IMPLEMENTASI DAN ANALISA TEKNIK STEGANOGRAFI MULTI-CARRIER PADA FILE MULTIMEDIA

Canggih Satriatama[1], Izbat Uzzin Nadhori,S.Kom,MT[2], Drs. Miftahul Huda, MT[3]
[1]Mahasiswa Jurusan Teknik Informatika, [2]Dosen Jurusan Teknik Informatika, 3 Dosen Jurusan Teknik Telekomunikasi
Jurusan Teknik Informatika
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
Tel: (+62-31) 594 7280; Fax:(+62-31) 594 6114
E-mail: pens@eepis-its.edu or satriatama@gmail.com
Homepage: http://www.eepis-its.edu

ABSTRAK

Perkembangan media digital yang pesat dan penggunaannya yang meliputi berbagai bidang menimbulkan tuntutan yang semakin besar untuk menciptakan suatu sistem penyampaian informasi yang terjamin keamanannya. Salah satunya adalah dengan steganografi. Steganografi merupakan suatu metode untuk menyisipkan potongan informasi rahasia dalam suatu objek atau media lain. Dengan steganografi, informasi disembunyikan sedemikian rupa sehingga tidak diketahui keberadaannya, yang dikenal dengan istilah informasi hiding. Metode ini berbeda dengan metode kriptografi, yang menyandikan informasi yang ada sehingga tidak dapat dibaca tanpa mengetahui kunci atau sandi yang digunakan, namun keberadaannya tetap diketahui dan tidak disembunyikan.

Proyek akhir ini dikembangkan dengan menggunakan Microsoft Visual C# mengimplementasikan metode steganografi Simple Least Significant Bit Substitution ("simple LSB subtitution") untuk menyembunyikan suatu informasi ke dalam file multimedia. File multimedia yang digunakan adalah file citra, file audio, serta file video sebagai media pembawa informasi rahasia.

Penggunaan teknologi steganografi ini diharapkan dapat meningkatkan keamanan dalam penyampaian informasi, agar informasi-informasi penting akan terlindungi dan tersamarkan keberadaannya dalam file multimedia. Hal ini juga diharapkan dapat membantu proses perlindungan atas hak cipta hasil karya media elektronik.

Kata kunci: *steganografi, multimedia, penyisipan informasi, simple LSB substitution,*

ABSTRACT

The rapid development of digital media and its use in various fields lead to greater demands for creating secured information delivery systems. Among of those is steganography. Steganography is a method to insert a piece of confidential information in an object or other media. By using steganography, the information hidden in a way that meant to make it is not known to exist, which is known as informasi *hiding. This method differs from cryptography method which encodes the information so it cannot be read without knowing the key or password used, but its existence remains known and not completely hidden.*

This final project is developed using Microsoft Visual C # implements the method steganography Simple Least Significant Bit Substitution to hide an information into a multimedia file. Multimedia file that is used is a file image, audio files, as well as a video file as a medium of confidential information carrier.

The use of steganography technology is expected to improve safety in the delivery of information, so that key information will be protected and obscured its existence in the multimedia file. This is also expected to assist in the protection of copyright works of the electronic medium.

keywords: *steganography, multimedia, information embedding, simple LSB substitution.*

1. PENDAHULUAN

1.1. Latar Belakang

Keamanan suatu informasi pada era digital ini makin vital peranannya dalam berbagai aspek kehidupan, terutama untuk suatu informasi yang memiliki nilai lebih dibandingkan dengan informasi yang lain. Misalnya informasi yang berkaitan dengan aspek-aspek keputusan bisnis, keamanan negara, ataupun kepentingan umum. Tentunya informasi-informasi tersebut diminati oleh berbagai pihak.

Oleh karena itu pengamanan informasi, dalam hal ini adalah steganografi, semakin dibutuhkan guna memberikan rasa aman dalam proses penyampaian informasi. Steganografi sendiri merupakan cara untuk menyembunyikan suatu informasi rahasia di dalam suatu informasi atau informasi lain yang tampak tidak bermakna, kecuali bagi orang yang mengerti kuncinya. Teknik steganografi menggunakan dua media yang berbeda secara bersamaan, dimana salah satunya berfungsi sebagai media yang berisikan informasi informasi rahasia (dapat juga disebut *secret file*) dan yang lain berfungsi sebagai media pembawa informasi tersebut (*carrier file*). Namun satu hal yang perlu dicatat, meskipun menggunakan istilah *carrier* namun pada proyek akhir ini tidak akan ada proses transmisi dan penerimaan informasi, istilah tersebut diberikan kepada *file* yang disisipi *secret file.*

Pada proyek akhir ini akan dibangun suatu aplikasi berbasis Microsoft Visual C# yang mengimplementasikan steganografi dengan menggunakan metode s*imple least lignificant bit substitution* sebagai cara untuk menyembunyikan suatu informasi ke dalam *file* multimedia. Penggunaan teknologi steganografi ini diharapkan bukan hanya dapat membantu upaya meningkatkan keamanan penyampaian informasi, namun juga dapat membantu dalam proses perlindungan atas hak cipta hasil karya media elektronik.

1.2. Perumusan Masalah

Dalam pelaksanaan tugas akhir ini terdapat beberapa permasalahan yang menjadi titik utama pembahasan, adapun permasalahan-permasalahan tersebut adalah sebagai berikut :

a. Bagaimana menyisipkan suatu informasi ke dalam *file multimedia*.
b. Bagaimana melindungi suatu informasi menggunakan *key file*.
c. Bagaimana membagi suatu infromasi ke dalam beberapa *carrier file*.
d. Bagaimana mengambil kembali suatu informasi dari berkas *stego*.

1.3. Batasan Masalah

Batasan masalah pada proyek akhir agar tidak terjadi kesalahan persepsi dan tidak meluasnya pokok bahasan adalah sebagai berikut:

a. Format *file* citra digital yang akan digunakan adalah .bmp, .png.
b. Format *file* audio digital yang akan digunakan adalah .wav.
c. Format *file* video digital yang akan digunakan adalah .avi standar.
d. Format *key file* digital yang akan digunakan adalah .txt.
e. Ukuran *secret file* dan *carrier file* disesuaikan dengan metode dan skenario yang tercantum pada bagian pengujian.

1.4. Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah :

a. Memberikan informasi dan mengimplementasikan teknik steganografi ke dalam *file* multimedia.
b. Mengembangkan aplikasi berbasis Microsoft Visual *C#* yang mampu menyembunyikan, melindungi, dan membagi informasi ke dalam *file* multimedia menggunakan teknik steganografi.

