semanTIK, Vol.2, No.1, Jan-Jun 2016, pp. 349-360
ISSN : 2502-8928 (Online)

PENINGKATAN KEAMANAN DATA RAHASIA MENGGUNAKAN KRIPTOGRAFI *DATA ENCRYPTION STANDARD* PADA STEGANOGRAFI *AUDIO* BERFORMAT *MP3*

Afriyadin [*1], Sutardi [2], LM. Fid Aksara [3]
[*1,2,3]Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail: [*1] **afri.adin@gmail.com** , [2] sutardi_hapal@yahoo.com , [3]fid_laode@yahoo.com

Abstrak

Informasi dapat menjadi sesuatu yang sangat berharga dan perlu dijaga kerahasiaannya. Salah satu solusi dalam mengamankan informasi adalah dengan kriptografi dan steganografi. Steganografi merupakan ilmu dan seni yang mempelajari penyembunyian data rahasia pada suatu media sedemikian rupa sehingga keberadaannya tidak terdeteksi oleh pihak lain yang tidak berhak atas informasi tersebut, sedangkan kriptografi adalah teknik menyandikan (enkripsi) sebuah data rahasia menjadi data tersandi yang tidak dimengerti.

Metode steganografi *Least* Significant *Bit* (LSB) atau *low bit coding* merupakan metode yang sederhana dalam proses menyembunyikan data, yaitu dengan cara mengganti *bit* yang kurang penting/*least significant bit* dari setiap *sampling point file* MP3 dengan rentetan *bit binary* dari data yang disembunyikan. Secara matematis LSB melakukan pengubahan nilai *bit* paling rendah dari sampel *MP3* dengan nilai *bit* pesan yang akan disisipkan.

Algoritma kriptografi *Data Encryption Standard* (DES) diadopsi oleh NIST (*National Institue of Standard and Technology*) sebagai standar pengolahan informasi Federal AS. Secara umum *DES* terbagi menjadi tiga kelompok, yaitu pemrosesan kunci, enkripsi data 64 *bit*, dan dekripsi data 64 *bit* yang mana satu kelompok saling berinteraksi satu dengan yang lainnya.

Aplikasi dalam penelitian ini dibangun menggunakan metode steganografi *Least* Significant *Bit* (LSB), algoritma kriptografi *Data Encryption Standard* (DES) dan bahasa pemrograman java.

Kata kunci— Steganografi, LSB, MP3, Kriptografi, *DES,* Java

Abstract

Information can be something very precious and needs to be kept confidential. One of the solutions in securing information is to cryptography and steganography. Steganography is the art and science which studies data hiding on a medium such that its presence is not detected by the other party is not entitled to that information, whereas cryptography is a technique to encode (encryption) a secret data into encrypted data is incomprehensible.

Steganography method Least Significant Bit (LSB) or low bit coding is a simple method of hiding data in the process, namely by replacing bit less important / least significant bit of each sampling point MP3 filewith a series of binary bits of hidden data. Mathematically LSB perform the conversion value of the lowest bit MP3 samples with values message bits to be inserted.

Cryptography algorithm Data Encryption Standard (DES) was adopted by NIST (National Institute of Standards and Technology) a US Federal information processing standards. Generally DES divided into three groups, namely key processing, 64-bit data encryption and decryption of data 64 bits in which a group of interacting with each other.

The system in this research was built using steganography method Least Significant Bit (LSB), a cryptography algorithm Data Encryption Standard (DES) and the Java programming language.

Keywords*— Steganography, LSB,MP3, Cryptography, DES, Java*

1. PENDAHULUAN

Dengan semakin berkembangnya pemanfaatan teknologi informasi dalam membantu pekerjaan manusia di berbagai jenis kegiatan yang melibatkan komputer sebagai medianya, maka keamanan menjadi aspek yang sangat penting dalam sistem informasi. Beberapa informasi umumnya hanya ditujukan bagi segolongan orang tertentu, oleh karena itu keamanan data sangat dibutuhkan untuk mencegah informasi tersebut sampai pada pihak – pihak lain yang tidak berkepentingan sehingga adanya kemungkinan kebocoran atau penyalahgunaan data dapat dihindari, maka dirancang suatu sistem keamanan yang berfungsi untuk melindungi informasi tersebut.

Informasi dapat menjadi sesuatu yang sangat berharga dan perlu dijaga kerahasiaannya. Salah satu solusi dalam mengamankan informasi adalah dengan kriptografi dan steganografi. Steganografi merupakan ilmu dan seni yang mempelajari penyembunyian data rahasia pada suatu media sedemikian rupa sehingga keberadaannya tidak terdeteksi oleh pihak lain yang tidak berhak atas informasi tersebut, sedangkan kriptografi adalah teknik menyandikan (enkripsi) sebuah data rahasia menjadi data tersandi yang tidak dimengerti.

Media pembawa pesan pada steganografi dapat menggunakan berkas sistem komputer, transmisi protokol, dokumen teks dan representasi digital dari produk multimedia seperti berkas *audio*, citra, dan video. Keberadaan berkas audio khususnya *MP3* di internet telah menyebar sangat luas dan hampir semua pengguna internet mengenalnya. Karena keberadaannya yang sangat umum, berkas ini sangat sesuai digunakan sebagai media steganografi. Terlebih lagi kebanyakan pengguna hanya mengenal *MP3* sebagai sebuah sarana hiburan semata, sehingga dapat meminimalkan kecurigaan bahwa terdapat informasi rahasia di dalamnya.

