanjutkan proses penyisipan kedalam file audio	Citra dapat terenkripsi dan tersisipkan kedalam file audio	Berhasil
5	Melakukan proses dekripsi pesan teks	Ketika tombol <i>decrypt</i> ditekan, maka dilakukan proses ekstraksi, kemudian dilanjutkan pada proses dekripsi pesan teks.	Pesan teks dapat terekstraksi dan terdekripsi.	Berhasil
6	Melakukan proses dekripsi citra	Ketika tombol <i>decrypt</i> ditekan, maka dilakukan proses ekstraksi, kemudian dilanjutkan pada proses dekripsi citra.	Citra dapat terekstraksi dan terdekripsi.	Berhasil
7	Menyimpan file keluaran pada folder <i>output</i>	File keluaran dari proses enkripsi maupun dekripsi, otomatis tersimpan pada folder <i>output</i> .	File keluaran dapat tersimpan secara otomatis pada folder <i>output</i> .	Berhasil

Dari tabel 6.1 dapat disimpulkan bahwa aplikasi dapat berjalan dengan baik ketika pengujian aplikasi dilakukan dengan menggunakan parameter yang sesuai. Pengujian berikutnya dilakukan untuk mengetahui notifikasi *error* ketika aplikasi mendapatkan skenario yang memiliki parameter tidak sesuai. Pengujian notifikasi *error* dapat dilihat pada tabel 6.2.

Tabel 6.2 Pengujian Notifikasi *Error*

Aplikasi				
No	Skenario	Proses	Notifikasi <i>Error</i>	Keterangan
1	Nilai masukan tidak lengkap	Seluruh masukan yang terdapat pada jendela <i>text encryption, image encryption, text decryption, dan image decryption</i> harus terisi lengkap		Notifikasi <i>error</i> muncul apabila masukan untuk proses enkripsi atau dekripsi tidak lengkap
2	Kesalahan pada saat proses enkripsi atau dekripsi	Terjadi kesalahan pada saat melakukan proses enkripsi atau dekripsi dikarenakan kondisi tertentu (ukuran masukan pesan yang disisipkan melebihi batas file audio pembawa pesan yang dimasukkan, masukan file stego audio tidak sesuai, dan lain-lain)		Notifikasi <i>error</i> muncul beserta penjelasan penyebab terjadinya <i>error</i> pada saat melakukan proses enkripsi atau dekripsi.

6.2 Pengujian Hasil

Pengujian hasil aplikasi kemanan informasi menggunakan algoritma kriptografi RC4 dan metode steganografi audio 2LSB, dilakukan dengan menggunakan 3 file uji audio pembawa pesan yang dijelaskan pada tabel 6.3.

Tabel 6.3 File Uji Audio

No	Nama File Audio	Ukuran File	Durasi Audio
1	all.wav	3.985.688 byte	1:30
2	alone.wav	4.546.052 byte	1:42
3	life.wav	6.204.800 byte	2:20

Sedangkan untuk pesan yang disisipkan pada file audio menggunakan 3 file uji citra dan 3 sample uji pesan teks dapat dilihat pada tabel 6.4 dan 6.5.

Tabel 6.4 File Uji Citra

No	Nama File Citra	Ukuran File	Ukuran Pixel	Citra
1	Uji_1.jpg	571 byte	30 x 30	
2	Uji_2.jpg	6.839 byte	350 x 350	
3	Uji_3.jpg	15.315 byte	250 x 260	

Tabel 6.5 Sampel Uji Pesan Teks

No	Teks Pengujian	Pesan Teks	Panjang Plaintek
1	plainteks 1	Hallo, nama saya Binar Prihadmantyo.	36 karakter (36 byte)
2	plainteks 2	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi.	106 karakter (106 byte)
3	plainteks 3	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi. Usia saya 22 tahun, tinggal di Perum Permata Saxofone E-6, Kelurahan Jatimulyo, Kecamatan Lowokwaru, Kota Malang.	220 karakter (220 byte)

6.2.1 Pengujian Enkripsi dan Dekripsi

Pengujian enkripsi dan dekripsi dilakukan untuk mengetahui keberhasilan aplikasi dalam melakukan proses enkripsi dan penyisipan, serta ekstraksi dan dekripsi pesan ke dalam file audio. Berikut ini adalah pengujian enkripsi dan dekripsi pesan teks dan citra pada file audio.

Tabel 6.6 Enkripsi Pesan Teks dan Citra pada File Audio

No	Sebelum Penyisipan			Informasi Rahasia	Setelah Penyisipan			Kunci	Waktu Enkripsi	Waktu Penyisipan (Encoding)	Keterangan
	File Audio	Ukuran Audio (Byte)	Durasi Audio		File Stego	Ukuran Stego (Byte)	Durasi Stego				
Pesan Teks											
1	all.wav	3.985.688	1:30	plainteks 1	all1.wav	3.985.688	1:30	polinema2017	0 s	6,485 s	Berhasil
				plainteks 2	all2.wav	3.985.688	1:30		0 s	8,949 s	Berhasil
				plainteks 3	all3.wav	3.985.688	1:30		0 s	13,866 s	Berhasil
2	alone.wav	4.546.052	1:42	plainteks 1	alone1.wav	4.546.052	1:42	polinema2017	0 s	6,516 s	Berhasil
				plainteks 2	alone2.wav	4.546.052	1:42		0 s	9,345 s	Berhasil
				plainteks 3	alone3.wav	4.546.052	1:42		0 s	14,448 s	Berhasil
3	life.wav	6.204.800	2:20	plainteks 1	life1.wav	6.204.800	2:20	polinema2017	0 s	8,566 s	Berhasil
				plainteks 2	life2.wav	6.204.800	2:20		0 s	11,032 s	Berhasil
				plainteks 3	life3.wav	6.204.800	2:20		0 s	15,923 s	Berhasil
Citra											
4	all.wav	3.985.688	1:30	uji_1.jpg	all_1.wav	3.985.688	1:30	polinema2017	0,016 s	27,877 s	Berhasil
				uji_2.jpg	all_2.wav	3.985.688	1:30		0,205 s	273,031 s	Berhasil
				uji_3.jpg	all_3.wav	3.985.688	1:30		0,546 s	608,231 s	Berhasil
5	alone.wav	4.546.052	1:42	uji_1.jpg	alone_1.wav	4.546.052	1:42	polinema2017	0,018 s	28,340 s	Berhasil
				uji_2.jpg	alone_2.wav	4.546.052	1:42		0,213 s	273,148 s	Berhasil
				uji_3.jpg	alone_3.wav	4.546.052	1:42		0,494 s	612,531 s	Berhasil
6	life.wav	6.204.800	2:20	uji_1.jpg	life_1.wav	6.204.800	2:20	polinema2017	0,010 s	30,400 s	Berhasil
				uji_2.jpg	life_2.wav	6.204.800	2:20		0,176 s	275,437 s	Berhasil
				uji_3.jpg	life_3.wav	6.204.800	2:20		0,603 s	607,947 s	Berhasil