2. TEORI PENUNJANG

2.1. Steganografi[1]

Kata steganografi berasal dari bahasa Yunani steganos (στεγανός) yang berarti "ditutupi atau dilindungi", dan graphein (γράφειν) yang berarti "menulis". Dari asal katanya steganografi berarti "tulisan tersembunyi". Steganografi adalah seni dan ilmu menulis informasi tersembunyi sedemikian rupa sehingga tak seorang pun, selain pengirim dan penerima yang dituju, mengetahui keberadaan informasi tersebut. Secara umum, informasi akan muncul dalam bentuk yang lain: foto, artikel, daftar belanja, atau beberapa *covertext* lain. Secara klasik, informasi disembunyikan menggunakan tinta tak terlihat diantara garis-garis yang tampak dalam sepucuk surat pribadi.

Steganografi juga meliputi penyisipan informasi dalam *file* komputer. Dalam steganografi digital, komunikasi elektronik dapat mencakup pengkodean *steganographic* dalam lapisan transportasi, seperti *file* dokumen, *file* gambar, program atau protokol. *File* media sangat ideal untuk transmisi *steganographic* karena ukurannya yang besar. Sebagai contoh sederhana, pengirim mungkin mulai dengan *file* gambar yang tidak terlalu kompleks dan menyesuaikan warna setiap piksel 100 dengan huruf dalam alfabet. Karena perubahannya begitu halus, maka seseorang yang tidak secara khusus mencarinya tidak dapat melihat informasi yang disisipkan.

Teknik steganografi yang digunakan pada proyek akhir ini adalah variasi *least significant bit substitution*, karena meskipun tergolong sederhana, namun dengan perpaduan algoritma yang tepat dapat menjadi teknik yang dapat diandalkan.

2.2. Struktur Format File[2]

Sebuah format *file* adalah cara tertentu untuk mengkodekan informasi dalam proses penyimpanan *file* dalam sistem.

Karena *disc drive*, atau bahkan setiap mekanisme penyimpanan dalam komputer, hanya dapat menyimpan informasi dalam bentuk *bit*, maka komputer harus memiliki beberapa cara untuk mengkonversi informasi ke dalam nilai 0 dan 1 dan sebaliknya. Cara mengkonversi atau mengkodekan informasi ini berbeda-beda untuk tiap jenis *file*, dan tergambar dalam bentuk format *file*. Dalam setiap tipe format, misalnya dokumen pengolah kata, biasanya akan ada format yang berbeda. Bahkan kadang-kadang masing-masing format bersaing satu sama lain.

Pada proyek akhir ini akan dipelajari mengenai struktur sebuah format *file* sehubungan dengan steganografi, baik untuk *carrier file* ataupun untuk *secret file*. Struktur format *file* harus benar-benar diperhatikan, karena informasi yang akan disisipkan atau *secret file* akan dibagi menjadi beberapa bagian agar dapat disisipkan ke dalam lebih dari satu *carrier file*.

Contoh studi struktur format *file* adalah saat kita mempelajari struktur *file* Wave untuk aplikasi steganografi. Untuk *file* digital kita dapat mencoba melihat strukturnya dengan menggunakan HEX-editor. Untuk *file* Wave kurang lebih akan muncul tampilan seperti pada gambar 2.1.

```
52 49 46 46 24 40 01 00  57 41 56 45 66 6D 74 20  RIFF$@..WAVEfmt
10 00 00 00 01 00 02 00  11 2B 00 00 44 AC 00 00  .........+..D...
04 00 10 00 64 61 74 61  00 40 01 00 00 00 00 00  ....data.@......
```

Gambar 2. 1 Pembacaan *file* Wave dengan HEX editor

2.2.1. Least Significant Bit[3]

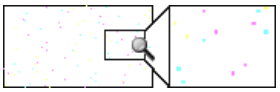
Dalam dunia komputer**,** *Least Significant Bit* (LSB) adalah posisi bit dalam sebuah *integer* biner yang memberikan nilai kepada unit, berupa nilai ganjil ataupun genap. LSB juga seringkali dianggap sebagai bit paling kanan sehubungan dengan metode penulisan bit yang kurang berguna cenderung berada di kanan yang ditunjukkan pada gambar 2.7.

1	0	0	1	0	1	0	1

Gambar 2. 2 *Right-most Digit*

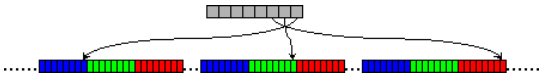
LSB memiliki sifat yang unik yaitu mampu berubah dengan cepat jika terjadi sedikit saja perubahan dari nilai awalnya. Misalnya, jika 1 (biner 00000001) ditambahkan ke 3 (biner 00000011), hasilnya akan menjadi 4 (biner 00000100). Tiga dari bit yang kurang signifikan (*Least Significant Beat*) akan berubah (011-100), dan sebaliknya, tiga bit yang signifikan tetap tidak berubah (000-000). LSB sering digunakan dalam penghitungan jumlah *pseudorandom generator*, *hash function* dan *checksum*.

Contoh implementasi studi tentang LSB untuk kasus steganografi adalah penggunaanya pada pemrosesan *file* citra digital. Hal ini dilakukan dengan mempelajari pola pelangi (*rainbow pattern*) pada citra digital tersebut saat kita mengganti bit RGB-nya seperti ditunjukkan pada gambar 2.3.



Gambar 2. 3 Pola Pelangi

Setelah didapat pola pelangi maka gambar 2.4 akan menunjukkan proses selanjutnya yaitu penghilangan pola pelangi tersebut karena kita akan menyisipkan informasi ke dalamnya.



Gambar 2. 4 Proses Pengambilan bit dan *byte*

Untuk tiap *byte* informasi, yang harus dilakukan adalah:

a. Mengambil sebuah piksel.
b. Mengambil bit pertama dari *byte* informasi.
c. Mengambil satu komponen warna dari piksel.
d. Mengambil bit pertama dari komponen warna.
e. Jika bit-warna berbeda dari bit-informasi, lakukan *set/reset*
f. Lakukan hal yang sama untuk ketujuh bit yang lain.

2.3. Digital Signal Processing (DSP)

Pemrosesan sinyal digital (*Digital Signal Processing* atau DSP) berkaitan dengan representasi sinyal dengan urutan angka atau simbol dan pemrosesan sinyal itu sendiri. Pemrosesan sinyal digital adalah sub dari pemrosesan sinyal. DSP sendiri meliputi berbagai sub-bagian, seperti: pemrosesan sinyal ucapan, pemrosesan sinyal radar, sensor pengolahan *array*, estimasi spektral, pengolahan sinyal statistik, pengolahan citra digital, pemrosesan sinyal untuk komunikasi, kontrol sistem, pemrosesan sinyal biomedis, pengolahan informasi seismik, dan masih banyak lagi.

Dalam proyek akhir ini DSP dibutuhkan untuk mempelajari tentang bagaimana mengenali dan menggunakan struktur *carrier file* agar dapat dimanfaatkan secara optimal. Ada beberapa algoritma DSP yang akan digunakan dalam proyek akhir ini:

2.3.1. Simple LSB (Least Significant Bit) Substitution[4]

Algoritma *Simple LSB Subtitution* merupakan algoritma dasar dalam pemilihan *least significant bit* dari *carrier file* untuk aplikasi steganografi.