Dalam penelitian ini akan dilakukan penyembunyian data ke dalam *file audio* berformat *MP3* (*.mp3) dengan metode penggantian bit (*Least Significant Bit*), yaitu mengganti bagian tertentu dari *bit-bit file MP3* dengan data atau informasi yang disisipkan. Namun pada perkembangannya kini disadari bahwa teknik substitusi (*Least Significant Bi*t) dalam steganografi rentan terhadap analisis statistik dalam proses steganalisis [1].

Refensi lainnya pada masalah yang diangkat diangkat oleh [2]yang berjudul "Peningkatan Keamanan Data Menggunakan Algoritma *Rijndael* Pada Audio Steganografi Berbasis *MP3*". Penelitian tersebut membahas tentang bagaimana menerapkan algoritma Rijndael dalam mengenkripsi pesan rahasia sebelum disisipkan ke dalam *MP3*. Bagaimana menyisipkan pesan rahasia terenkripsi ke dalam *MP3*. Bagaimana pengaruh penyisipan pesan rahasia terhadap keamanan data.

Referensi lain pada masalah yang diangkat oleh [3] yang berjudul "Simulasi Kerahasiaan/Keamanan Informasi Dengan Menggunakan Algoritma DES (*Data Encryption Standard*)". Penelitian tersebut membahas tentang bagaimana merancang suatu perangkat lunak yang dapat melakukan simulasi metode *DES*.

2. METODE PENELITIAN

2.1 Audio

Suara atau bunyi adalah suatu gelombang longitudinal yang merambat melalui suatu medium, seperti zat cair, padat dan gas. Bunyi dapat terdengar oleh manusia apabila gelombang tersebut mencapai telinga manusia dengan frekuensi 20Hz - 20kHz , suara ini disebut dengan *audiosonic* atau dikenal dengan *audio*, gelombang suara pada batas frekuensi tersebut disebut dengan sinyal akustik. Akustik merupakan cabang fisika yang mempelajari bunyi. Level tekanan suara (volume suara) dihitung dalam *desibel* (dB), yaitu perhitungan rasio antara titik referensi yang dipilih dalam skala logaritmik dan level yang benar-benar dialami. Keras lemahnya bunyi atau tinggi rendahnya gelombang disebut dengan amplitudo. Bunyi mulai dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga.

2.2 MP3

MPEG-1 Layer 3 atau dikenal MP3 yaitu berkas *audio* yang akan dibahas pada skripsi ini dikembangkan oleh seorang insinyur Jerman, Karlheinz Brandenburg. *MP3* memakai pengkodean *Pulse Code Modulation* (PCM). *MP3* adalah salah satu format

pengkodean berkas suara yang memiliki kompresi yang baik (meskipun bersifat *lossy*) sehingga ukuran berkas bisa memungkinkan menjadi lebih kecil, *MP3* mengurangi jumlah *bit* yang diperlukan dengan menggunakan model *psychoacoustic* untuk menghilangkan komponen-komponen suara yang tidak terdengar oleh manusia.

2.3 Steganografi

Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos*, yang artinya tersembunyi atau terselubung, dan *graphia* yang artinya menulis, sehingga arti steganografi adalah "menulis (tulisan) terselubung" [3]. Dengan steganografi, kita dapat menyisipkan pesan rahasia kedalam media lain dan mengirimkannya tanpa ada yang menyadari keberadaan pesan tersebut [1].

2.4 Steganografi Audio

Steganografi *Audio* adalah teknik penyisipan pesan rahasia dalam media suara (*audio*). Proses penyisipan pesan rahasia dalam sistem steganografi pada dasarnya dilakukan dengan mengidentifikasi media *audio* pembawa pesan, yaitu *redundant bit* yang mana dapat dimodifikasi tanpa merusak integritas dari media *audio* itu sendiri. Dalam mengaplikasikan steganografi pada berkas *audio* dapat dilakukan dengan berbagai teknik. Berikut adalah beberapa teknik yang dapat digunakan:

1. Penggantian *bit*. Cara ini lazim digunakan dalam teknik digital steganografi yaitu mengganti bagian tertentu dari *bit-bit* datanya dengan data rahasia yang disisipkan. Dengan metode ini keuntungan yang didapatkan adalah ukuran pesan yang disisipkan relatif besar, namun berdampak pada hasil *audio* yang berkualitas kurang dengan banyaknya derau.
2. Metode kedua yang digunakan adalah merekayasa fasa dari sinyal masukan. Teori yang digunakan adalah dengan mensubstitusi awal fasa dari tiap awal segmen dengan fasa yang telah dibuat sedemikian rupa dan merepresentasikan pesan yang disembunyikan. Fasa dari tiap awal segmen ini dibuat sedemikian rupa sehingga setiap segmen masih memiliki hubungan yang berujung pada kualitas suara yang tetap terjaga. Teknik ini menghasilkan keluaran yang jauh lebih baik daripada metode pertama namun dikompensasikan dengan kerumitan dalam realisasinya.
3. Metode yang ketiga adalah penyebaran spektrum. Dengan metode ini pesan dikodekan dan disebar ke setiap spektrum frekuensi yang memungkinkan. Maka dari itu akan sangat sulit bagi yang akan mencoba memecahkannya kecuali ia memiliki akses terhadap data tersebut atau dapat merekonstruksi sinyal acak yang digunakan untuk menyebarkan pesan pada *range* frekuensi.
4. Metode terakhir yang sering digunakan adalah menyembunyikan pesan melalui teknik *echo*. Teknik menyamarkan pesan ke dalam sinyal yang membentuk *echo*. Kemudian pesan disembunyikan dengan memvariasikan tiga parameter dalam *echo* yaitu besar amplitude awal, tingkat penurunan atenuasi dan *offset*. Dengan adanya *offset* dari *echo* dan sinyal asli maka *echo* akan tercampur dengan sinyal aslinya, karena sistem pendengaran manusia yang tidak memisahkan antaha *echo* dan sinyal asli.

