

IMPLEMENTASI STEGANOGRAFI LSB DENGAN ENKRIPSI VIGENERE CIPHER PADA CITRA JPEG

Tri Cahyadi^{*)}

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)Email : trie_cahyadi@windowslive.com}

Abstrak

Saat ini teknologi informasi sudah sangat berkembang menjadi salah satu media yang paling populer di dunia. Sayangnya dengan berkembangnya teknologi informasi semakin berkembang pula tindak penyalahgunaan informasi yang bukan haknya. Dengan berbagai teknik banyak yang mencoba untuk mengakses informasi yang bukan haknya. Maka dari itu sejalan dengan berkembangnya media internet ini harus juga dibarengi dengan perkembangan pengamanan informasi. Berbagai macam teknik digunakan untuk melindungi informasi yang dirahasiakan dari orang yang tidak berhak, salah satunya adalah teknik steganografi. Pada tugas akhir ini, dibuat aplikasi steganografi yang bertujuan untuk mengamankan informasi berupa pesan teks dengan menyisipkan (menyembunyikan) kedalam pesan lainnya yaitu pada citra digital dengan menggunakan metode algoritma LSB (Least Significant Bit) dan dengan enkripsi Vigenere Cipher. Hasil dari aplikasi ini adalah dapat menyisipkan pesan tersembunyi berupa teks ke dalam berkas citra digital berformat JPEG dan dapat mengekstraksi kembali pesan tersembunyi tersebut dari dalam citra (stego-image).

Kata kunci: Steganografi, Least Significant Bit (LSB), Vigenere Cipher.

Abstract

Today information technology has greatly evolved into one of the most popular media in the world. Unfortunately, with the rapid development of information technology is growing also follow misuse of information that is not right. With the many techniques that attempt to access information that is not right. Therefore in line with the growth of the Internet media should also be coupled with the development of information security. Various techniques are used to protect confidential information from unauthorized people, one of which is the technique of steganography. In this final report, made steganography application that aims to secure the information in the form of a text by inserting (hide) into another message that the digital image using the algorithm LSB (Least Significant Bit) and encryption Vigenere Cipher. The result of this application is to insert hidden messages (text file) into digital image (JPEG) and can extract the hidden message back from the image (stego-image).

Keywords: Steganography, Least Significant Bit (LSB), Vigenere Cipher.

1. Pendahuluan

Steganografi adalah seni dan ilmu untuk menyembunyikan pesan dalam sebuah pesan. Seni dan ilmu ini telah diterapkan sejak dahulu oleh orang Yunani kuno yang menyembunyikan pesan dengan cara membuat tato di kepala pembawa berita yang dibotaki dan menunggu sampai rambutnya tumbuh. Teknik steganografi lainnya adalah dengan menggunakan "invisible ink" (tinta yang tidak tampak). Tulisan yang ditulis dengan menggunakan *invisible ink* ini hanya dapat dibaca jika kertas tersebut diletakkan di atas lampu atau diarahkan ke matahari. Ketika perang dunia pertama,

orang Jerman menyembunyikan pesan dalam bentuk "microdot", yaitu titik-titik yang kecil. Agen dapat membuat foto kemudian mengecilkannya sampai sekecil titik di tulisan dalam buku. Buku ini kemudian bisa dibawa-bawa tanpa ada yang curiga bahwa tanda titik di dalam tulisan di buku itu berisi pesan ataupun gambar.

Kerahasiaan pesan yang ingin disampaikan merupakan faktor utama sehingga digunakan metode steganografi. Dengan metode steganografi, pesan yang ingin disampaikan disembunyikan dalam suatu media umum sehingga diharapkan tidak akan menimbulkan kecurigaan dari pihak lain yang tidak diinginkan untuk mengetahui

pesan rahasia tersebut. Oleh sebab itu metode steganografi terus digunakan dan dikembangkan sampai saat ini.

Atas dasar uraian diatas, maka pada penulisan tugas akhri ini akan membahas mengenai bagaimana mengamankan suatu pesan dengan menyisipkan (menyembunyikan) kedalam pesan lainnya yaitu file citra dengan menggunakan algoritma LSB (*Least Significant Bit*) pada suatu aplikasi steganografi.