Pada algoritma ini C sebagai 8-bit awal dari Mc x Nc piksel citra *carrier*, direpresentasikan sebagai berikut:

$$C = \{x_{ij} | 0 \le i < M_c, 0 \le j < N_c, x_{ij} \in \{0,1,\dots,255\}\} \quad (1)$$

M menjadi *n*-bit informasi rahasia direpresentasikan sebagai

$$M = \{m_i | 0 \le i < n, m_i \in \{0,1\}\} \quad (2)$$

Anggaplah *n*-bit informasi rahasia *M* disisipkan pada *k* LSB paling kanan dari citra *carrier*. Pertama, informasi rahasia *M* disusun ulang untuk membentuk *k*-bit informasi virtual *M'* konsepsual yang direpresentasikan sebagai:

$$M' = \{m_i' | 0 \le i < n', m_i' \in \{0,1,\dots,2^k - 1\}\} \quad (3)$$

Dimana $n' < M_c \; x \; N_c$. Mapping diantara *n*-bit informasi rahasia $M = \{m_i\}$ dan informasi yang telah disisipkan $M' = \{m_i'\}$ dapat didefinisikan seperti berikut:

$$m_i' = \sum_{j=0}^{k-1} m_{i\,x\,k+j} \; x \; 2^{k-1-j}$$

Kedua, sebuah subset dari piksel $n' \{x_{l1}, x_{l2}, \dots, x_{ln'}\}$ dipilih dari citra *carrier* C dari sebuah urutan yang telah ditentukan sebelumnya. Proses penyisipan disempurnakan dengan mengganti *k* LSB dari x_{li} dengan m_i'. Penjelasan secara matematis, nilai x_{li} dari piksel terpilih untuk menyimpan *k*-bit informasi m_i' diubah menjadi bentuk *stego-pixel* x_{l_i}' adalah sebagai berikut:

$$x_{l_i}' = x_{l_i} - x_{l_i} mod 2^k + m_i' \quad (4)$$

Pada proses ekstraksi, untuk *stego-image S* yang diberikan, informasi yang disisipkan telah siap untuk dapat diekstrak tanpa mengacu kepada citra *carrier* awal-nya. Dengan menggunakan urutan yang sama dengan proses penyisipan, terdapat sekumpulan piksel $\{x_{l_i}', x_{l_2}', \dots, x_{l_{n'}}'\}$ yang dipilih dari *stego-image* dan digunakan untuk menyimpan bit-bit informasi rahasia. *k* LSB dari piksel terpilih diekstrak dan disusun untuk merekonstruksi bit-bit informasi rahasia. Secara matematis, bit-bit m_i' dari informasi yang disisipkan dapat dikembalikan dengan cara:

$$m_i' = x_{l_i}' mod 2^k \quad (5)$$

Dengan anggapan bahwa seluruh piksel pada citra *carrier* digunakan untuk proses penyisipan dengan metode *simple LSB substitution.*

2.3. Microsoft Visual C#[5]

C# didisain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan *Rapid Application Development* yang 'tanpa batas' (dibandingkan dengan RAD yang 'terbatas' seperti yang terdapat pada Delphi dan Visual Basic).

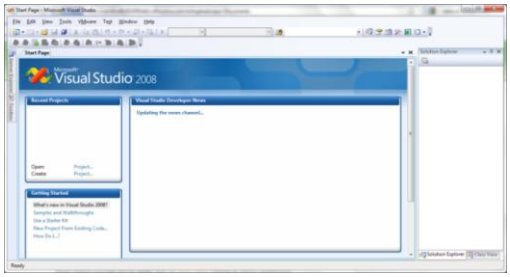
Agar mampu mempromosikan penggunaan besar-besaran dari bahasa C#, Microsoft, dengan dukungan dari Intel Corporation dan Hewlett-Packard, mencoba mengajukan standardisasi terhadap bahasa C#. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh European Computer Manufacturers Association atau *Ecma International* (ECMA), dengan nomor standar ECMA-334. Pada Desember 2002, standar kedua pun diadopsi oleh ECMA, dan tiga bulan kemudian diterima oleh *International Organization for Standardization* (ISO), dengan nomor standar ISO/IEC 23270:2006.

Kelebihan C# :

- ***Flexible*:** C# program dapat di eksekusi di mesin computer sendiri atau di transmiskan melalu web dan di eksekusi di computer lainnya
- ***Powerful*:** C# memiliki sekumpulan perintah yang sama dengan C++ yang kaya akan fitur yang lengkap tetapi dengan gaya bahasa yang lebih diperhalus sehingga memudahkan penggunanya
- ***Easier to use*:** C# memodifikasi perintah yang sepenuhnya sama dengan C++ dan memberitahu dimana letak kesalahan kita bila ada kesalahan dalam aplikasi , hal ini dapat mengurangi waktu kita dalam mencari error
- ***Visually oriented*:** library code .NET yang digunakan oleh C# menyediakan bantuan yang dibutuhkan untuk membuat tampilan yang *sophisticated* dengan *frames, dropdown , tabbed windows, group button , scroll bar , background image* , dan lainnya
- ***Secure*:** semua bahasa pemprograman yg digunakan untuk kebutuhan internet mesti memiliki security yg benar-benar aman untuk menghindari aksi kejahatan dari

pihak lain seperti, C# memiliki segudang fitur untuk menanganinya

Adapun tampilan dari Microsoft Visual Studio 2008 yang akan digunakan untuk mengembangkan aplikasi ini dapat dilihat pada gambar 2.10



Gambar 2. 2 Microsoft Visual Studio 2008

3. PERANCANGAN SISTEM DAN APLIKASI

3.1. Perancangan Umum

Dalam perkembangan dunia informasi, keamanan suatu informasi merupakan suatu hal yang sangat vital. Hal ini dikarenakan tidak semua pihak, berhak untuk mengakses informasi yang bersangkutan. Oleh karena itu diperlukan suatu aplikasi yang dapat digunakan untuk menyembunyikan atau menyamarkan suatu informasi ke dalam media lain.

Steganografi merupakan suatu teknik berkomunikasi dimana informasi disembunyikan pada media pembawa seperti citra, suara atau video tanpa memberikan perubahan yang berarti pada media tersebut. Berbeda dengan kriptografi hanya menyembunyikan arti atau isi dari sebuah informasi, steganografi mampu menyembunyikan keberadaan informasi.

Pada bagian ini akan dijelaskan mengenai gambaran umum dari proses kerja aplikasi steganografi yang akan dibuat sebelum kita memulai fase perancangan sistem.