2.5 Least Significant Bit

Least Significant Bit atau *low bit coding* merupakan metode yang sederhana dalam proses menyembunyikan data, yaitu dengan cara mengganti *bit* yang kurang penting/*least significant bit* dari setiap *sampling point* dengan rentetan *bit binary* dari data yang disembunyikan. Secara matematis ***LSB*** melakukan pengubahan nilai *bit* paling rendah dari sampel *audio* dengan nilai *bit* pesan yang akan disisipkan. Pada Tugas Akhir ini *file* penampung atau *cover* dari metode *LSB* adalah *audio* digital yang berupa *file MP3* dan *bit–bit* pesan atau data rahasia yang akan disisipkan ialah *file* yang telah terenkripsi dengan algoritma *DES*. Keluarannya adalah *file MP3* yang telah disisipi dengan *bit–bit* pesan rahasia.

Contoh:

Misalkan data yang ingin sisipkan berupa teks "*sec*". Kalau direpresentasikan ke dalam binary, maka kata "sec" diubah menjadi *binary* seperti ditunjukkan pada Tabel 1.

Tabel 1 Contoh Perubahan Karakter Menjadi Binary

Karakter	Binary
s	01110011
e	01100101
c	01100011

Misalkan media suara yang akan anda sisipi mempunyai panjang 24 *byte*, dengan nilai yang ditujukkan pada Tabel 2.

Tabel 2 Media Penampung

Byte			
00000000	00000000	00000001	00000001
00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001
00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001
00000001	00000001	00000001	00000001

Binari dari kata "*sec*" kemudian di urutkan "01110011 01100101 01100011". Setiap 1 *bit* dari kata "*secret*" akan menggantikan *bit* terakhir dari setiap *byte* media penampung, maka stego yang dihasilkan ditunjukkan pada Tabel 3.

Tabel 3 Urutan *Byte* Media Penampung Setelah Dilakukan Penyisipan

Byte			
00000000	00000001	00000001	00000001
00000000	00000000	00000001	00000001
00000000	00000001	00000001	00000000
00000000	00000001	00000000	00000001
00000000	00000001	00000001	00000000
00000000	00000000	00000001	00000001

2.6 Kriptografi

Kriptografi (*cryptographi*) berasal dari Bahasa Yunani: "*cryptos*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan). Sehingga kriptografi berarti "*secret writing*" (tulisan rahasia). Jadi kriptografi didefinisikan sebagai ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya dalam bentuk yang tidak dapat dimengerti lagi maknanya [5].

Kata "seni" dalam definisi tersebut berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kroptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia, pesan tersebut mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan.

2.7 Data Encryption Standard (DES)

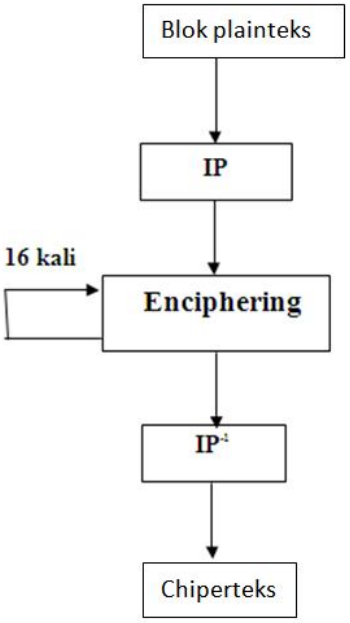
Data Encryption Standard (DES) merupakan algoritma enkripsi yang paling banyak dipakai di dunia. *DES* diadopsi oleh NIST (*National Institue of Standard and Technology*) sebagai standar pengolahan informasi Federal AS. Secara umum DES terbagi menjadi tiga kelompok, yaitu pemrosesan kunci, enkripsi data 64 *bit*, dan dekripsi data 64 *bit* yang mana satu kelompok saling berinteraksi satu dengan yang lainnya.

DES termasuk ke dalam sistem kriptografi kunci simetri dan tergolong ke dalam cipher blok. DES beroperasi pada ukuran blok 64 *bit*. DES mengenkripsi 64 *bit* plainteks menjadi 64 *bit* cipherteks dengan menggunakan 56 *bit* kunci internal (*internal key*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 *bit*. Untuk skema globanya dapat dilihat pada Gambar 1.

Keterangan skema global kriptografi *DES* adalah sebagai berikut:

1. Blok plainteks dipermutasikan dengan matriks permutasi awal (*Initial Permutation*, IP).
2. Hasil permutasi awal kemudian di*enchipering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasikan dengan matriks permutasi

balikan (*invers initial permutation*, IP^{-1}) menjadi blok cipherteks.