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat sebuah program steganografi yang mampu menyisipkan data atau informasi berupa teks dengan menggunakan teknik LSB (*Least Significant Bit*) dan dengan Enkripsi Vigenere Cipher pada media citra berformat JPEG dengan menggunakan bahasa pemrograman Microsoft Visual Basic 6.

2. Metode

2.1 Steganografi

Steganografi berasal dari bahasa Yunani yaitu *Steganós* yang berarti menyembunyikan dan *Graptos* yang artinya tulisan, sehingga secara keseluruhan artinya adalah "tulisan yang disembunyikan". Secara umum steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan sesuatu informasi.

Secara teori, semua file umum yang ada di dalam komputer dapat digunakan sebagai media, seperti file gambar berformat JPEG, GIF, BMP, atau di dalam musik MP3, atau bahkan di dalam sebuah film dengan format WAV atau AVI. Semua dapat dijadikan tempat bersembunyi, asalkan file tersebut memiliki bit-bit data redundan yang dapat dimodifikasi. Setelah dimodifikasi file media tersebut tidak akan banyak terganggu fungsinya dan kualitasnya tidak akan jauh berbeda dengan aslinya.

Steganografi merupakan suatu ilmu atau seni dalam menyembunyikan informasi dengan memasukkan informasi tersebut kedalam pesan lain. Dengan demikian keberadaan informasi tersebut tidak diketahui oleh orang lain. Tujuan dari steganografi adalah menyembunyikan keberadaan pesan dan dapat dianggap sebagai pelengkap dari kriptografi yang bertujuan untuk menyembunyikan isi pesan. Oleh karena itu, berbeda dengan kriptografi, dalam steganografi pesan disembunyikan sedemikian rupa sehingga pihak lain tidak dapat mengetahui adanya pesan rahasia. Pesan rahasia tidak diubah menjadi karakter aneh seperti halnya kriptografi. Pesan tersebut hanya disembunyikan ke dalam suatu media berupa gambar, teks, musik, atau media digital lainnya dan terlihat seperti pesan biasa.

Beberapa contoh media penyisipan pesan rahasia yang digunakan dalam teknik Steganography antara lain adalah

1. **Teks.** Dalam algoritma Steganografi yang menggunakan teks sebagai media penyisipannya biasanya digunakan teknik NLP sehingga teks yang telah disisipi pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.
2. **Audio.** Format ini pun sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar. Sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.
3. **Citra.** Format ini juga sering digunakan, karena format ini merupakan salah satu format file yang sering dipertukarkan dalam dunia internet. Alasan lainnya adalah banyaknya tersedia algoritma Steganografi untuk media penampung yang berupa citra.
4. **Video.** Format ini memang merupakan format dengan ukuran file yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

2.2 Metode Algoritma Kriptografi

Untuk melakukan kriptografi digunakan algoritma kriptografi. Algoritma kriptografi terdiri dari dua bagian, yaitu fungsi enkripsi dan dekripsi. Enkripsi adalah proses untuk mengubah *plaintext* menjadi *ciphertext*, sedangkan dekripsi adalah kebalikannya yaitu mengubah *ciphertext* menjadi *plaintext*. Terdapat dua jenis algoritma kriptografi berdasar jenis kuncinya[1], yaitu :

1. Algoritma Simetri, adalah algoritma yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Algoritma standar yang menggunakan prinsip kunci simetri antara lain OTP, DES, RC2, RC4, RC5, RC6, IDEA, Twofish, Blowfish, dan lain lain.
2. Algoritma Asimetri, adalah algoritma yang kunci untuk enkripsi dan dekripsinya jauh berbeda. Algoritma standar yang termasuk algoritma asimetri adalah ECC, LUC, RSA, EI, Gamal dan DH.

Penilaian sebuah algoritma steganography yang baik dapat di nilai dari beberapa faktor yaitu :

1. **Imperctibility.** Keberadaan pesan rahasia dalam media penampung tidak dapat dideteksi oleh inderawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan *coverttext*-nya. Jika *coverttext* berupa audio (misalnya berkas file *mp3*, *wav*, *midi* dan sebagainya), maka indera telinga tidak dapat mendeteksi perubahan pada file *stegotext*-nya.
2. **Fidelity.** Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan itu tidak dapat dipersepsi oleh inderawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan dapat membuat citra *stegotext* sukar dibedakan oleh mata dengan citra

covertext-nya. Jika *covertext* berupa audio (misalnya berkas file *mp3*, *wav*, *midi* dan sebagainya), maka audio *stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan pada file *stegotext*-nya.