Tabel 6.7 Dekripsi Pesan Teks dan Citra pada Stego Audio

No	Stego audio	Ukuran Stego	Durasi	Kunci Dekripsi	Keluaran	Waktu Ekstraksi	Waktu Dekripsi	Keterangan
Pesanan Teks								
1	all1.wav	3.985.688	1:30	polinema2017	all1_out	1,509 s	0,001 s	Berhasil
2	all2.wav	3.985.688	1:30	polinema2017	all2_out	1,787 s	0,002 s	Berhasil
3	all3.wav	3.985.688	1:30	polinema2017	all3_out	2,357 s	0,002 s	Berhasil
4	alone1.wav	4.546.052	1:42	polinema2017	alone1_out	1,996 s	0,001 s	Berhasil
5	alone2.wav	4.546.052	1:42	polinema2017	alone2_out	2,062 s	0,001 s	Berhasil
6	alone3.wav	4.546.052	1:42	polinema2017	alone3_out	2,535 s	0,005 s	Berhasil
7	life1.wav	6.204.800	2:20	polinema2017	life1_out	2,415 s	0,002 s	Berhasil
8	life2.wav	6.204.800	2:20	polinema2017	life2_out	2,544 s	0,002 s	Berhasil
9	life3.wav	6.204.800	2:20	polinema2017	life3_out	3,370 s	0,003 s	Berhasil
Citra								
10	all_1.wav	3.985.688	1:30	polinema2017	all_1_out	4,188 s	0,009 s	Berhasil
11	all_2.wav	3.985.688	1:30	polinema2017	all_2_out	36,311 s	0,158 s	Berhasil
12	all_3.wav	3.985.688	1:30	polinema2017	all_3_out	79,592 s	0,430 s	Berhasil
13	alone_1.wav	4.546.052	1:42	polinema2017	alone_1_out	4,509 s	0,009 s	Berhasil
14	alone_2.wav	4.546.052	1:42	polinema2017	alone_2_out	36,486 s	0,114 s	Berhasil
15	alone_3.wav	4.546.052	1:42	polinema2017	alone_3_out	79,846 s	0,422 s	Berhasil
16	life_1.wav	6.204.800	2:20	polinema2017	life_1_out	4,968 s	0,009 s	Berhasil
17	life_2.wav	6.204.800	2:20	polinema2017	life_2_out	37,260 s	0,116 s	Berhasil
18	life_3.wav	6.204.800	2:20	polinema2017	life_3_out	80,415 s	0,426 s	Berhasil

Dari hasil pengujian proses enkripsi pesan teks dan citra pada file audio berdasarkan pada tabel 6.6, sebanyak 18 kali pengujian didapatkan hasil: 9 pesan teks dan 9 file citra berhasil disisipkan. Sehingga, dari 18 kali pengujian enkripsi dan penyisipan, didapatkan file stego audio yang berhasil dibentuk sebanyak 18 file stego audio .wav.

Pengujian yang dilakukan berikutnya adalah proses dekripsi file stego audio. Proses dekripsi menggunakan kunci yang sama dengan kunci yang digunakan pada saat melakukan enkripsi. Dari hasil pengujian proses dekripsi pesan teks dan citra pada file stego audio berdasarkan pada tabel 6.7, sebanyak 18 kali pengujian didapatkan hasil: 9 pesan teks dan 9 file citra berhasil dilakukan ekstraksi dan dekripsi pesan. Dari proses dekripsi pesan teks dan citra pada file stego audio didapatkan hasil keluaran yang dapat dilihat pada tabel 6.8 dan 6.9.

Tabel 6.8 Hasil Keluaran Dekripsi Pesan Teks pada File Stego Audio

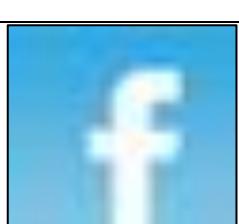
No	Teks Keluaran	Cipherteks	Pesan Teks	Panjang Pesan
1	all1_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349	Hallo, nama saya Binar Prihadmantyo.	36 Karakter
2	all2_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7 ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb2	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi.	106 Karakter
3	all3_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di	220 Karakter

No	Teks Keluaran	Cipherteks	Pesan Teks	Panjang Pesan
		c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7 ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb21dd9ca 2c687edf76bcbf9c133e46 77155a33cdab1a406ea7f2 31db8589028b6fa5f3f596 476b69662c06c66e4b83c 8b402606e1eb6d602d2b3 b6b8e92b72bdec2f07a79d 6102c917acb4e500d4bd3 eb0475a92ee7b6704c160 c443200c4df4467ca8bbc4 79dff29d591cd30b7d3948 e01084	Politeknik Negeri Malang, jurusan Teknologi Informasi. Usia saya 22 tahun, tinggal di Perum Permata Saxofone E- 6, Kelurahan Jatimulyo, Kecamatan Lowokwaru, Kota Malang.	
4	alone1_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb0 5e32042f6a14c2714e9f16 c05349	Hallo, nama saya Binar Prihadmantyo.	36 Karakter
5	alone2_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb0 5e32042f6a14c2714e9f16 c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7 ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb2	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi.	106 Karakter

No	Teks Keluaran	Cipherteks	Pesan Teks	Panjang Pesan
6	alone3_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7 ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb21dd9ca 2c687edf76bcf9c133e46 77155a33cdab1a406ea7f2 31db8589028b6fa5f3f596 476b69662c06c66e4b83c 8b402606e1eb6d602d2b3 b6b8e92b72bdec2f07a79d 6102c917acb4e500d4bd3 eb0475a92ee7b6704c16c 443200c4df4467ca8bbc47 9dff29d591cd30b7d3948e 01084	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi. Usia saya 22 tahun, tinggal di Perum Permata Saxofone E- 6, Kelurahan Jatimulyo, Kecamatan Lowokwaru, Kota Malang.	220 Karakter
7	life1_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349	Hallo, nama saya Binar Prihadmantyo.	36 Karakter
8	life2_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi.	106 Karakter

No	Teks Keluaran	Cipherteks	Pesan Teks	Panjang Pesan
		ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb2		
9	life3_out	f79782d01f4a658f68cc15 0a833dc3ddf8edfe1d4fb 5e32042f6a14c2714e9f16 c05349efc43832407ddbba ceb40fecda5ebc86119040 bf97decba3b670bd0d20ef 05d0939200e61575004a7 ba712c69ff266f3a3c113d d33aebbd0bdabed069733 e6a0ae8b83343fb21dd9ca 2c687edf76bc9f9c133e46 77155a33cdab1a406ea7f2 31db8589028b6fa5f3f596 476b69662c06c66e4b83c 8b402606e1eb6d602d2b3 b6b8e92b72bdec2f07a79d 6102c917acb4e500d4bd3 eb0475a92ee7b6704c160 c443200c4df4467ca8bbc4 79dff29d591cd30b7d3948 e01084	Hallo, nama saya Binar Prihadmantyo. Saya kuliah di Politeknik Negeri Malang, jurusan Teknologi Informasi. Usia saya 22 tahun, tinggal di Perum Permata Saxofone E- 6, Kelurahan Jatimulyo, Kecamatan Lowokwaru, Kota Malang.	220 Karakter

Tabel 6.9 Hasil Keluaran Dekripsi Citra pada File Stego Audio

No	Keluaran Citra	Ukuran File (byte)	Ukuran Pixel	Cipher Citra	Citra
1	all_1_out.jpg	571	30 x 30	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e21764f0 472ae1b29f86abe346 dc59e52463752.....	
2	all_2_out.jpg	6.839	350 x 350	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a51e2df261ba3f6 3cc945d4e2958.....	
3	all_3_out.jpg	15.315	250 x 260	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a41e2df260bb3e6 4cc945a4f2759.....	
4	alone_1_out.jpg	571	30 x 30	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e21764f0 472ae1b29f86abe346 dc59e52463752.....	
5	alone_2_out.jpg	6.839	350 x 350	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a51e2df261ba3f6 3cc945d4e2958.....	
6	alone_3_out.jpg	15.315	250 x 260	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a41e2df260bb3e6 4cc945a4f2759.....	