Gambar 1 Skema global kriptografi DES

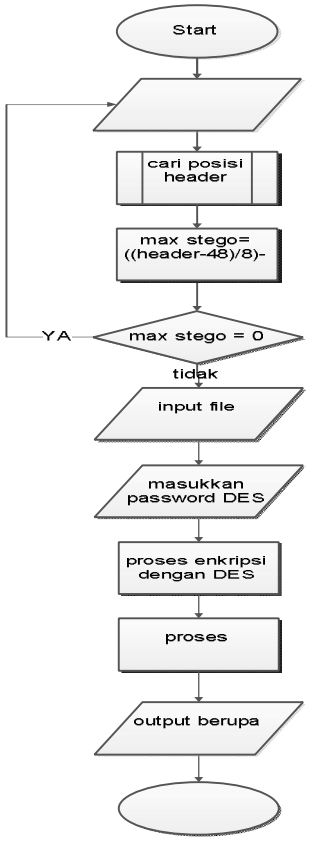
2.8 Perancangan *Flowchart*

1) *Flowchart penyisipan*

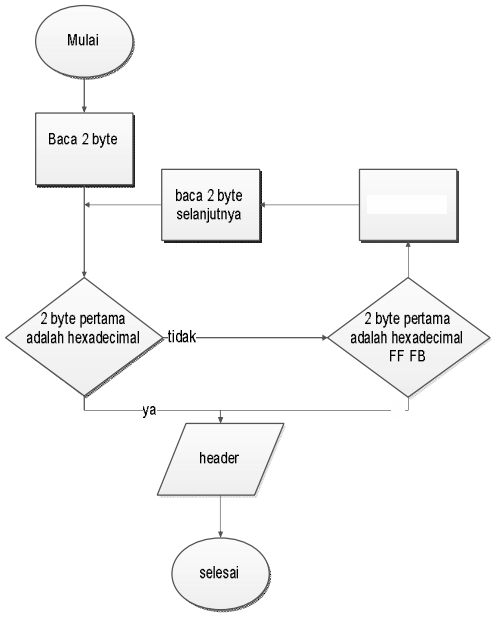
Proses penyisipan seperti yang di tunjukan pada Gambar 2. Proses penyisipan di mulai dengan

a. Meng-*input* berkas *MP3* yang akan digunakan sebagai *MP3 cover*
b. Kemudian sistem akan mencari posisi *Header* pertama. Posisi *frame* pertama sangat penting sebab untuk diketahui karena di gunakan untuk menghitung jumlah maksimal penampungan suatu *MP3*. Seperti yang ditunjukan oleh Gambar 3.

Pertama-tama sistem akan membaca 2 *byte* pertama dalam bentuk *hexadecimal* apakah 2 *byte* pertama tersebut merupakan karakter FF FA atau FF FB yang merupakan ciri dari *Header* pertama sebuah berkas *MP3*, jika tidak ditemukan maka sistem akan mencari bergeser 1 *byte* dan mencari di 2 *byte* selanjutnya, proses tersebut diulangi sampai mendapat karakter FF FA atau FF FB jika sampai *byte* akhir tidak di temukan maka berkas *MP3* tersebut tidak dapat digunakan



Gambar 2 *Flowchart* Proses Penyisipan



Gambar 3 Flowchart Proses Pencarian Header Pertama

1. *Max stegged* ialah hasil perhitungan dari posisi *Header* pertama di bagi 8 sebab steganografi ini menggunakan *LSB* di kurangi 96 yang merupakan informasi dari berkas rahasia yang telah dienkripsi, menghitung nilai *Max stegged* dapat dituliskan jabarkan dalam Persamaan (1).

$$M = \frac{H - 48}{8} - 96 \qquad (1)$$

M = maksimal ukuran pesan rahasia yang dapat disisipkan
H = letak header pertama
8 = 1 *byte* sama dengan 8 *bit* atau karakter *LSB* mengganti 1 *bit* terakhir
48 = 48 *byte* pertama dari sebuah *MP3* yang mengandung informasi *MP3*
96 = Informasi *file* yang di enkripsi dengan *DES* yang terdiri dari 96 *byte.*

2. Menginputkan berkas rahasia berekstensi tiga huruf misal *.txt, *.pdf, *.doc, *.gif, *.rar.
3. Menginputkan password atau kata kunci berupa 8 karakter
4. Sistem melakukan proses enkripsi dan penyisipan
5. Sistem membuat berkas *MP3* baru dengan nama steg.nama berkas *MP3*.mp3

2) *Flowchart* Ektraksi
Proses ekstraksi di mulai dengan :

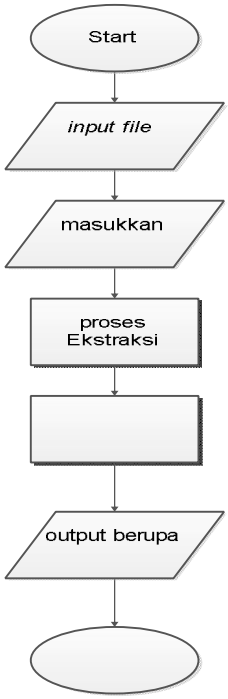
1. Menginput berkas *stegged MP3*
2. Menginput password kata kunci sebanyak 8 karakter
3. Sistem mengetraksi pesan rahasia kemudian didekrispsi menggunakan *DES*
4. Sistem membuat berkas baru hasil ektraksi dengan nama des.steg.nama berkas *MP3* adapun gambar *flowchart* proses ekstraksi ditunjukkant pada Gambar 4.

2.9 Perancangan *UML*

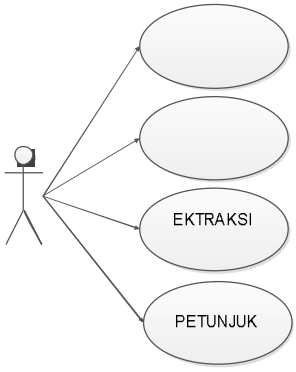
Sistem dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *usecase diagram, activity diagram*, dan *sequence diagram.*

a) Use Case Diagram

Use case diagram adalah diagram yang membantu *end user* untuk mengetahui apa yang akan dilakukan pada sistem. Berikut adalah rancangan *use case diagram* dalam penelitian ini. *Use case diagram* sistem dapat dilihat pada Gambar 5.