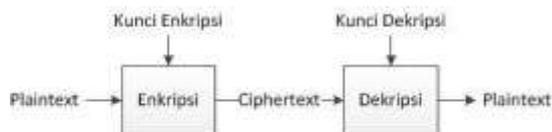
3. **Recovery.** Pesan yang disembunyikan harus dapat diungkapkan kembali (*reveal*). Karena tujuan steganography adalah *data hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

2.3 Metode Penyembunyian Data

Pada tugas akhir ini, akan digunakan metode LSB (*Least Significant Bit*) yang merupakan teknik penyembunyian data yang bekerja pada domain spasial. Tiga aspek yang berbeda yang mempengaruhi sifat sistem penyembunyian atau penyisipan pesan rahasia pada gambar adalah : kapasitas, keamanan, dan ketahanan. Kapasitas merujuk pada jumlah informasi yang dapat disembunyikan dalam medium cover. Keamanan adalah ketidakmampuan pengamat untuk mendeteksi pesan yang tersembunyi, dan ketahanan yaitu jumlah modifikasi *stego medium* yang dapat bertahan sebelum musuh dapat merusak pesan rahasia yang tersembunyi tersebut.

Pada dasarnya komunikasi steganografi yaitu, pengirim dan penerima setuju pada suatu sistem steganografi dan membagi pakai (*sharing*) kunci rahasia untuk menentukan bagaimana suatu pesan dikodekan dalam citra digital. Untuk mengirim suatu pesan rahasia yang tersembunyi.

Proses utama dalam kriptografi ada dua, yaitu enkripsi dan dekripsi. Enkripsi adalah proses penyembunyian pesan dengan menggunakan key tertentu. Sedangkan dekripsi adalah proses pembacaan atau ekstraksi pesan dari *ciphertext*. Berikut ini gambaran umum dari proses tersebut.



Gambar 2.1 Enkripsi-Dekripsi

Berikut ini penjelesana mengenai istilah dan component utama yang sering dipakai dalam kriptografi:

1. **Plaintext** . Plaintext adalah pesan yang akan kita kirim atau simpan dalam bentuk aslinya. Plaintext dapat dibaca secara langsung dan bermakna.
2. **Ciphertext** . Ciphertext adalah pesan yang sudah kita enkripsi. Ciphertext tidak dapat dibaca secara langsung dan tidak bermakna.
3. **Enkripsi**. Enkripsi adalah proses penyembunyian pesan. Proses enkripsi merubah pesan plaintext menjadi ciphertext yang tidak bermakna. Pada algoritma saat ini, untuk melakukan enkripsi diperlukan suatu kunci.

4. **Dekripsi**. Dekripsi adalah proses mengekstraksi pesan yang ada dalam ciphertext. Proses dekripsi akan menghasilkan plaintext yang sama seperti sebelum dienkripsi. Dalam dekripsi diperlukan juga kunci.

5. **Key / kunci**. Key adalah suatu parameter yang digunakan untuk melakukan enkripsi maupun dekripsi. Kunci yang digunakan dapat berbentuk apapun seperti abjad, bilangan, atau bahkan dalam kriptografi modern dapat berupa bit.

Lewat notasi, proses tersebut dapat ditulis sebagai berikut:

Enkripsi

$$Ek(P) = C$$

E = fungsi enkripsi C = ciphertext

P = plaintext

K = key

Dekripsi

$$Dk(C) = P$$

D = fungsi dekripsi

C = ciphertext

P = plaintext

K = key

Semakin besar wadah(*cover-image*) yang digunakan untuk penyembunyian pesan maka semakin besar atau banyak pula jumlah karakter yang dapat disembunyikan dan semakin besar teks yang disembunyikan di dalam citra, semakin besar pula kemungkinan teks tersebut rusak akibat manipulasi pada citra penampung. Rumus untuk menghitung jumlah maksimal karakter yang dapat disisipkan ke gambar :

$$Max\ char = \frac{lebar\ gambar \times panjang\ gambar}{8\ bit\ karakter} = \frac{200 \times 200}{8} = 5000\ char$$