No	Keluaran Citra	Ukuran File (byte)	Ukuran Pixel	Cipher Citra	Citra
7	life_1_out.jpg	571	30 x 30	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e21764f0 472ae1b29f86abe346 dc59e52463752.....	
8	life_2_out.jpg	6.839	350 x 350	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a51e2df261ba3f6 3cc945d4e2958.....	
8	life_3_out.jpg	15.315	250 x 260	402e115c70760fa740 e7742bf15dbabcd8af 9773d1147e2176450 077a41e2df260bb3e6 4cc945a4f2759.....	

Berdasarkan hasil keluaran dari proses dekripsi pesan teks dan citra yang dijelaskan pada tabel 6.8 dan 6.9 dapat disimpulkan bahwa keluaran pesan teks dan citra yang telah didekripsi tidak mengalami perubahan, atau memiliki kondisi yang sama seperti sebelum dilakukan proses enkripsi.

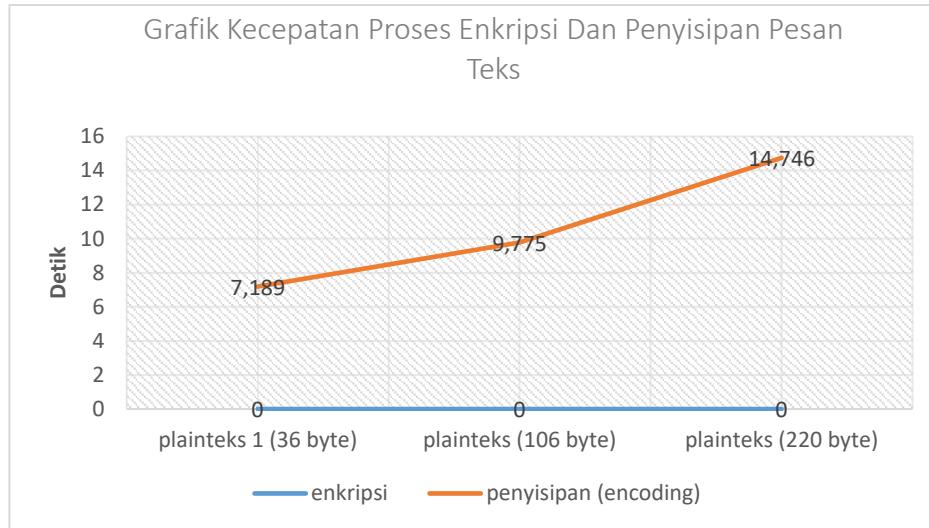
6.2.2 Pengujian Kecepatan Proses Enkripsi dan Dekripsi

Pengujian kecepatan enkripsi dan dekripsi dilakukan untuk mengetahui waktu yang dibutuhkan aplikasi dalam melakukan proses enkripsi dan penyisipan pesan, serta proses ekstraksi dan dekripsi pesan. Pengujian dilakukan menggunakan sampel file uji audio, citra dan pesan teks dengan ukuran yang berbeda-beda.

Berdasarkan tabel 6.6 pada kolom ‘waktu enkripsi’ dan ‘waktu penyisipan (*encoding*)’, didapatkan hasil waktu yang berbeda-beda dalam melakukan proses enkripsi dan penyisipan pesan. Kecepatan proses enkripsi dan penyisipan pesan bergantung pada ukuran pesan yang disisipkan. Semakin besar ukuran pesan, maka

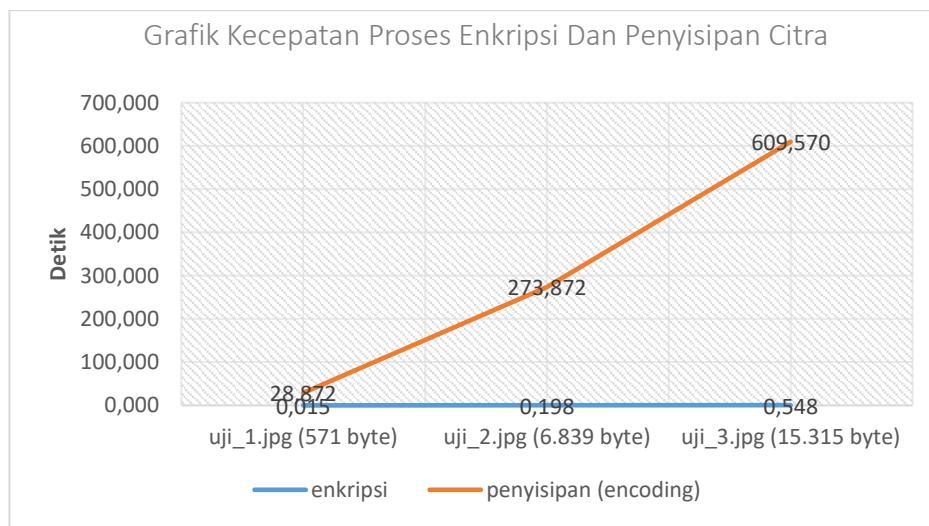
waktu yang dibutuhkan untuk melakukan proses enkripsi dan penyisipan semakin lama.

Dari pengujian yang telah dilakukan, rata-rata kecepatan proses enkripsi dan penyisipan pesan teks pada file audio, dapat digambarkan dalam grafik seperti pada gambar 6.1.



Gambar 6.1 Grafik Kecepatan Proses Enkripsi dan Penyisipan Pesan Teks

Sedangkan rata-rata kecepatan proses enkripsi dan penyisipan citra pada file audio, dapat digambarkan dalam grafik seperti pada gambar 6.2.

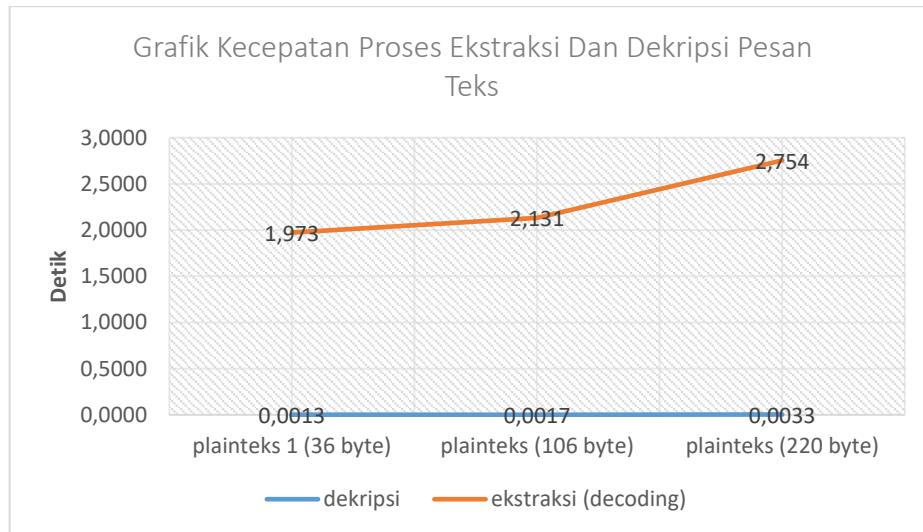


Gambar 6.2 Grafik Kecepatan Proses Enkripsi dan Penyisipan Citra

Berdasarkan tabel 6.7 pada kolom ‘waktu ekstraksi’ dan ‘waktu dekripsi’, didapatkan hasil waktu yang berbeda-beda dalam melakukan proses ekstraksi dan