Gambar 4 *Flowchart* Proses Ekstraksi



Gambar 5 *Use case diagram* sistem

b) Activity Diagram

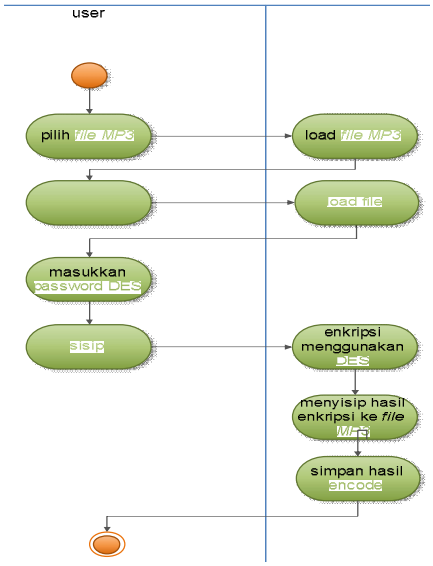
Activity diagram memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. *Activity diagram* penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar

objek. Berikut adalah rancangan *activity diagram* dalam penelitian ini.

1) Activity diagram Penyisipan

Proses penyisipan adalah proses dimana user melakukan penyisipan pesan rahasia ke dalam *file MP3. Activity Diagram* penyisipan ditunjukkan pada Gambar 6, Adapun tahapannya sebagai berikut :

1. Memasukkan *file MP3*
2. Memasukkan *file* rahasia
3. Memasukkan password
4. Melakukan penyisipan



Gambar 6 *Activity diagram Penyisipan*

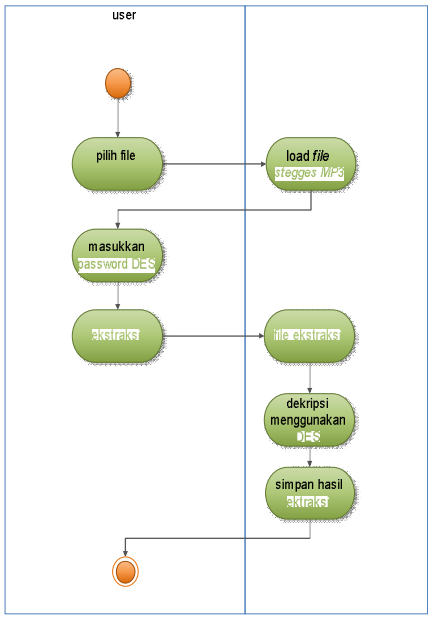
2) Activity diagram Ekstraksi

Proses penyisipan adalah proses dimana user melakukan penyisipan pesan rahasia ke dalam *file MP3, Activity Diagram* penyisipan ditunjukkan pada Gambar 7. Adapun tahapannya sebagai berikut :

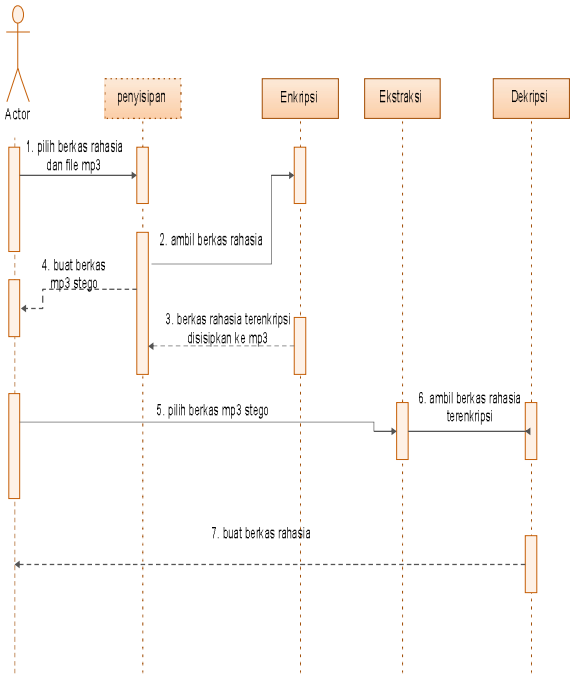
1. Memasukkan *file stegged MP3*
2. Memasukkan password
3. Melakukan ektraksi.

c) Sequence Diagram

Sequence diagram memperlihatkan urutan aktivitas ke aktivitas lainnya dalam suatu sistem. *Sequence Diagram* sistem ditunjukkan pada Gambar 8.



Gambar 7 *Activity diagram* Ekstraksi



Gambar 8 *Sequence diagram*

3. HASIL DAN PEMBAHASAN

Implementasi merupakan tahapan pengoperasian sistem yang telah siap digunakan.Hasil analisis dan perancangan diimplementasikan dalam bentuk aplikasi steganografi *MP3*.

Spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam implementasi ditunjukkan pada Tabel 4 dan 5.

Hardware	*Specification*
Memory	4GB
Harddisk	250 GB
Processor	Intel® Core 2 Duo™

Tabel 5 Spesifikasi perangkat lunak

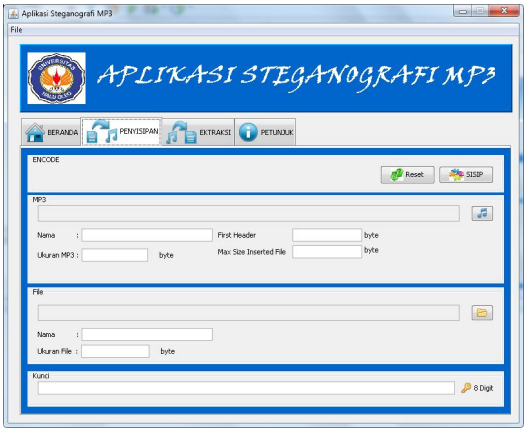
No.	Nama Perangkat Lunak	Spesifikasi
1.	*Operating System*	*Windows 7 Ultimate*
2.	Bahasa Pemrograman	Java
3.	Tools Pemodelan	E-draw

Tampilan utama atau beranda *interface* sistem ditunjukkan pada Gambar 9.