	0	1	2	3
0				
1				
2				
3				

Gambar 2.2 Gambar berukuran 4x4x8b

Ukuran teks yang akan disembunyikan bergantung pada ukuran gambar yang dijadikan sebagai wadah penyimpanan. Misalnya Suatu gambar yang digunakan sebagai wadah penyimpanan berukuran 200 x 200 x 8b, berarti gambar mempunyai panjang 200 piksel, lebar 200 piksel dan 8b menunjukkan format pikselnya 8 bit. Gambar tersebut mempunyai 40000 piksel, karena 1 karakter terdiri dari 8 bit (ASCII) maka pesan disisipkan pada setiap 8 piksel sehingga teks maksimal yang dapat disembunyikan pada gambar adalah sebanyak 40000/8=5000 karakter.

2.4 Metode Steganografi Least Significant Bit (LSB)

Teknik Steganografi dengan menggunakan metode modifikasi Least Significant Bit (*LSB*) adalah teknik yang paaling sederhana, pendekatan yang sederhana untuk menyisipkan informasi di dalam suatu citra digital

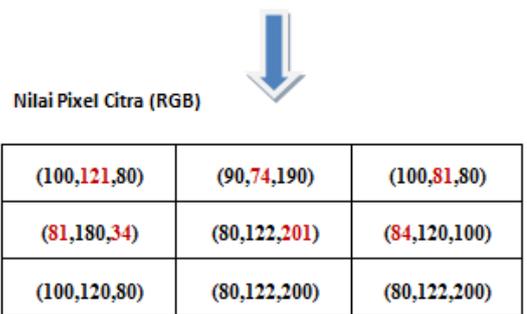
(medium cover). Mengkonversi suatu gambar dari format GIF atau BMP, yang merekonstruksi pesan yang sama dengan aslinya (*lossless compression*) ke JPEG yang *lossy compression*, dan ketika dilakukan kembali akan menghancurkan informasi yang tersembunyi dalam LSB.

Untuk menyembunyikan suatu gambar dalam LSB pada setiap byte dari gambar 24-bit, dapat disimpan 3 byte dalam setiap pixel. Gambar 1,024 x 768 mempunyai potensi untuk disembunyikan seluruhnya dari 2,359,296 bit (294,912 byte) pada informasi. Jika pesan tersebut dikompres untuk disembunyikan sebelum ditempelkan, dapat menyembunyikan sejumlah besar dari informasi. Pada pandangan mata manusia, hasil *stego-image* akan terlihat sama dengan gambar cover.

2.5 Algoritma LSB Proses penyisipan (Embedding) pesan ke Citra Digit



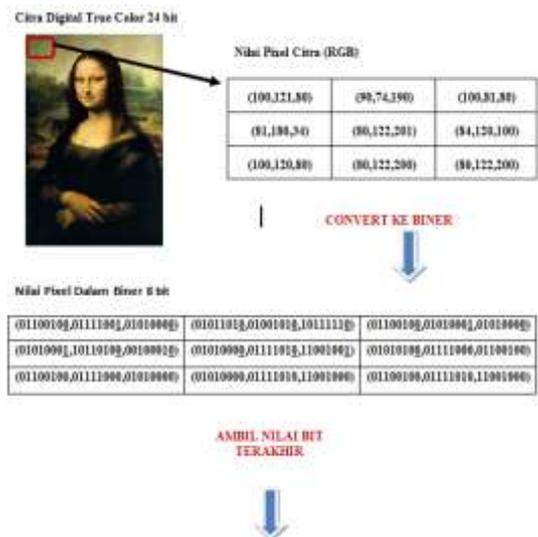
PESAN YANG AKAN DISISIPKAN ADALAH "AB"
 NILAI ASCII "A" ADALAH 65,
 BINERNYA 01000001
 NILAI ASCII "B" ADALAH 66,
 BINERNYA 01000010



Convert Nilai Desimal ke Pixel, maka Citra digital yang sudah disisipi pesan tidak tampak / perbedaannya jika di lihat dengan kasat mata.