Sebelum masuk ke dalam proses penyisipan (*hiding*), ada beberapa hal yang harus dilakukan terlebih dahulu oleh aplikasi ini nantinya. Pertama dilakukan penghitungan ukuran *file* informasi yang akan disisipkan. Dan yang kedua adalah penghitungan LSB dari calon *carrier* apakah mampu menampung keseluruhan *file* informasi tersebut. Jika tidak, maka harus ditambahkan *file* lainnya hingga mampu menampung keseluruhan *file* informasi yang akan disisipkan dan kemudian mempersiapkan pembagian *file* informasi untuk disisipkan ke dalam lebih dari satu *carrier file.* Proses penyisipan dan ekstraksi informasi dapat dilihat pada gambar 3.1.

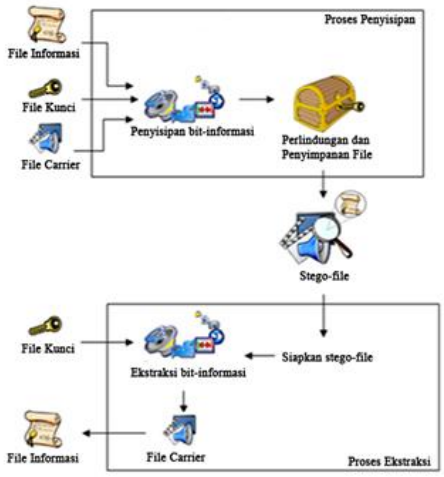
Penjelasan Skema Proses Kerja

a. Proses Penyisipan

Proses penyisipan dimulai dengan menyiapkan *key file, file* informasi, dan *carrier file*. Langkah berikutnya adalah menentukan LSB dari *file carrier*, dan kemudian menyisipkan informasi *key file* dan *file* informasi di dalamnya.

Untuk penyisipan *file* citra digital, maka pada bagian ini akan menggunakan algortima *simple LSB substitution* mekanisme penyisipan informasinya dan menggunakan dukungan `system.dll`. Sementara untuk proses penyisipan informasi pada *file* .wav akan digunakan

Algoritma *simple LSB substitution* dan algoritma ITSAS dengan dukungan system.dll dan waveIn/waveOut API. Dan untuk proses penysipan pada *file* .avi standar akan digunakan algoritma *simple LSB substitution* dengan dukungan `system.dll` dan `avifil32.dll`. Kesemua proses ini dilanjutkan dengan proses penulisan kembali dan penyimpanan *carrier* menjadi *stego-file.* Setelah proses *hiding* selesai maka *carrier file* akan disimpan menjadi *stego-file.*



Gambar 3. 1 Skema Proses Penyisipan Dan Ekstraksi.

b. Proses Ekstraksi

Proses ekstraksi informasi dimulai dengan menyiapkan *stego-file* dan *key file*. Aplikasi steganografi akan mencocokkan informasi *key file* yang ada dalam *stego-file* dengan *key file* yang disertakan saat proses ekstraksi. Dan jika terjadi

kecocokan maka *file* informasi dapat diekstrak kembali.

Proses ekstraksi informasi secara umum akan dilakukan dengan cara membalik algoritma proses penyisipan dan juga menggunakan *library* yang digunakan proses penyisipan tersebut.

3.2. Perancangan Sistem

Prosedur perancangan sistem secara umum untuk pembangunan aplikasi steganografi pada *file* multimedia ini terdiri atas beberapa tahap, antara lain meliputi perancangan :

1. Proses

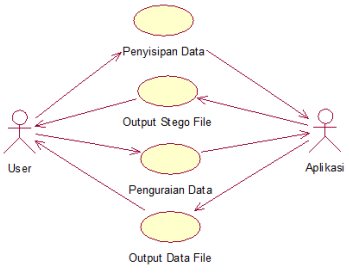
Perancangan proses yang dimaksudkan adalah bagaimana sistem akan bekerja, proses-proses yang digunakan, mulai dari user melakukan input kemudian hingga aplikasi mengeluarkan *output* berupa stego *file* pada proses penyisipan (*hiding*). Dan juga saat user melakukan input *stego file* dan *key file* hingga aplikasi memberikan *output* berupa *secret file* dan *carrier file* pada proses penguraian (*extracting*).

2. *Antarmuka*

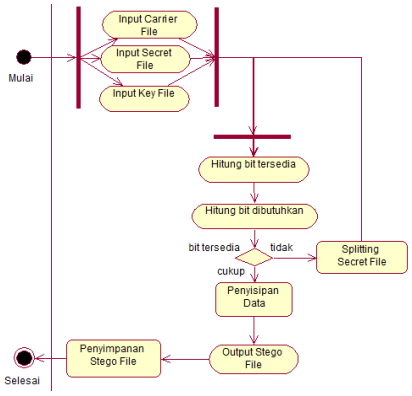
Perancangan antarmuka mengandung penjelasan tentang desain dan implementasi sistem yang digunakan dalam sistem yang dibuat dan diwujudkan dalam tampilan antarmuka sang menghubungkan *user* dengan aplikasi. Gambar 3.2 menunjukkan *use case* diagram dari sistem yang akan dibangun.

Proses utama yang dilakukan ada empat yaitu proses penyisipan informasi, proses output stego image, proses penguraian informasi dan proses output informasi
file. Berikut *activity diagram* dari keempat proses tersebut :

Proses yang dijelaskan oleh gambar 3.3 berlangsung saat *user* ingin melakukan penyisipan informasi ke dalam *file* multimedia. Dari proses penyisipan informasi maka diperoleh hasil output *file* berupa *stego-file*. Proses ini dijelaskan oleh gambar 3.4.



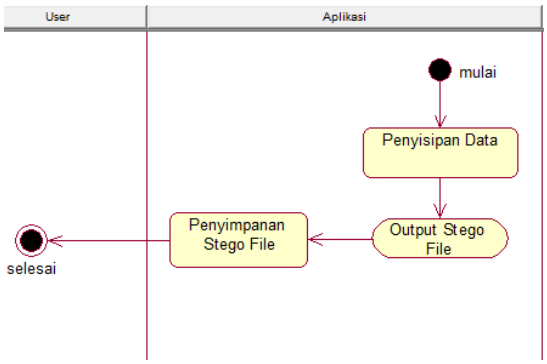
Gambar 3. 2 *Use Case Diagram* Aplikasi



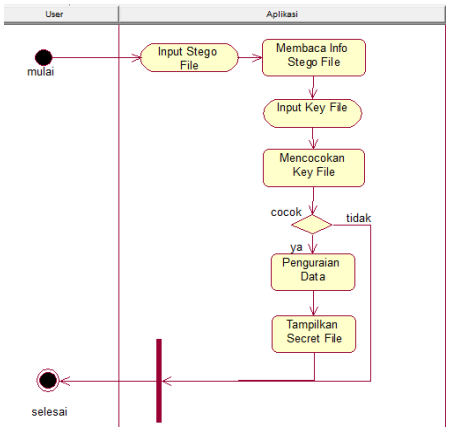
Gambar 3. 3 *Activity Diagram* Penyisipan Informasi

Proses yang ditunjukkan oleh gambar 3.5 berlangsung saat *user* ingin melakukan ekstraksi informasi dari *stego-file.*

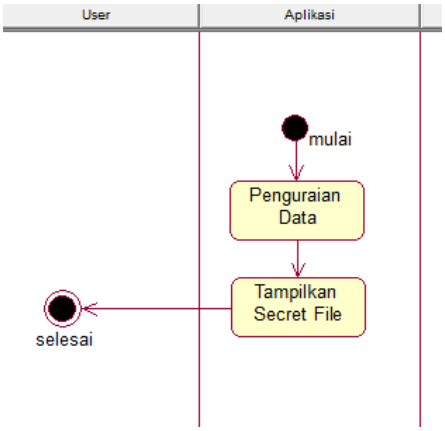
Dari hasil proses penguraian informasi, diperoleh hasil keluaran berupa *secret file* yang telah disisipkan pada proses sebelumnya. Hal ini ditunjukkan oleh gambar 3.6.