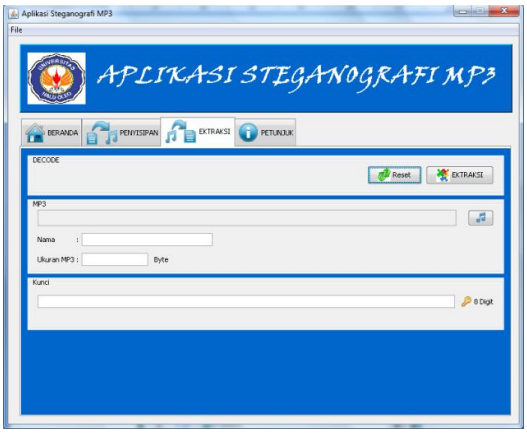
Gambar 9 Tampilan utama/ *interface* sistem

Pada menu penyisipan *user* melakukan proses penyisipan pesan rahasia ke dalam pada *file MP3*. Tampilan dari menu penyisipan ditunjukkan pada Gambar 10.

Pada menu ekstraksi *user* melakukan proses mengambil kembali file rahasia atau ekstraksi dari dalam *file MP3*. Tampilan dari menu penyisipan seperti pada Gambar 11.

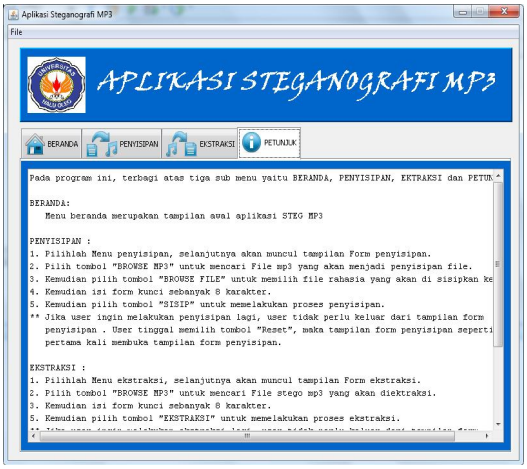


Gambar 10 Tampilan *Menu* Penyisipan



Gambar 11 Tampilan *Menu* ekstraksi

Pada menu Petunjuk *user* dapat melihat cara menggunakan aplikasi steganografi MP3. Tampilan dari menu petunjuk ditunjukkan pada Gambar 12.



Gambar 12 Tampilan *Menu* Petunjuk

3.1 Pengujian Penyisipan

Untuk mengetahui aplikasi dapat melakukan proses penyispan sesuai dengan fungsinya, maka dilakukan percobaan dengan menggunakan 10 *file MP3* sebagai *cover* dan *file* ujicoba.rar dengan ukuran 100 *byte* sebagai *file* rahasia, hasil percobaan penyispan dapat dilihat pada Tabel 6.

Tabel 6 Hasil Pengujian Penyisipan

No	Nama *MP3*	Ukuran *MP3* (*byte*)	*Max* stego (*byte*)	Nama *Stegged MP3*	Ukuran *Stegged MP3* (*byte*)	keterangan
1	After-Dark	2.408.425	10349	Steg.After-Dark	2.408.425	Sukses
2	Alones	3.221.547	10346	Steg.Alones	3.221.547	Sukses
3	Anima-Rossa	3.371.082	10348	Steg.Anima-Rossa	3.371.082	Sukses
4	Answer is Near	8.855.723	7353	Steg.Answer is Near	8.855.723	Sukses
5	Feed_A	10.793.029	21206	Steg.Feed_A	10.793.029	Sukses
6	Kaimu	8.822.424	13581	Steg.Kaimu	8.822.424	Sukses
7	Karasu	8.546.906	13581	Steg.Karasu	8.546.906	Sukses
8	Strawberry Fuzz	11.874.988	45754	Steg.Strawberry Fuzz	11.874.988	Sukses
9	ONION!	8.001.899	9425	Steg.ONION!	8.001.899	Sukses
10	RAINBOWS	11.783.116	45754	Steg.RAINBOWS	11.783.116	Sukses

Dari percobaan-percoobaan yang telah dilakukan dapat disimpulkan bahwa proses penyisipan tidak mempengaruhi ukuran *file MP3*, sebab ukuran *file MP3* baik sebelum dan sesudah disisipkan *file* rahasia masih tetap sama.

Untuk mengetahui bagaimana sistem dapat menentukan batas maksimal penyisipan sebuah *file MP3*, maka penulis akan memperlihatkan proses perhitungan *max* stego dengan percobaan pada Feed_A.mp3 yang memiliki informasi seperti yang ditampilkan pada Gambar 13.