2.6 Algoritma proses Dekripsi (Ekstraksi) Pesan dari Citra Digital



Ambil Nilai Bit Terakhir:
 01000001 = 65 = A
 01000010 = 66 = B
 PESAN HASIL EKSTRAKSI ADALAH "AB"

2.7 Metode Enkripsi Vigenere Cipher

Sandi Vigenere adalah metode menyandi teks alphabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci. Sandi Vigenere merupakan bentuk sederhana dari sandi polialfabetik. Kelebihan sandi ini dibanding sandi Caesar dan sandi mono alfabetik lainnya adalah sandi ini tidak begitu rentan terhadap metode pemecahan sandi yang disebut analisis frekuensi.

Giovan Batista Belaso menjelaskan metode ini dalam buku *La cifra del. Sig. Giovan Batista Nelaso* (1553) dan disempurnakan oleh diplomat Perancis Blaise de Vigenere pada tahun 1586. Pada abad ke19 banyak orang yang mengira vigenere adalah penemu sandi ini, sehingga sandi ini dikenal sebagai "sandi Vigenere". Sandi ini dikenal dengan luas karena cara kerjanya mudah dimengerti dan dijalankan dan bagi para pemula sulit dipecahkan.

Pada saat kejayaannya, sandi ini dijuluki *le chiffre indenchiffable* (bahasa perancis: "sandi yang tak terpecahkan"). Metode pemecahan sandi ini baru ditemukan pada abad ke19. Pada tahun 1854, Charles Babbage menemukan cara untuk memecahkan sandi vigenere. Metode ini dinamakan tes Kasiski karena Friedrich Kasiskilah yang pertama mempublikasikannya. Kunci pada kriptografi Vigenere adalah sebuah kata bukan sebuah huruf. Kata kunci ini akan dibuat berulang sepanjang plaintext, sehingga jumlah huruf pada kunci akan sama dengan jumlah huruf pada plaintext. Pergeseran setiap huruf pada plaintext akan ditentukan oleh huruf pada kunci yang mempunyai posisi yang sama dengan huruf pada plaintext.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2.3Bujur Sangkar Vigenere Cipher

Algoritma enkripsi vigenere cipher :

$$C_i = (P_i + K_i) \bmod 26$$

Algoritma dekripsi vigenere cipher :

$$P_i = (C_i - K_i) \bmod 26$$

Dimana :

C_i = nilai desimal karakter *ciphertext* ke-i

P_i = nilai desimal karakter *plaintext* ke-i

K_i = nilai desimal karakter kunci ke-i.

Sebagai contoh, jika *plaintext* adalah THEBEAUTYANDTHEBEAST dan kunci adalah ABC maka proses enkripsi yang terjadi adalah sebagai berikut :

Plaintext : THEBEAUTYANDTHEBEAST
 Kunci : ABCABCABCABCABCABCAB
Chipertext : TIGBFCUUAOFTIGBFCSU

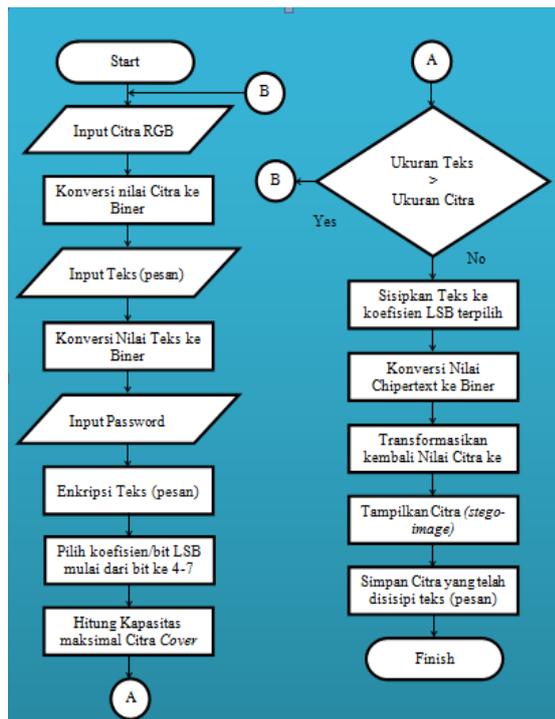
Pada contoh di atas kata kunci ABC diulang sedemikian rupa hingga panjang kunci sama dengan panjang plainteksnya. Kemudian setelah panjang kunci sama dengan panjang plainteks, proses enkripsi dilakukan dengan melakukan menggeser setiap huruf pada plainteks sesuai dengan huruf kunci yang bersesuaian dengan huruf plainteks tersebut.