Gambar 3. 4 *Activity Diagram* Output Stego-File



Gambar 3. 5 *Activity Diagram Penguraian* Informasi



Gambar 3. 6 *Activity Diagram Output File* Informasi

3.3. Uraian Perancangan Sistem

Perancangan sistem ini terdiri atas dua tahap, yaitu tahap penyisipan informasi dan tahap ekstraksi informasi yang akan diuraikan pada sub bab di bawah ini :

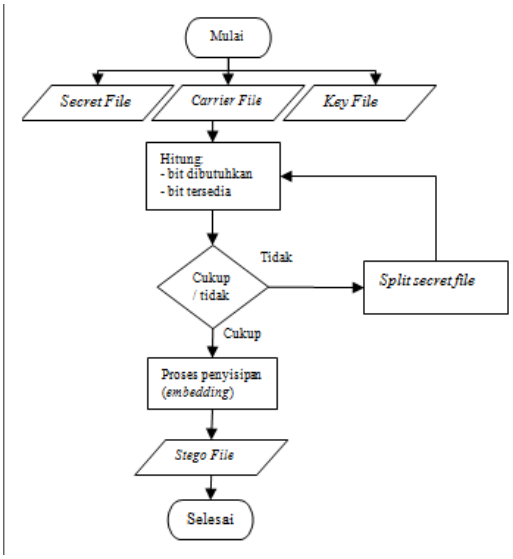
3.3.1. Tahap Penyisipan Informasi

Tahap penyisipan informasi merupakan tahap penyisipan atau penyamaran suatu informasi ke dalam *file* multimedia yang bertujuan untuk menyembunyikan informasi tersebut agar tidak terlihat oleh pihak yang tidak berhak.

Pada tugas akhir ini, dilakukan beberapa teknik penyisipan informasi ke dalam *file* multimedia bergantung dari jenis *file* multimedia yang digunakan. Adapun metode yang digunakan adalah *Least Significant Bit Modification* untuk proses penyisipan informasi.

3.3.1.1. Diagram Alir Embedding Informasi

Dari gambar 3.7 dapat dijelaskan langkah-langkah proses sebagai berikut: setelah memulai sistem (start), selanjutnya *user* melakukan input untuk *secret file, carrier file, key file*. Kemudian lakukan penghitungan bit yang dibutuhkan oleh key *file* dan secret *file*, serta berapakah bit yang mampu disediakan oleh carrier *file*, apakah bit yang tersedia mencukupi atau tidak. Jika tidak mencukupi maka akan diambil langkah-langkah untuk mencukupkan, jika yang dipilih adalah membagi secret *file* ke dalam lebih dari satu carrier maka akan dilakukan penambahan carrier *file*. Selanjutnya akan dimulai proses penyisipan informasi. Dari proses ini, akan dihasilkan stego *file*.

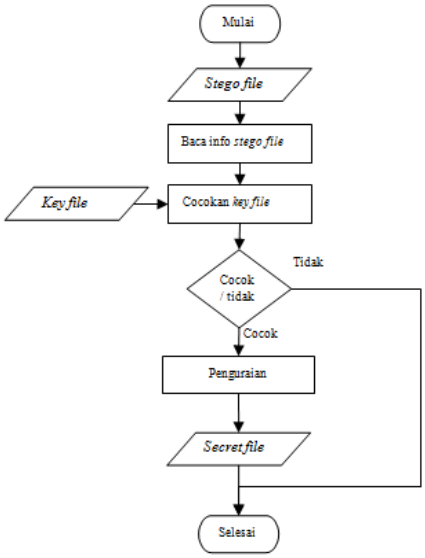


Gambar 3. 7 Diagram Alir Penyisipan Informasi

3.3.2. Tahap Ekstraksi Informasi

Pada tahap ini akan dilakukan proses ekstraksi informasi yang telah disisipkan dalam stego-file.

3.3.2.1. Diagram Alir Embedding Informasi



Gambar 3. 8 Diagram Alir Ekstraksi Informasi

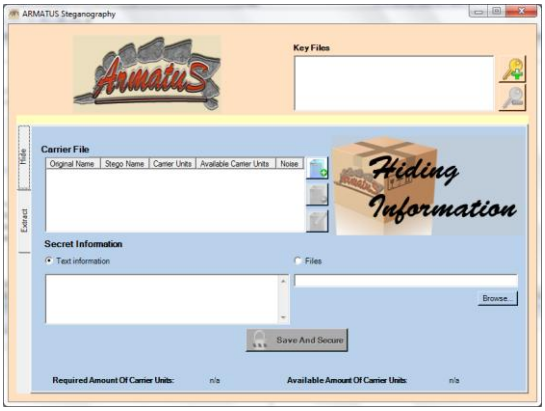
3.3.4. Tampilan Antar Muka

Sesuai dengan algoritma dan flowchart maka dibuatlah tampilan-tampilan yang bertujuan untuk memudahkan pengguna untuk menjalankan aplikasi ini. Aplikasi piranti lunak steganografi ini diberi nama Armatus yang merupakan salah satu spesies dari Stegosaurus yang memiliki kemiripan nama dengan stego-file yang digunakan dalam proses steganografi. Tampilan-tampilan yang ada dibuat semenarik mungkin namun tetap sederhana

dan bersifat fungsional termasuk dengan pembuatan logo yang menjadi simbol identitas dari aplikasi ini. Aplikasi ini dibuat dalam bahasa Inggris semata-mata karena alasan bahwa sebagian besar pengguna lebih familiar dengan istilah teknis berbahasa Inggris dibandingkan dengan bahasa Indonesia.