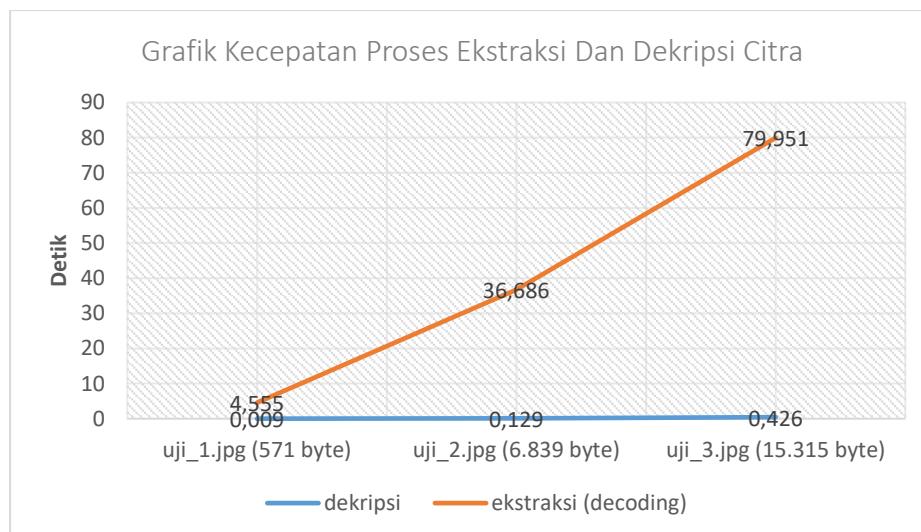
dekripsi pesan. Sama seperti proses enkripsi dan penyisipan pesan, kecepatan proses ekstraksi dan dekripsi pesan juga bergantung pada ukuran pesan yang disisipkan. Semakin besar ukuran pesan, maka waktu yang dibutuhkan untuk melakukan proses ekstraksi dan dekripsi pesan semakin lama.

Dari pengujian yang telah dilakukan, kecepatan proses ekstraksi dan dekripsi pesan teks dapat digambarkan dalam grafik seperti pada gambar 6.3.



Gambar 6.3 Grafik Kecepatan Proses Ekstraksi dan Dekripsi Pesan Teks

Sedangkan rata-rata kecepatan proses ekstraksi dan dekripsi citra pada file audio, dapat digambarkan dalam grafik seperti pada gambar 6.4.



Gambar 6.4 Grafik Kecepatan Proses Ekstraksi dan Dekripsi Citra

6.2.3 Pengujian Serangan Stego Audio

Terdapat berbagai serangan yang dapat dilakukan pada stego audio, seperti melakukan pemotongan durasi (*cropping*), membalik (*reversing*), dan mengubah amplitudo audio. Pengujian dilakukan dengan melakukan serangan pada stego audio, kemudian dilakukan proses dekripsi dengan menggunakan kunci yang sesuai. Stego audio yang digunakan adalah all_1.wav. Kunci yang digunakan saat melakukan enkripsi sama dengan kunci yang digunakan saat melakukan dekripsi, yaitu ‘polinema2017’.

Tabel 6.10 Pengujian Serangan Stego Audio

Pengujian Serangan Stego Audio				
No	Serangan Stego Audio	Stego Audio	Hasil Serangan Stego Audio	Hasil Keluaran
1.	Memotong durasi stego audio 5 detik dari belakang	all_1.wav 3.985.688 byte / 1:30	all_1.wav 3.753.928 byte / 1:25	Berhasil 
2.	Memotong durasi stego audio 5 detik dari depan	all_1.wav 3.985.688 byte / 1:30	all_1.wav 3.765.688 byte / 1:25	Gagal
3.	Reverse	all_1.wav 3.985.688 byte / 1:30	all_1.wav 3.986.188 byte / 1:30	Gagal
4.	Mengubah amplitudo +1.5 dB	all_1.wav 3.985.688 byte / 1:30	all_1.wav 3.986.188 byte / 1:30	Gagal

No	Serangan Stego Audio	Stego Audio	Hasil Serangan Stego Audio	Hasil Keluaran
5.	Mengubah amplitudo -1.5 dB	all_1.wav 3.985.688 byte / 1:30	all_1.wav 3.986.188 byte / 1:30	Gagal

Pada keseluruhan uji coba serangan yang ditunjukkan pada tabel 6.10, hasil yang didapatkan membuktikan bahwa stego audio tidak tahan terhadap serangan. Hal tersebut terjadi karena adanya perubahan *byte* pesan yang terdapat pada *byte* data stego audio yang dilakukan serangan, sehingga pesan yang disisipkan tidak dapat diekstraksi dan didekripsi. Namun apabila serangan yang dilakukan tidak merubah *byte* pesan yang terdapat pada *byte* data stego audio, maka pesan yang disisipkan dapat diekstraksi dan didekripsi, seperti pada pengujian serangan yang pertama yaitu memotong durasi stego audio 5 detik dari belakang.

6.2.4 Pengujian Kualitas Audio

Proses enkripsi dan penyisipan pesan kedalam file audio menghasilkan suatu file stego audio yang terdapat *noise*. Terjadinya *noise* diakibatkan perubahan bit yang dilakukan pada proses penyisipan pesan. *Noise* pada stego audio dapat didengar menggunakan indra pendengaran manusia secara langsung. Pengukuran *noise* stego audio dilakukan dengan menggunakan PSNR (*Peak Signal to Noise Ratio*). Secara matematis perhitungan *noise* akan memakai perhitungan nilai PSNR dengan nilai minimal 30db. Perhitungan PSNR ini dilakukan dengan memakai rumus persamaan sebagai berikut.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{P_1^2}{P_1^2 + P_0^2 - 2P_1P_0} \right) \quad [9]$$

Dimana P_1 adalah kekuatan sinyal audio setelah proses penyisipan pesan dan P_0 adalah kekuatan sinyal audio awal. Jika nilai PSNR < 30 dB, maka dapat dikatakan bahwa kualitas stego audio buruk dan menimbulkan *noise* yang sangat jelas terdengar oleh indra pendengaran manusia.

Pengujian PSNR menggunakan 3 file uji audio, yang masing-masing audio disisipi file uji citra dengan ukuran file yang berbeda-beda. Hasil pengujian PSNR dijabarkan pada tabel berikut.