Dari Gambar 13 Feed_A.mp3 mempunyai ukuran 10.793.029 *byte* posisi *first header* berada pada *byte* ke 170.470 dan *max* stego sebesar 21206 *byte*. Dengan menggunakan Persamaan (1) mencari *max* stego(1).

$$M = \frac{H - 48}{8} - 96$$

Dik : H = 170.470

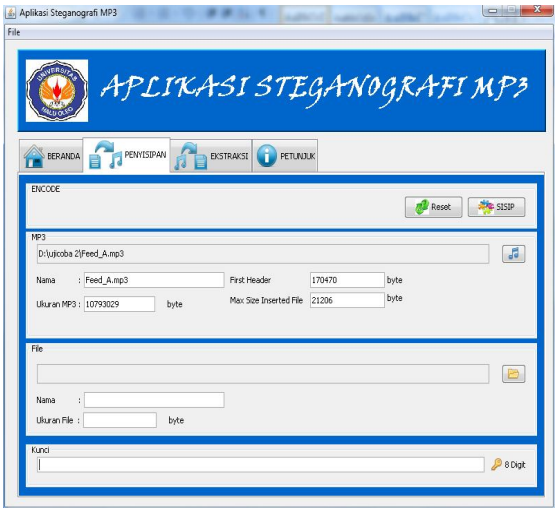
Dit : M ?

Maka : $M = \frac{170.470 - 48}{8} - 96$

$$M = \frac{170.422}{8} - 96$$

$$M = 21302{,}75 - 96$$

$$M = 21206{,}75$$



Gambar 13 Percobaan Menggunakan *File Mp3* Yang Bernama Feed_A.Mp3

Dari pengujian *max* stego menggunakan persamaan *max* stego diketahui max stego dari Feed_A.mp3 ialah 21206,75 atau dibulatkan menjadi 210206, sesuai dengan didapatkan oleh aplikasi bahwa *max* stego dari Feed_A.mp3 adalah 210206 *byte*.

3.2 Pengujian Kekuatan *Signal* Audio

Penilaian kualitas *file audio MP3* dilakukan secara subjektif dan objektif .penilaian subjektif dengan cara mendengarkan suara hasil pemutaran *file MP3* asli dan membandingkannya dengan sura hasil *file stegged MP3*, Dalam melakukan pengujian peneliti menggunakan aplikasi Daum Potplayer untuk memutar *file MP3*.

Penilaian objektif dilakukan dengan cara membandingkan nilai *signal* audio yaitu *left peak signal* dan *right peak signal* antara *file MP3* asli dengan *file stegged MP3* menggunakan aplikasi Goldwave v.5.70, hasil pengujian *signal* audio dengan satuan *decibel* (dB). Dimana P0 mewakili *file MP3* asli dan P1 mewakili *stegged MP3*. Hasil pengujian dapat dilihat pada Tabel 7.

Tabel 7 Hasil Pengujian *Left Peak Signal* Dan *Right Peak Signal*

MP3 asli	Ukuran *MP3* (*byte*)	*Max* stego (*byte*)	*File* rahasia	Ukuran *file* rahasia (*byte*)	*left peak signal* P0 (dB)	*left peak signal* P1 (dB)	*Right peak signal* P0 (dB)	*Right peak signal* P1 (dB)
Feed_A	10.793.029	21206	HP-Radio.rar	16358	-7.83	-7.83	-8.02	-8.02
Kaimu	8.822.424	13581	yoichi.gif	10138	-8.59	-8.59	-8.83	-8.83
Karasu	8.546.906	13581	ujiteks.txt	1413	-8.91	-8.91	-8.97	-8.97
ONION!	8.001.899	9425	qw.bmp	3054	-9.22	-9.22	-9.13	-9.13
RAINBOWS	11.783.116	45754	Audio.pdf	42646	-9.84	-9.84	-10.17	-10.17
Strawberry Fuzz	11.874.988	45754	hiruma2.jpg	18437	-9.82	-9.82	-9.80	-9.80

3.3 Aspek *Recovery*

Aspek *recovery* yang dimaksud adalah *file* rahasia yang disisipkan harus dapat diekstraksi kembali. Maka pada tahap ini akan dilakukan pengujian untuk memastikan bahwa *file* rahasia yang disipkan ke *file MP3* dapat diekstraksi dalam keadaan baik. Indikator keberhasilan pengujian ini adalah jika ukuran *file* rahasia sama dengan *file* hasil ekstraksi.maka dilakukan percobaan dengan menggunakan 10 *file MP3 Stegged* yang telah di sisipkan *file* ujicoba.rar dengan ukuran 100 *byte* sebagai *file* rahasia, hasil percobaan penyispan dapat dilihat pada Tabel 8.

Tabel 8 Hasil Pengujian *Recovery*

No	Nama *Stegged MP3*	Ukuran *Stegged MP3* (*byte*)	Nama *File* rahasia hasil ekstraksi	Status ekstraksi	Ukuran *file* rahasia hasil ekstraksi (*byte*)
1	Steg.After-Dark	2.408.425	Dsa.Steg.After-Dark.rar	Sukses	100
2	Steg.Alones	3.221.547	Dsa.Steg.Alones.rar	Sukses	100
3	Steg.Anima-Rossa	3.371.082	Dsa.Steg.Anima-Rossa.rar	Sukses	100
4	Steg.Answer is Near	8.855.723	Dsa.Steg.Answer is Near.rar	Sukses	100
5	Steg.Feed_A	10.793.029	Dsa.Steg.Feed_A.rar	Sukses	100
6	Steg.Kaimu	8.822.424	Dsa.Steg.Kaimu.rar	Sukses	100
7	Steg.Karasu	8.546.906	Dsa.Steg.Karasu.rar	Sukses	100

8	Steg.Strawberry Fuzz	11.874.988	Steg.Strawberry Fuzz.rar	Sukses	100
9	Steg.ONION!	8.001.899	Dsa.Steg.ONION! .rar	Sukses	100
10	Steg.RAINBOWS	11.783.116	Dsa.Steg.RAINBOWS.rar	Sukses	100