3.3.4.1. Menu Penyisipan Informasi (Hide)

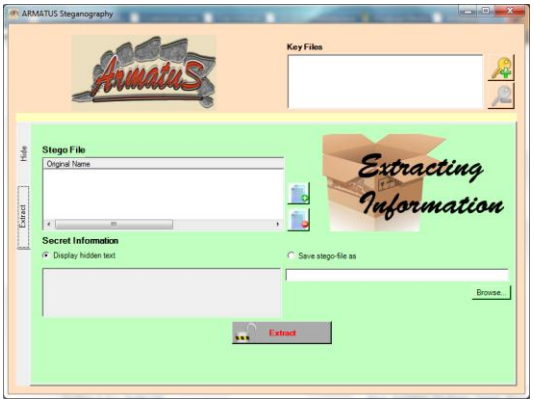
Pada menu penyisipan informasi atau juga disebut hiding karena istilah embedding kurang familiar.



Gambar 3.9 Tampilan Menu Hide

3.3.4.2 Menu Ekstraksi Informasi

Pada menu ekstraksi ini akan dilakukan ekstraksi informasi rahasia yang telah disisipkan sebelumnya.



Gambar 3.10 Tampilan Menu Extract

4. PENGUJIAN DAN ANALISA

4.1 Lingkungan Uji Coba

Lingkungan uji coba aplikasi ini dilakukan pada sebuah notebook Compaq Pressario V3700 dalam karakteristik spesifikasi lingkungan sebagai berikut :

- Intel Pentium Dual Core T2930 @ 1,86GHZ
- DDR II SD-RAM 2GB
- Hardisk Seagate 16GB, 5400 rpm, SATA
- VGA Mobile Intel 965 Express
- DVD-RW Optical Drive
- Fast Ethernet Adapter
- Intel Chipset
- Display 14.1" WXGA (1280 x 800) TFT
- Sistem Operasi :
  - Microsoft Windows Vista Professional 32-bit

4.2 Skenario Pengujian

Sebelum melakukan pengujian maka akan ditentukan tentang skenario pengujian yang digunakan pada proyek akhir ini. Skenario ini dibuat dengan tujuan untuk dapat melakukan pengujian fungsi fitur yang dimiliki Armatus namun dapat dilakukan dalam waktu yang relatif singkat mengingat banyaknya pengujian yang harus dilakukan. Skenario ini akan memberikan batasan ukuran file informasi rahasia yang kita sisipkan sebesar ≤ 100KB, file kunci yang berupa file teks sebesar ≤10KB dan ukuran carrier yang digunakan ≤ 6MB. File-file carrier yang digunakan pada pengujian ini ditunjukkan oleh tabel 4.1.

No	Nama File	Tipe	Ukuran
1	Image1.bmp	Citra digital	2.25 MB
2	Image2.bmp	Citra digital	769 KB
3	Image3.bmp	Citra digital	2.25 MB
4	Audio1.wav	Audio digital	6 MB
5	Audio2.wav	Audio digital	5.55 MB
6	Audio3.wav	Audio digital	70 KB
6	Video1.avi	Video digital	2.49 MB
7	Video2.avi	Video digital	5.5 MB

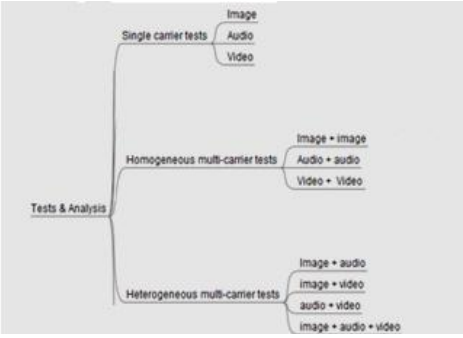
Tabel 4. 1 File-File Carrier yang digunakan

Dan dalam pengujian ini juga akan digunakan informasi rahasia berupa file-file yang dirasa mewakili tipe-tipe file yang kita gunakan sehari-hari. file-file tersebut mewakili file dokumen, file audio, file video, file kompresi, dan file citra digital. File-file yang digunakan sebagai informasi rahasia ini dapat dilihat pada tabel 4.2. Selain itu berbagai kombinasi pengujian yang digunakan dalam pengujian dapat dilihat pada gambar 4.2 Pengujian performa dan pengujian kualitas ekstraksi juga dilakukan dalam proyek akhir ini.

No	Nama File	Tipe	Ukuran
1	Secret1.doc	Dokumen	10 KB
2	Secret2.jpg	Citra digital	10 KB
3	Secret3.avi	Video	100 KB

4	Secret4.zip	File Kompresi	10 KB
5	Secret5.mp3	Audio digital	10 KB

Tabel 4. 2 File-File Informasi Rahasia



Gambar 4. 1 Variasi Pengujian

4.3 Pengujian Sistem

4.3.1 Pengujian Tahap Penyisipan Informasi

Pada Pada bagian ini akan dilakukan proses penyisipan informasi pada file carrier tunggal dan multi-carrier. Untuk pengujian multi-carrier terdapat pengujian homogen yaitu pengujian untuk carrier yang sejenis dan juga heterogen.

Contoh hasil pengujian ini dapat dilihat dalam tabel 4.3. Sementara untuk hasil pengujian lain tidak dapat ditampilkan karena keterbatasan tempat.

No	File				Hasil
	Key	Carrier	Secret	Stego	
1	Key.txt	Audio1.wav	Test.doc	AudioStego.wav	x
2	Key.txt	Audio1.wav	Test.avi	AudioStego2.wav	x
3	Key.txt	Audio1.wav	Test.mp3	AudioStego3.wav	x
4	Key.txt	Audio1.wav	Test.zip	AudioStego4.wav	x
5	Key.txt	Audio1.wav	Test.jpg	AudioStego5.wav	x
6	Key.txt	Image1.bmp	Test.doc	ImageStego.bmp	x
7	Key.txt	Image1.bmp	Test.avi	ImageStego2.bmp	x
8	Key.txt	Image1.bmp	Test.mp3	ImageStego3.bmp	x
9	Key.txt	Image1.bmp	Test.zip	ImageStego4.bmp	x
10	Key.txt	Image1.bmp	Test.jpg	ImageStego5.bmp	x
11	Key.txt	Video.avi	Test.doc	VideoStego.avi	√
12	Key.txt	Video.avi	Test.avi	VideoStego2.avi	√
13	Key.txt	Video.avi	Test.mp3	VideoStego3.avi	√
14	Key.txt	Video.avi	Test.zip	VideoStego4.avi	√
15	Key.txt	Video.avi	Test.jpg	VideoStego5.avi	√

Tabel 4. 3 Pengujian Penyisipan Terhadap File Carrier Tunggal

Pada saat proses penyisipan ini dihasilkan sebuah file baru yang disebut stego-file yang berisi informasi rahasia yang kita sisipkan dan informasi mengenai file kunci yang melindunginya. gambar 4.2 dan gambar 4.3 akan menunjukkan carrier file yang belum menyimpan apapun dibandingkan dengan stego-file. Untuk perbandingan spesifik mengenai keduanya akan dilakukan pada pengujian performa.