Tabel 6.11 Pengujian PSNR

Audio Asli	Pesan	Stego Audio	Pengujian	
			Subjektif	PSNR
	uji_1.jpg (571 byte)	all_1.wav (3.985.688 byte)	Baik	45,15038819 dB
	uji_2.jpg (6.839 byte)	all_2.wav (3.985.688 byte)	Baik	45,16826506 dB
	uji_3.jpg (15.315 byte)	all_3.wav (3.985.688 byte)	Baik	45,14382346 dB
	uji_1.jpg (571 byte)	alone_1.wav (4.546.052 byte)	Baik	46,46977211 dB
	uji_2.jpg (6.839 byte)	alone_2.wav (4.546.052 byte)	Baik	45,54895838 dB
	uji_3.jpg (15.315 byte)	alone_3.wav (4.546.052 byte)	Baik	45,52715661 dB

Audio Asli	Pesan	Stego Audio	Pengujian	
			Subjektif	PSNR
	uji_1.jpg (571 byte)	life_1.wav (6.204.800 byte)	Baik	49,80236613 dB
	uji_2.jpg (6.839 byte)	life_2.wav (6.204.800 byte)	Baik	49,17697854 dB
	uji_3.jpg (15.315 byte)	life_3.wav (6.204.800 byte)	Baik	49,14184025 dB

Hasil pengujian yang ditunjukkan pada tabel 6.11, dari 9 kali pengujian PSNR stego audio, didapatkan nilai $PSNR > 30$ dB. Sehingga dapat disimpulkan bahwa kualitas stego audio yang dihasilkan baik dan tidak menghasilkan *noise* yang dapat terdengar oleh indra pendengaran manusia.

BAB VII. KESIMPULAN

Bab ini menjelaskan tentang kesimpulan yang didapat pada saat proses penggerjaan skripsi melalui pengujian yang dilakukan dan analisa yang digunakan dalam penelitiannya. Bab ini juga berisi saran yang bisa dilakukan untuk penelitian di masa yang akan datang

7.1 Kesimpulan

Kesimpulan dari skripsi yang berjudul: “IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI” adalah sebagai berikut:

1. Langkah pengamanan pesan pada aplikasi ini ialah, pesan teks dan citra dienkripsi terlebih dahulu menggunakan algoritma kriptografi RC4. Selanjutnya pesan yang telah terenkripsi disisipkan ke dalam file audio menggunakan metode steganografi 2LSB dan menghasilkan file stego audio. Berdasarkan hasil pengujian enkripsi dan penyisipan pesan pada file audio, sebanyak 18 kali pengujian penyisipan didapatkan persentase keberhasilan sebesar 100%. Keberhasilan penyisipan pesan bergantung pada ukuran pesan yang disisipkan, apabila ukuran pesan yang disisipkan $\leq 4\%$ ukuran audio, maka penyisipan dapat dilakukan. Namun apabila ukuran pesan yang disisipkan $>4\%$ ukuran audio, maka pesan tidak dapat disisipkan.
2. Proses dekripsi pesan dinyatakan berhasil apabila pesan yang dienkripsi dan dilakukan penyisipan, tidak mengalami perubahan ketika di dekripsi. Pengujian kecocokan antara pesan teks dan citra sebelum dienkripsi dan setelah didekripsi, didapatkan tingkat keberhasilan sebesar 100% dari 18 kali pengujian dekripsi pesan.
3. Kecepatan pemrosesan enkripsi dan penyisipan pesan bergantung pada ukuran pesan yang disisipkan, hal itu juga berlaku pada ekstraksi dan dekripsi pesan. Semakin besar ukuran pesan yang disisipkan pada file audio, maka semakin lama waktu yang diperlukan untuk melakukan proses enkripsi dan penyisipan pesan, serta proses ekstraksi dan dekripsi pesan. Pada proses yang pertama yaitu enkripsi dan penyisipan pesan, untuk

enkripsi pesan teks memerlukan waktu rata-rata 0 detik dan dalam proses penyisipan memerlukan waktu rata-rata 10,570 detik. Selanjutnya proses enkripsi citra memerlukan waktu rata-rata 0,254 detik dan proses penyisipan memerlukan waktu rata-rata 304,105 detik. Kemudian pada pengujian selanjutnya yaitu proses ekstraksi dan dekripsi pesan, untuk dekripsi pesan teks memerlukan waktu rata-rata 0,0021 detik dan 2,286 detik waktu rata-rata yang digunakan untuk ekstraksi. Selanjutnya, proses deskripsi dan ekstrasi citra masing-masing memerlukan waktu rata-rata selama 0,188 detik dan 40,397 detik.

4. Stego audio tidak tahan terhadap serangan yang menyebabkan perubahan nilai *byte* pesan pada file stego. Hal ini dibuktikan dengan gagalnya proses ekstraksi dan dekripsi pesan.
5. Stego audio yang dihasilkan memiliki kualitas baik. Hal ini dibuktikan dengan nilai PSNR stego audio > 30 dB, sehingga tidak menimbulkan *noise* yang dapat didengarkan oleh indra pendengaran manusia secara langsung.

7.2 Saran

Dalam pembuatan skripsi yang berjudul: “IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI”, masih banyak hal yang dapat dikembangkan untuk penelitian selanjutnya, antara lain :

1. Pengembangan pesan yang diamankan menggunakan berbagai jenis tipe file, seperti file dokumen, audio, dan video.
2. Pengembangan keamanan penyisipan dengan melakukan pengacakan pola penyisipan pesan, misal menggunakan metode *Random Byte Position Encoding*.
3. Penyisipan pesan pada file pembawa pesan menggunakan metode yang tidak membatasi ukuran pesan yang disisipkan, misal menggunakan metode *End of File* (EOF).
4. Pengembangan aplikasi berbasis web dan android.

Demikian saran yang dapat penulis berikan, semoga saran tersebut dapat dijadikan sebagai bahan masukan yang dapat bermanfaat bagi penulis khususnya dan umumnya bagi akademisi di kemudian hari.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. 2006. “*Kriptografi*”. Informatika Bandung.
- [2] Eko Krist Setyono dan M.A. Ineke Pakareng. 2014. “*Perancangan dan Implementasi Aplikasi Steganografi Citra Digital dengan Metode 2LSB*”. Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
- [3] Sekarsari, Galuh Adjeng. 2015. “*Analisa Algoritma Kriptografi RC4 Pada Enkripsi Citra Digital*”. Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
- [4] Ariyus, Dony. 2008. “*Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi*”. STMIK AMIKOM.
- [5] Munir, Rinaldi. 2004. “*Pengolahan Citra Digital Dengan Pendekatan Algoritmik*”. Informatika Bandung.
- [6] Binanto, Iwan. 2010. “*Multimedia Digital – Dasar Teori dan Pengembangannya*”. ANDI
- [7] Schneier, Bruce. 1996. “*Applied Cryptography: Protocols, Algorithms and Source Code in C*”. John Wiley & Sons.
- [8] O'Brien, James. 1999. “*Management Information Systems: Managing Information Technology in The Internet-worked Enterprise Forth Edition*”. New York: McGraw-Hill
- [9] Yoga Bagus, Wahyu Suadi, dan Baskoro Adi. 2012. “*Implementasi Kriptografi dan Steganografi pada Audio Menggunakan Metode DES dan Parity Coding*”. Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