Pada contoh di atas plainteks huruf pertama adalah T akan dilakukan pergeseran huruf dengan kunci $K_i=0$ (kunci huruf pertama adalah A yang memiliki $K_i=0$) menjadi T. Huruf kedua pada plainteks adalah H akan dilakukan pergeseran huruf dengan kunci $K_i=1$ (kunci huruf kedua adalah B yang memiliki $K_i=1$) menjadi I. Begitu seterusnya dilakukan pergeseran sesuai dengan kunci pada tiap huruf hingga semua plainteks telah terenkripsi menjadi ciphertext.

3. Hasil dan Analisa

3.1 Diagram alir program penyisipan teks ke dalam gambar

Diagram alir penyisipan teks ke dalam gambar dengan menggunakan Teknik *Least Significant Bit* pada perangkat lunak steganografi dapat dijabarkan dalam diagram alir pada Gambar 3.2.



Gambar 3.2 Flowchart penyisipan (*enkripsi*) teks ke dalam gambar

Pada proses penyisipan (*enkripsi*) teks ke dalam gambar, gambar yang telah dipilih, nilai pixel citra (*cover-image*) akan dikonversi ke biner (8-bit). Setelah itu memasukan atau memilih *file* teks (*plaintext*) yang akan disisipkan dan dilanjutkan dengan membangkitkan *pseudo-number* Vigenere Cipher. Setelah proses *embedding* selesai maka sistem akan menghitung daya tampung maksimal citra, jika *plaintext* melebihi kapasitas kemampuan dari *cover-image* maka sistem akan menolak untuk melanjutkan dan meminta untuk mengurangi jumlah pesan. Jika *cover-image* mampu menampung maka proses melanjutkan mengkonversi nilai biner *chipertext* (*stego-image*) ke nilai desimal dan dilanjutkan konversi nilai desimal ke nilai pixel dan akhirnya menyimpan (*save*) *file cover-image* yang sudah disisipi pesan (*chipertext*).

3.2 Diagram alir program ekstraksi teks yang terdapat pada gambar

Diagram alir pengestrakan teks yang ada pada gambar dapat dijelaskan pada Gambar 3.3.



Gambar 3.3 Flowchart untuk proses pengestrakan teks yang ada dalam gambar

Pada proses ekstraksi (*dekrpsi*) teks yang terkandung di dalam *stego-image* dipilih dan nilai pixel citra (*stego-image*) akan dikonversi ke biner (8-bit) dan dilanjutkan dengan membangkitkan *pseudo-number* Vigenere Cipher. selanjutnya mengambil nilai bit terakhir (LSB) di tiap pixelnya, dan nilai biner dikonversi dari biner ke desimal dan pesan (*plain-text*) ditampilkan.

Aplikasi “Steganografi LSB dengan Enkripsi Vigenere Cipher Pada Citra Jpeg” ini dibuat dengan menggunakan bahasa pemrograman Visual Basic 6.0. Tahap implementasi merupakan tahap yang akan membangun sebuah sistem berdasarkan atas analisis kebutuhan sistem yang telah dirancang sehingga akan dihasilkan sistem yang dapat menghasilkan tujuan yang akan dicapai.

Sebelum program diterapkan dan diimplementasikan, maka program harus *free error* (bebas kesalahan). Kesalahan program yang mungkin terjadi antara lain kesalahan penulisan bahasa, kesalahan waktu proses, atau kesalahan logikal. Setelah program bebas dari kesalahan, program di tes dengan memasukkan data yang akan diolah.

Sebagai contoh pengujian sebagai berikut :

Nilai Citra Digital True Color 24-Bit RGB yang mengandung pesan (*stego-image*).