Gambar 4. 2 Carrier File

Gambar 4. 3 Stego-file

Setelah berhasil menyisipkan pada file carrier tunggal maka akan dilakukan pengujian keberhasilan proses penyisipan pada lebih dari satu carrier atau yang disebut multi-carrier yang merupakan esensi dari proyek akhir ini. Proses penambahan carrier ini dilakukan karena kurangnya carrier unit yang tersedia untuk proses penyisipan informasi. Untuk melakukan penambahan file carrier, cara yang dilakukan adalah dengan meng-input-kan carrier file tambahan dengan cara yang sama dengan cara meng-input-kan carrier file sebelumnya. Proses penambahan carrier file ini dapat terus dilakukan hingga jumlah carrier unit yang tersedia lebih besar daripada jumlah carrier unit yang dibutuhkan. Hal ini dapat dilihat dari petunjuk informasi carrier unit pada bagian bawah menu penyisipan (hide)

Gambar 4. 2 Petunjuk Informasi Carrier Unit

Dari keseluruhan pengujian terhadap berbagai variasi skenario pengujian proses peyisipan terjadi beberapa kegagalan namun bukan disebabkan oleh kegagalan Armatus melainkan tidak cukup tersedianya *carrier unit.* Dan proses penyisipan untuk carrier yang memiliki cukup carrier unit berhasil dilakukan dengan baik. Dari pengujian tersebut diketahui bahwa carrier terbaik adalah file video karena terdiri dari sekumpulan citra digital.

4.3.2 Pengujian Tahap Ekstraksi Informasi

Setelah mampu untuk menyisipkan informasi maka akan dilakukan pengujian untuk memperoleh kembali informasi yang disisipkan dari skenario pengujian yang ada.

Setelah kita berhasil melakukan proses penyisipan maka kita perlu melakukan pengujian untuk mengetahui keberhasilan proses ekstraksi yang kita lakukan terhadap seluruh skenario penyisipan kita. Hasil untuk pengujan ekstraksi terhadap carrier tunggal dapat dilihat pada tabel 4.4. Sementara untuk hasil pengujian pada multi-carrier dapat dilihat pada lembar lampiran karena keterbatasan margin yang ada.

No	File			Hasil
	Key	Stego	Secret	
1	Key.txt	AudioStego.wav	Test.doc	x
2	Key.txt	AudioStego2.wav	Test.avi	x
3	Key.txt	AudioStego3.wav	Test.mp3	x
4	Key.txt	AudioStego4.wav	Test.zip	x
5	Key.txt	AudioStego5.wav	Test.jpg	x
6	Key.txt	ImageStego.bmp	Test.doc	x
7	Key.txt	ImageStego2.bmp	Test.avi	x
8	Key.txt	ImageStego3.bmp	Test.mp3	x
9	Key.txt	ImageStego4.bmp	Test.zip	x
10	Key.txt	ImageStego5.bmp	Test.jpg	x
11	Key.txt	VideoStego.avi	Test.doc	√
12	Key.txt	VideoStego2.avi	Test.avi	√
13	Key.txt	VideoStego3.avi	Test.mp3	√
14	Key.txt	VideoStego4.avi	Test.zip	√
15	Key.txt	VideoStego5.avi	Test.jpg	√

Tabel 4. 4 Ekstraksi Pada File Carrier Tunggal

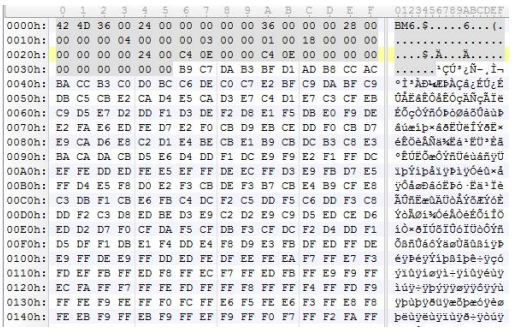
4.3.3 Pengujian Performa

Setelah dilakukan pengujian proses penyisipan dan ekstraksi informasi makan dilakukan prengujian untuk mengetahui sejauh mana unjuk kerja Armatus. Dalam pengujian ini akan dilakukan pengamatan terhadap hal-hal yang berkenaan langsung ataupun tidak langsung terhadap unjuk kerja Armatus.

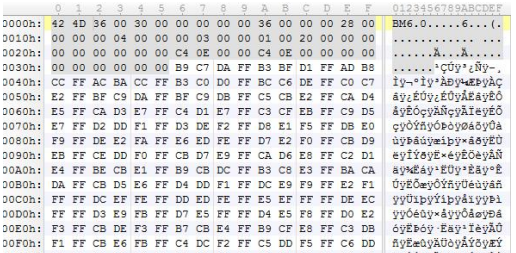
4.3.3.1 Efektifitas Penyisipan

Pada pengujian ini akan dilakukan pengamatan mengenai efektifitas peroses penyisipan dengan menggunakan 010 Editor sebuah aplikasi yang digunakan untuk mengetahui susunan binari dari file carrier yang kita gunakan sebelum dan sesudah proses penyisipan. Pada pengamatan ini akan dilakukan pengamatan terhadap file image1.bmp dan imagestego.bmp. Perbandingan mengenai keduanya dapat dilihat dari gambar 4.5 dan 4.6.

Dari perbandingan kedua struktur binari di atas dapat dilihat bahwa informsai yang disisipkan disebar keseluruh bagian dari file *carrier* bukannya dalam blok tertentu yang dikhawatirkan akan menunjukkan dimana kita menyisipkan informasi. Dengan cara seperti ini diharapkan pihak-pihak yang tidak berkepentingan yang ingin menguraikan iniformasi ini dengan cara mempelajari struktur binari dari stego-file untuk mempelajari dimana kita menyembunyikan informasi akan kesulitan.



Gambar 4. 3 Struktur Binari Image1.bmp



Gambar 4. 4 Struktur Binari Imagestego.bmp

4.3.3.3 Lain-lain

4.3.3.3.1 Perbandingan Channel Warna

Pada bagian ini akan dilaporkan mengenai informasi tambahan yang didapat dari proses-proses pengujian yang telah dilakukan namun pada dasarnya hasilnya tidak mempengaruhi proses penyisipan namun tetap memiliki hubungan dengan Armatus.

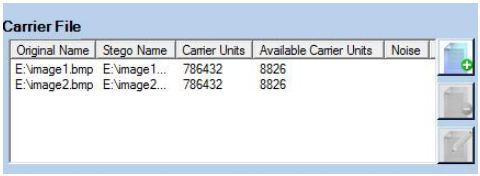
Informasi yang didapat adalah mengenai perbandingan penggunaan carrier citra digital yang memiliki *channel* RGB dan *greyscale* dalam proses penyisipan informasi rahasia.

File yang akan kita amati adalah image1.bmp yang memiliki *channel* RGB dan image2.bmp yang memiliki *channel greyscale*. Perbandingan kedua gambar tersebut dapat dilihat pada gambar 4.7 dan 4.8.