LAMPIRAN

Lampiran 1. Kode Program

Enkripsi teks RC4

```
Public Function textEncrypt(ByVal inputText As String, ByVal key  
As String)  
    Dim s As Integer() = New Integer(255) {}  
    Dim k As Integer() = New Integer(255) {}  
    Dim kE As Integer  
    Dim input As Integer  
    Dim output As New Integer  
    Dim outputText As String  
  
    For z = 0 To 255  
        s(z) = z  
        k(z) = Asc(key(z Mod key.Length))  
    Next  
  
    Dim j As Integer = 0  
    Dim tmp As Integer = 0  
  
    For n = 0 To 255  
        j = (j + s(n) + k(n)) Mod 256  
  
        tmp = s(n)  
        s(n) = s(j)  
        s(j) = tmp  
    Next  
  
    Dim i As Integer = 0  
    j = 0  
    tmp = 0  
  
    For n = 1 To inputText.Length  
        i = (i + 1) Mod 256  
        j = (j + s(i)) Mod 256  
  
        tmp = s(i)  
        s(i) = s(j)  
        s(j) = tmp  
  
        kE = s((s(i) + s(j)) Mod 256)  
  
        input = Asc(inputText(n - 1))  
        output = input Xor kE  
  
        If Hex(output).Length = 1 Then  
            outputText += "0" & Hex(output).ToLower  
        Else  
            outputText += Hex(output).ToLower  
        End If  
    Next  
    Return outputText  
End Function
```

Dekripsi teks RC4

```
Public Function textDecrypt(ByVal inputText As String, ByVal key As String)
    Dim s As Integer() = New Integer(255) {}
    Dim k As Integer() = New Integer(255) {}
    Dim kE As Integer
    Dim inputText_ As String
    Dim input As Integer
    Dim output As New Integer
    Dim outputText As String

    For z = 1 To inputText.Length Step 2
        inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
    Next

    For z = 0 To 255
        s(z) = z
        k(z) = Asc(key(z Mod key.Length))
    Next

    Dim j As Integer = 0
    Dim tmp As Integer = 0

    For n = 0 To 255
        j = (j + s(n) + k(n)) Mod 256

        tmp = s(n)
        s(n) = s(j)
        s(j) = tmp
    Next

    Dim i As Integer = 0
    j = 0
    tmp = 0

    For n = 1 To inputText_.Length
        i = (i + 1) Mod 256
        j = (j + s(i)) Mod 256

        tmp = s(i)
        s(i) = s(j)
        s(j) = tmp

        kE = s((s(i) + s(j)) Mod 256)

        input = Asc(inputText_(n - 1))
        output = input Xor kE

        outputText += Chr(output)
    Next
    Return outputText
End Function
```

Enkripsi citra RC4

```
Public Function imageEncrypt(ByVal inputText As String, ByVal key As String)
    Dim s As Integer() = New Integer(255) {}
    Dim k As Integer() = New Integer(255) {}
    Dim kE As Integer
    Dim inputText_ As String
    Dim input As Integer
    Dim output As New Integer
    Dim outputText As String

    For z = 1 To inputText.Length Step 2
        inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
    Next

    For z = 0 To 255
        s(z) = z
        k(z) = Asc(key(z Mod key.Length))
    Next

    Dim j As Integer = 0
    Dim tmp As Integer = 0

    For n = 0 To 255
        j = (j + s(n) + k(n)) Mod 256

        tmp = s(n)
        s(n) = s(j)
        s(j) = tmp
    Next

    Dim i As Integer = 0
    j = 0
    tmp = 0

    For n = 1 To inputText_.Length
        i = (i + 1) Mod 256
        j = (j + s(i)) Mod 256

        tmp = s(i)
        s(i) = s(j)
        s(j) = tmp

        kE = s((s(i) + s(j)) Mod 256)
        input = Asc(inputText_(n - 1))
        output = input Xor kE

        If Hex(output).Length = 1 Then
            outputText += "0" & Hex(output).ToLower
        Else
            outputText += Hex(output).ToLower
        End If
    Next
    Return outputText
End Function
```

Dekripsi citra RC4

```
Public Function imageDecrypt(ByVal inputText As String, ByVal key As String)
    Dim s As Integer() = New Integer(255) {}
    Dim k As Integer() = New Integer(255) {}
    Dim kE As Integer
    Dim inputText_ As String
    Dim input As Integer
    Dim output As New Integer
    Dim outputText As String

    For z = 1 To inputText.Length Step 2
        inputText_ += Chr(myMod.hexToDec(Mid$(inputText, z,
2)))
    Next

    For z = 0 To 255
        s(z) = z
        k(z) = Asc(key(z Mod key.Length))
    Next

    Dim j As Integer = 0
    Dim tmp As Integer = 0

    For n = 0 To 255
        j = (j + s(n) + k(n)) Mod 256

        tmp = s(n)
        s(n) = s(j)
        s(j) = tmp
    Next

    Dim i As Integer = 0
    j = 0
    tmp = 0

    For n = 1 To inputText_.Length
        i = (i + 1) Mod 256
        j = (j + s(i)) Mod 256

        tmp = s(i)
        s(i) = s(j)
        s(j) = tmp

        kE = s((s(i) + s(j)) Mod 256)
        input = Asc(inputText_(n - 1))
        output = input Xor kE

        If Hex(output).Length = 1 Then
            outputText += "0" & Hex(output).ToLower
        Else
            outputText += Hex(output).ToLower
        End If
    Next
    Return outputText
End Function
```

Penyisipan (*encoding*) 2LSB

```
Public Sub encoding(ByVal message As String, ByVal audio As String, ByVal savefile As String, ByVal type As String)
    Dim Text_audio As String
    Dim Text_string As String
    Dim fileReader As String
    Dim oReturn As New StringBuilder

    Dim sa As New IO.StreamReader(audio)
    Dim ss As New IO.StreamReader(Message)
    Text_audio = sa.ReadToEnd
    Text_string = ss.ReadToEnd
    sa.Close()
    ss.Close()

    fileReader = Text_audio.Substring(0, 400)

    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
    "_stego.txt", fileReader, True)

    Dim j As Integer = 400
    Dim prog As Integer

    For i As Integer = 0 To Text_string.Length - 1 Step 2
        fileReader = Text_audio.Substring(j, 8)
        fileReader = fileReader.Remove(6, 2) +
    Text_string.Substring(i, 2)
        j = j + 8
        fileReader = fileReader + Text_audio.Substring(j, 40)
        j = j + 40
        prog = i
        oReturn.Append(fileReader)
        Console.WriteLine(fileReader)
    Next
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
    "_stego.txt", oReturn.ToString, True)

    For z = 0 To 11
        prog += 1
        fileReader = Text_audio.Substring(j, 8)
        fileReader = fileReader.Remove(6, 2) + "11"
        j = j + 8
        fileReader = fileReader + Text_audio.Substring(j, 40)
        j = j + 40
        My.Computer.FileSystem.WriteAllText("temp\" + savefile +
    + "_stego.txt", fileReader, True)
    Next

    caps = (j / Text_audio.Length) * 100

    Dim k As Integer = Text_audio.Length - j
    fileReader = Text_audio.Substring(j, k)
    My.Computer.FileSystem.WriteAllText("temp\" + savefile +
    "_stego.txt", fileReader, True)

End Sub
```