Setelah dilakukan 10 pengujian tersebuat maka akan dihitung persentase keberhasilan proses *recovery* dengan cara membandingkan jumlah pengujian dengan status sukses dengan jumlah percobaan yang dilakukan.

$$Pk = \frac{Jb - Jg}{Jp} X\ 100\%$$

$$= \frac{10 - 0}{10} x\ 100\%$$

$$= 100\ \%$$

Pk= persentase keberhasilan
Jb = jumlah percobaan berhasil
Jg = jumlah percobaan gagal
Jp = jumlah seluruh percobaan

Dari pengujian pada aspek *recovery* yang penulis lakukan tingkat keberhasilnya adalah 100%. Artinya aplikasi steganografi *MP3* ini relah memenuhi aspek *recovery* padankeamanan data.

3.4 Aspek Robustness

File rahasia rahasia yang disisipkan harus tahan tehadap pengolahan sinyal yang mungkin dilakukan untuk mendukung agar mendukung aspek *Robustness* , maka dari itu penulis melakukan pengujian dengan mengubah bitrate pada **steg.Feed_A.mp3** yang semula 320 kbps stereo menjadi 240 kbps stereo menggunakan bantuan aplikasi Goldwave v.5.70. setelah itu penulis mencoba mengekstraksi **steg.Feed_A.mp3** yang telah di ubah *bitrate* nya, Hasilnya *file* rahasia yang berada di dalamnya tidak dapat diekstraksi. Dari hasil pengujian tersebut diketahui aplikasi steganografi *MP3* tidak mendukung aspek *Robustness*. Hal ini disebabkan *bit-bit* dari *MP3* tersebut telah berubah, sehingga bit-bit *file* rahasia yang disisipkan juga ikut berubah.

4. KESIMPULAN

Berdasarkan pembangunan dan pengujian yang telah dilakukan terhadap aplikasi peningkatan keamanan data rahasia menggunakan kriptografi *data encryption standard* pada steganografi *audio* berformat *MP3*, maka dapat diambil kesimpulan sebagai berikut :

1. Steganografi *file audio* berformat *MP3* dengan algoritma *Least Significant Bit* (LSB) dan enkripsi dengan *Data Encryption Standard* (DES) dapat diterapkan.
2. Kapasitas yang dapat ditampung oleh sebuah *file MP3* bergantung pada posisi *header* bukan bergantung pada ukuran *MP3*.
3. Perubahan *bit-bit* pada *file MP3* hasil steganografi tidak mempengaruhi ukuran *MP*3 sehingga mendukung aspek *fidelity* pada keamanan data. dan suara yang dihasilkan masih baik serta perubahan yang terjadi tidak dapat dideteksi oleh indra pendengaran manusia.
4. Ekstraksi *file* rahasia hanya dapat dilakukan apabila pengguna memasukkan kunci dekripsi yang sesuai, jika tidak, maka *file* rahasia masih tetap dalam keadaan terenkripsi.
5. Aplikasi dapat melakukan penyisipan dan ekstraksi dengan baik sehingga mendukung aspek *recovery* pada keamanan data
6. Aplikasi ini tidak mendukung aspek *robustness* sebab ketika *file stegged MP3* dilakukan perubahan *bitrate* dari 320 *stereo* ke 240 *stereo file* rahasia tidak dapat di ekstraksi sebab *bit-bit* pada *file stegged MP3* telah mengalami perubahan.

5. SARAN

Berdasarkan hasil penelitian yang telah dilakukan, ada beberapa saran untuk pengembangan lebih lanjut terhadap sistem, diantaranya sebegai berikut :

1. Untuk mengatasi kunci yang terlalu pendek pada algoritma *Data Encryption Standard* (DES) maka diharapkan adanya peningkatan algoritma *Data Encryption Standard* (DES) menjadi 3DES atau *Triple*

Data Encryption Standard atau dengan algoritma enkripsi lainnya.
2. Untuk memenuhi aspek *Robustness*, maka diharapkan perlu adanya metode-metode steganografi khususnya pada *file audio MP3*.
3. Perlu adanya pengembangan terhadap aplikasi agar dapat meneyambunyikan *file* ekstensi lebih dari 3 karakter.
4. Perlu adanya pengembangan terhadap aplikasi agar tidak dapat menyembunyikan *file* yang melebihi batas penyembunyian, sebab akan merusak kualitas *file MP3*.
5. Perlu adanya pengembangan terhadap maksimal kapasitas pesan rahasia yang dapat ditampung tanpa mengurangi kualitas *MP3*.

DAFTAR PUSTAKA

[1] Krenn, J.R., 2004, Steganography and Steganalysis, *http://www.krenn.nl/univ/cry/steg/article.pdf*, diakases 12 Mei 2016.

[2] Nurhasanah, R., 2010, Peningkatan Keamanan Data Menggunakan Algoritma *Rijndael* Pada Audio Steganografi Berbasis *MP3*, *Skripsi*, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan.

[3] Alvi, M.R., 2011, Perancangan Aplikasi Kriptografi Menggunakan Algoritma *TDES*, *Skripsi*, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara,Medan.

[4] Cvejic, N., 2004, Algorithms for Audio Watermarking and Steganography.Oulu : Oulu University Press, *http://www.herkules.oulu.fi/isbn9514273842/isbn9514273842.pdf*, diakses 15 Mei 2016.

[5] Kurniawan, Y., 2004, Kriptografi, Keamanan Internet dan Jaringan Komunikasi. Bandung : Informatika.