Gambar 4. 6 Gambar Image1.bmp

Gambar 4. 5 Gambar Image2.bmp

File image1.bmp memiliki ukuran yang lebih besar yaitu 2.25 MB dan image2.bmp memiliki ukuran 769 KB namun keduanya memiliki ukuran *carrier unit* yang sama besar. Hal ini dapat dilihat dari bagian input *carrier file* yang menampilkan *carrier unit* kedua file tersebut sebagaimana ditunjukkan oleh gambar 4.9.



Gambar 4. 7 Perbandingan Image1.bmp Dan Image2.bmp

Hal ini terjadi karena adanya perbedaan nilai warna (*colour value*) dari file dengan format RGB dan *greyscale*. Namun sayangnya meski saat sebelum proses penyisipan keduanya ukuran yang berbeda namun saat digunakan untuk melakukan proses steganografi dengan informasi rahasia dan file kunci yang besarnya sama maka kedua file yang difungsikan sebagai *carrier* tersebut akan memiliki ukuran yang sama. Hal ini terjadi karena file *carrier* yang disimpan sebagai stego-file akan memiliki *channel* RGB. Namun meskipun demikian, tidak ada perbedaan kasat mata yang nampak antara image2.bmp saat sebelum proses dan setelah menjadi stego-file.

4.3.3.3.2 Modifikasi Stego-file Yang Diizinkan

Pengujian ini akan dilakukan untuk mengetahui sejauh mana kita dapat memodifikasi stego-file tanpa merusak informasi yang kita sisipkan. Adapun pengujian-pengujian yang dilakukan adalah sebagai berikut:

- Mengganti nama stego-file: Saat mengganti nama stego-file dan kemudian kita melakukan ekstraksi maka informasi yang kita sisipkan akan tetap dapat diekstrak. Hal ini dimungkinkan karena letak informasi mengenai nama file tidak bersinggungan dengan bit-bit yang kita gunakan dalam proses penyisipan informasi.
- Mengkompres stego-file: Saat distribusi stego-file (misalnya lewat email) terkadang kita merasa perlu untuk mengkompres stego-file untuk memudahkan dan mempersingkat waktu distribusi. Berdasarkan pengujian yang dilakukan, informasi yang disisipkan tetap dapat diekstrak selama kompresi tersebut bisa mengembalikan struktur file-stego kembali ke sediakala.
- Merubah format stego-file: Saat pengujian ini dilakukan diharapkan dapat diketahui apakah informasi yang disisipkan dapat diekstrak kembali jika stego-file yang kita miliki diubah menjadi bentuk lain seperti .bmp diubah menjadi .jpg dan .wav duibah menjadi .mp3. Setelah pengujian didapati bahwa perubahan format stego-file akan merusak informasi yang kita sisipkan. Hal ini terjadi karena adanya perbedaan nilai-nilai yang dibutuhkan saat proses ekstraksi. Contohnya adalah saat kita merubah stego-file dalam format bitmap menjadi JPEG maka *colour value* stego-file akan berubah sehingga akan menjadi tidak mungkin untuk melakukan proses ekstraksi.

5. KESIMPULAN DAN SARAN

Pada bab-bab sebelumnya, mulai dari bab I sampai dengan bab IV telah diuraikan beberapa hal yang berhubungan dengan pembuatan aplikasi ini, mulai dari latar belakang, dasar teori, perancangan dan pembuatan aplikasi, sampai dengan implementasinya yang disertai uji coba dan analisa. Selanjutnya pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari hasil-hasil pengujian aplikasi dan beberapa saran dengan harapan untuk lebih menyempurnakan perancangan yang telah dibuat.

5.1 KESIMPULAN

Setelah melakukan serangkaian pengujian terhadap aplikasi menganalisa hasil yang didapatkan dari pengujian tersebut, maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi ini mampu menyisipkan informasi tanpa merusak carrier file ataupun data yang telah disisipkan.
2. Aplikasi ini juga mampu mengekstrak pada berkas file stego.

3. Aplikasi ini menawarkan fitur pengamanan dengan menggunakan file kunci.

5.2 SARAN

Dari beberapa kesimpulan yang diambil, dapat diambil saran – saran yang dapat digunakan dalam membuat suatu aplikasi :

1. Untuk selanjutnya diharapkan mampu melakukan operasi steganography pada berbagai variasi ekstensi file multimedia.
2. Tampilan user interface untuk lebih disempurnakan agar lebih memudahkan user untuk mengoperasikan aplikasi ini.

DAFTAR PUSTAKA

[1] Maulana, Ahmad Mansur, "Data Hiding Steganograph Pada *File* Image Menggunakan Metode Least Significant Bit", PENS-ITS, Surabaya, 2009

[2] Romdhoni, Muhamad Arif, "Kriptografi Visual Pada Citra Biner Dan Berwarna Serta Pengembangannya Dengan Steganografi Dan Fungsi Xor", Teknik Informatika ITB, Bandung,2008

[3] Kirovski, D., Malvar, H.S. , "Microsoft Audio Watermarking Tool", IEEE, 2003.

[4] No name,"Steganography", *en.wikipedia.org/wiki/Steganography*, tanggal akses 11 Juli 2010 pukul 17.30

[5] No name,"*File* Format", *en.wikipedia.org/wiki/File_format*, tanggal akses 11 Juli 2010 pukul 17.30

[6] no name, "Least Significant Bit", en.wikipedia.org/wiki/least_significant_bit, tanggal akses 11 Juli 2010 pukul 17.30

[7] Chan, Chi-Kwong, Cheng, L.M., "Hiding data in images by simple LSB substitution", Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong, 2003.

[8] Agaian, Sos S., Akopian, David and D'Souza1, Sunil A. "Two algorithms in digital audio steganography Using quantized frequency domain embedding and Reversible *integer* transforms", Non-linear Signal Processing Lab, University of Texas at San Antonio,Texas,2004

[9] Tian, Jun, "High Capacity Reversible Data Embedding and Content Authentication," IEEE Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 517-520, 2003.

[10] Davis, Stephen Randy., Spar, Chuck., "C# 2005 for Dummies", Wiley Publishing,Inc, New Jersey, 2006

[11] Al-Mualla, Dr. Muhammed, Al – Ahmad, Prof. Husein, "Information Hiding: Steganography and Watermarking", Etisalat College of Engineering, UAE, 2003.

[12] Hakim A, Muhammad, "Makalah Studi dan Implementasi Steganography Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah", Teknik Informatika ITB, Bandung2007

[13] Munir, Rinaldi, "Diktat Kuliah IF5054 Kriptografi", Teknik Informatika ITB, Bandung, 2005

[14] No name, "Introduction to Steganography", io.acad.athabascau.ca/~grizzlie/Comp607Tanggal akses : 11 Juli 2010 pukul 17.1