Ekstraksi (*decoding*) 2LSB

```
Public Function decoding(ByVal audio_stego As String, ByVal savefile As String, ByVal type As String)
'get file
    Dim Text_stego As String
    Dim ss As New IO.StreamReader(audio_stego)

    Text_stego = ss.ReadToEnd
    ss.Close()

    Dim a As String
    Dim b As String = ""
    Dim filereader As New StringBuilder
    Dim Separator As String = ("")
    Dim j As Integer = 400
    Dim k As Integer = 16

    For i As Integer = 0 To Text_stego.Length / 8 - 1
        a = Text_stego.Substring(j, 8)
        filereader.Append(Strings.Right(a, 2))
        filereader.Append(Separator)
        'c += Strings.Right(a, 2)

        If filereader.Length Mod 8 = 0 And filereader.Length >
16 Then
            b = filereader.ToString.Substring(k - 16, 24)
            If b = "111111111111111111111111" Then
                i = Text_stego.Length / 8 - 1
            End If
            k += 8
            Console.WriteLine(b)
        End If

        j = j + 48
    Next

    Dim str As String
    str = filereader.ToString
    str = str.Remove(str.Length - 24)

    File.WriteAllText("temp2\" + savefile + "_2lsb.txt", str)
    Return "temp2\" + savefile + "_2lsb.txt"
    Return str
End Function
```

Konversi audio ke biner

```
Public Function audioToBinary(ByVal filename As String, ByVal savefile As String, ByVal type As String)
    Dim filebytes As Byte() = File.ReadAllBytes(filename)
    Dim oReturn As New StringBuilder
    Dim Separator As String = ("")
    For index As Integer = 0 To filebytes.Length - 1
        oReturn.Append(Convert.ToString(filebytes(index),
2).PadLeft(8, "0"))
        oReturn.Append(Separator)
```

```

        Next
        If type = "encode" Then
            File.WriteAllText("temp\" + savefile +
"_audio_bin.txt", oReturn.ToString)
            Return "temp\" + savefile + "_audio_bin.txt"
        Else
            File.WriteAllText("temp2\" + savefile +
"_audio_bin.txt", oReturn.ToString)
            Return "temp2\" + savefile + "_audio_bin.txt"
        End If
    End Function

```

Konversi biner ke audio

```

Public Sub binaryToAudio(ByVal filename As String, ByVal savefile
As String, ByVal audioFormat As String)
    Dim Text As String
    Dim sr As New IO.StreamReader(filename)
    Text = sr.ReadToEnd
    sr.Close()
    Dim Characters As String =
System.Text.RegularExpressions.Regex.Replace(Text, "[^01]", "")
    Dim ByteArray((Characters.Length / 8) - 1) As Byte
    Dim oReturn As New StringBuilder
    Dim Separator As String = ("")
    For Index As Integer = 0 To ByteArray.Length - 1
        ByteArray(Index) =
Convert.ToByte(Characters.Substring(Index * 8, 8), 2)
        oReturn.Append(ByteArray(Index).ToString)
        oReturn.Append(Separator)
    Next
    If audioFormat = "wav" Then
        File.WriteAllBytes("output\" + savefile + ".wav",
ByteArray)
    Else
        File.WriteAllBytes("output\" + savefile + ".mp3",
ByteArray)
    End If
End Sub

```

Konversi byte ke citra

```

Public Function byteToImage(ByVal byteArrayIn As Byte()) As Image
    Using mStream As New MemoryStream(byteArrayIn)
        Return Image.FromStream(mStream)
    End Using
End Function

```

Konversi citra ke byte

```

Public Function imageToByte(ByVal img As Image) As Byte()
    Using mStream As New MemoryStream()
        img.Save(mStream, img.RawFormat)
        Return mStream.ToArray()
    End Using
End Function

```

Lampiran 2. Identitas Penulis

PROFIL PENULIS



Nama Lengkap : Binar Prihadmantyo
Nomor Induk Mahasiswa: 1341180029
Jurusan : Teknologi Informasi
Program Studi : Teknik Informatika
Tempat Tanggal Lahir : Trenggalek, 12 Mei 1995
Alamat Asal : Jalan KH Agus Salim 16, Trenggalek
Agama : Islam
No. Telepon : 081259757493
Email : prihadmantyobinar@gmail.com

Riwayat Pendidikan :

2001 – 2007 : SDN 3 Ngantru, Trenggalek
2007 – 2010 : SMPN 1 Trenggalek
2010 – 2013 : SMAN 1 Trenggalek
2013 – 2017 : D4 – Teknik Informatika, Politeknik Negeri Malang

Lampiran 3. Lembar Bimbingan Pembimbing 1



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI: 73

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI

Nama : BINAR PRIHADMANTYO NIM : 1341180029

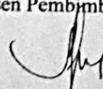
No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	7/3 2017	Progress enkripsi/dekripsi teks	Rita	r.
2.	14/3 2017	Progress enkripsi/dekripsi citra	Rita	r.
3.	21/3 2017	Progress laporan Skripsi 1-3	Rita	r.
4.	30/3 2017	Progress laporan Skripsi 1-4	Rita	r.
5.	6/4 2017	Progress laporan Skripsi 1-5	Rita	r.
6.	13/4 2017	Progress manual dan implementasi sistem, Publikasi pengujian	Rita	r.
7.	20/4 2017	Progress progres aplikasi enkripsi dan ekstrak aplikasi	Rita	r.
8.	26/4 2017	Progress encoding wav	Rita	r.
9.	4/5 2017	Progress encoding mp3	Rita	r.
10.	9/5 2017	Progress decoding dan pembuatan enkripsi citra	Rita	r.
11.	16/5 2017	Progress Finishing program	Rita	r.
12.	21/5 2017	Progress Finishing laporan	Rita	r.
13.	1/6 2017	ALL mayu ujian	Rita	r.
14.	5/6 2017	Bimbingan laporan	Rita	r.
15.	20/6 2017	Bimbingan laporan	Rita	r.
16.				
17.				
18.				
19.				

Malang, 30 Mei 2017.
Dosen Pembimbing Skripsi,

Ely Setyo Astuti, ST., MT
NIP.197605152009122001

Lampiran 4. Lembar Bimbingan Pembimbing 2

	KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI PROGRAM STUDI TEKNIK INFORMATIKA Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122			
NO SKRIPSI: 73				
LEMBAR BIMBINGAN SKRIPSI 2016/2017				
JUDUL : IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRafi AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI				
Nama : BINAR PRIHADMANTYO NIM : 1341180029				
No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	8/3/2017	perancangan antarmuka program, perancangan program		
2.	15/3/2017	bab 1-3 data file audio wav		
3.	22/3/2017	print bab 1-4, mewujudkan analisa bab 1-4, perancangan bab 1-4		
4.	29/3/2017	bab 1-4 perbaikan laporan		
5.	5/4/2017	Perbaikan bab 1, segmenasi, dafatr astawa vok 5, setting suara degradasi suara		
6.	12/4/2017	perbaikan taks, progress coding		
7.	19/4/2017	Progress coding		
8.	26/4/2017	Progress coding		
9.	3/5/2017	-		
10.	10/5/2017	-		
11.	17/5/2017	Final coding		
12.	24/5/2017	laporan bab VI		
13.	31/5/2017	final Laporan		
14.	5/6/2017	Revisi Laporan		
15.	22/6/2017	Review Laporan		
16.				
17.				
18.				
19.				

Malang, 20 Mei 2017....
 Dosen Pembimbing Skripsi,

Meyti Eka Apriyani, ST., MT.

Lampiran 5. Lembar Persetujuan Maju Ujian

	<p>KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI PROGRAM STUDI TEKNIK INFORMATIKA JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122</p>																																																																					
		NO SKRIPSI: 73																																																																				
LEMBAR PERSETUJUAN MENGIKUTI UJIAN SKRIPSI 2016/2017 PROGRAM STUDI TEKNIK INFORMATIKA																																																																						
NAMA	: BINAR PRIHADMANTYO		NIM / KELAS	: 1341180029 / TI-4A																																																																		
JUDUL SKRIPSI	: IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI		NIP	: 19760515 200912 2 001																																																																		
PEMBIMBING	: 1. ELY SETYO ASTUTI, ST., MT. 2. MEYTI EKA APRIYANI, ST., MT		NIP	: 19760515 200912 2 001																																																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">No.</th> <th style="width: 50%;">Uraian / Bab</th> <th style="width: 10%;">Dislesaikan</th> <th style="width: 15%;">Pembimbing 1</th> <th style="width: 15%;">Pembimbing 2</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>PENDAHULUAN</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>2.</td> <td>LANDASAN TEORI</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>3.</td> <td>METODOLOGI PENELITIAN</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>4.</td> <td>ANALISIS DAN PERANCANGAN</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>5.</td> <td>IMPLEMENTASI</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>6.</td> <td>PENGUJIAN DAN PEMBAHASAN</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>7.</td> <td>KESIMPULAN DAN SARAN</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>8.</td> <td>BAGIAN AKHIR</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>8.</td> <td>- Daftar Pustaka - Lampiran (Isi lampiran disesuaikan dengan judul laporan akhir) - Profile Penulis (Riwayat Penulis)</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>9.</td> <td>Hardware/Software</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>9.</td> <td>- Didemokan di depan pembimbing</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> <tr> <td>10</td> <td>Draft Makalah</td> <td style="text-align: center;">✓</td> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> </tbody> </table>			No.	Uraian / Bab	Dislesaikan	Pembimbing 1	Pembimbing 2	1.	PENDAHULUAN	✓			2.	LANDASAN TEORI	✓			3.	METODOLOGI PENELITIAN	✓			4.	ANALISIS DAN PERANCANGAN	✓			5.	IMPLEMENTASI	✓			6.	PENGUJIAN DAN PEMBAHASAN	✓			7.	KESIMPULAN DAN SARAN	✓			8.	BAGIAN AKHIR	✓			8.	- Daftar Pustaka - Lampiran (Isi lampiran disesuaikan dengan judul laporan akhir) - Profile Penulis (Riwayat Penulis)	✓			9.	Hardware/Software	✓			9.	- Didemokan di depan pembimbing	✓			10	Draft Makalah	✓			<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="margin: 0;">DISETUJUI UNTUK DAPAT MAJU UJIAN SETELAH HASIL KARYA DINILAI LAYAK SERTA HASIL UJI SESUAI DENGAN SPESIFIKASI YANG DIRENCANAKAN</p> <p style="margin: 0;">Pembimbing I Ely Setyo Astuti, ST., MT NIP. 19760515 200912 2 001</p> <p style="margin: 0;">Pembimbing II Meyti Eka Apriyani, ST., MT NIP. 19790313 200812 1 002</p> </div>		
No.	Uraian / Bab	Dislesaikan	Pembimbing 1	Pembimbing 2																																																																		
1.	PENDAHULUAN	✓																																																																				
2.	LANDASAN TEORI	✓																																																																				
3.	METODOLOGI PENELITIAN	✓																																																																				
4.	ANALISIS DAN PERANCANGAN	✓																																																																				
5.	IMPLEMENTASI	✓																																																																				
6.	PENGUJIAN DAN PEMBAHASAN	✓																																																																				
7.	KESIMPULAN DAN SARAN	✓																																																																				
8.	BAGIAN AKHIR	✓																																																																				
8.	- Daftar Pustaka - Lampiran (Isi lampiran disesuaikan dengan judul laporan akhir) - Profile Penulis (Riwayat Penulis)	✓																																																																				
9.	Hardware/Software	✓																																																																				
9.	- Didemokan di depan pembimbing	✓																																																																				
10	Draft Makalah	✓																																																																				
FRM.RTL.01.49.04																																																																						

Lampiran 6. Form Revisi Penguji 1

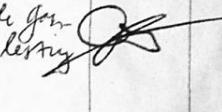
**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA**
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

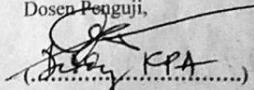


FORM REVISI SKRIPSI

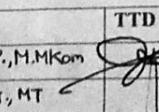
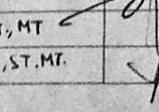
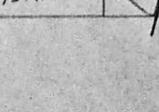
No. Skripsi : 73

Nama Mahasiswa : BINAR PRIHADMANTYO **NIM** : 1341180029
Tanggal Ujian : 17.07.2017
Judul : IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI

NO	SARAN PERBAIKAN	PARAF
1.	<i>Juster q.8 tambahka penambah file jas... file suara , gambar & lirik program</i>	

Malang, 17-07-2017.
Dosen Penguji,


FORM VERIFIKASI:
Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	Ir. Dedy Kubianto P., M.MKom		25/7/2017
Pembimbing 1	Ely Setyo Achut, ST, MT		27/7/2017
Pembimbing 2	Meyti Eka Apriyani, ST, MT		26/7/2017

FRM.RTL.01.35.03

Lampiran 7. Form Revisi Penguji 2



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
 POLITEKNIK NEGERI MALANG
 JURUSAN TEKNOLOGI INFORMASI
 PROGRAM STUDI TEKNIK INFORMATIKA

JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



No. Skripsi : 73

FORM REVISI SKRIPSI

Nama Mahasiswa : BINAR PRIHADMANTYO NIM : 1341180029
 Tanggal Ujian : 17.07.2017.....
 Judul : IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN
 METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM
 KEAMANAN INFORMASI

NO	SARAN PERBAIKAN	PARAF
1.	Benahi aplikasi agar proses enkripsi/denkripsi mulai lagi dari awal (bersih).	<i>[Signature]</i>
2.	flowchart baik dekripsi & enkripsi dibenahi : <ul style="list-style-type: none"> - image + audio - text + audio - Bentuk per modul .	<i>[Signature]</i>

Malang, 17-07-2017

Dosen Penguji

(Signature)

FORM VERIFIKASI:

Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	Lugman Affandi, S.Kom, MM	<i>[Signature]</i>	25-07-2017
Pembimbing 1	Ely Setyo Astuti, ST, MT	<i>[Signature]</i>	27 July 2017
Pembimbing 2	Meyti Eka Apriyani, ST, MT.	<i>[Signature]</i>	26-07-2017

FRM.RTI.01.35.03

Lampiran 8. Form Verifikasi Abstrak Bahasa Inggris dan Tata Tulis Buku Skripsi

		KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI PROGRAM STUDI TEKNIK INFORMATIKA JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122		
No. Skripsi : 73				
FORM VERIFIKASI ABSTRAK BAHASA INGGRIS DAN TATA TULIS BUKU SKRIPSI				
<p>Nama Mahasiswa 1 : BINAR PRIHADMANTYO NIM : 1341180029 Tanggal Ujian : 17 Juli 2017 Judul : IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN METODE STEGANOGRAFI AUDIO 2LSB PADA SISTEM KEAMANAN INFORMASI</p>				
NO	BAGIAN YANG DIVERIFIKASI	NAMA VERIFIKATOR	TANGGAL VERIFIKASI	TTD
1	Abstrak Berbahasa Inggris	Aulia Neurma P, S.S., M.A.	9 Agustus 2017	
2	Tata Tulis Buku Skripsi	Meyti E. A	24 Agustus 2017	

FRM.RTI.01.46.01