(100, 121, 80)	(90, 75, 191)	(101, 80, 80)
(81, 180, 34)	(81, 122, 200)	(84, 120, 101)
(100, 121, 120)	(120, 122, 200)	(100, 123, 200)
(100, 121, 81)	(91, 74, 190)	(101, 80, 81)
(80 180, 34)	(81, 122, 201)	(84, 120, 100)
(101, 121, 81)	(120, 122, 200)	(100, 122, 200)

Konversi nilai piksel berkas citra stego-image ke biner :

(01100100,01111001,01010000)	(01011010,01001011,10111111)	(01100101,01010000,01010000)
(01010001,10110100,00100010)	(01010001,01111110,10,11001000)	(01010100,01111000,01100101)
(01100100,01111001,01010000)	(01010000,01111110,10,11001000)	(01100100,01111011,11001000)
(01100100,01111001,01010001)	(01011011,01001010,10,10111110)	(01100101,01010000,01010001)
(01010000,10110100,00100010)	(01010001,01111110,10,11001001)	(01010100,01111000,01100100)
(01100101,01111001,01010001)	(01010000,01111110,10,11001000)	(01100100,01111010,11001000)

Enkripsi Pesan.

Plaintext : "ATTACK"
Kunci : "NOW"

Nilai Citra Digital True Color 24-Bit RGB (*Cover-image*)

(100,121,80)	(90,74,190)	(100,81,80)
(81,180,34)	(80,122,201)	(84,120,100)
(100,120,80)	(80,122,200)	(80,122,200)
(100,121,80)	(90,74,190)	(100,81,80)
(81,180,34)	(80,122,201)	(84,120,100)
(100,120,80)	(80,122,200)	(80,122,200)

Enkripsi Vigenere Cipher :

Plaintext	A	T	T	A	C	K
Kunci	N	O	W	N	O	W
Ciphertext	N	H	P	N	Q	G

Konversi nilai piksel berkas citra digital 24-Bit RGB ke biner :

(01100100,01111001,01010000)	(01011010,01001011,10111111)	(01100101,01010000,01010000)
(01010001,10110100,00100010)	(01010001,01111110,10,11001000)	(01010100,01111000,01100101)
(01100100,01111001,01010000)	(01010000,01111110,10,11001000)	(01100100,01111011,11001000)
(01100100,01111001,01010001)	(01011011,01001010,10,10111110)	(01100101,01010000,01010001)
(01010000,10110100,00100010)	(01010001,01111110,10,11001001)	(01010100,01111000,01100100)
(01100101,01111001,01010001)	(01010000,01111110,10,11001000)	(01100100,01111010,11001000)

Konversi biner ke nilai piksel citra : Nilai Citra Digital 24-Bit RGB yang sudah disisipi pesan.

(100, 121, 80)	(90, 75, 191)	(101, 80, 80)
(81, 180, 34)	(81, 122, 200)	(84, 120, 101)
(100, 121, 120)	(120, 122, 200)	(100, 123, 200)
(100, 121, 81)	(91, 74, 190)	(101, 80, 81)
(80 180, 34)	(81, 122, 201)	(84, 120, 100)
(101, 121, 81)	(120, 122, 200)	(100, 122, 200)

Konversi ciphertext ke biner :

- ❖ Nilai ASCII huruf "N" : 78, Biner : 01001110
- ❖ Nilai ASCII huruf "H" : 72, Biner : 01001000
- ❖ Nilai ASCII huruf "P" : 80, Biner : 01010000
- ❖ Nilai ASCII huruf "N" : 78, Biner : 01001110
- ❖ Nilai ASCII huruf "Q" : 81, Biner : 01010001
- ❖ Nilai ASCII huruf "G" : 71, Biner : 01000111

DekripsiPesan

Cipherteks : "NHPNQG"
Kunci : "NOW"

Ambil nilai bit terakhir tiap byte (LSB) dan susun menjadi tiap 8-bit :

010011100100100001010000
010011100101000101000111

Konversi Nilai Biner Ciphertext ke Desimal dan Alfabet :

01001110	01001000	01010000	01001110	01010001	01000111
78	72	80	78	81	71
N	H	P	N	Q	G

Hasil dari Dekripsi Vigenere Cipher :

Ciphertext	N	H	P	N	Q	G
Kunci	N	O	W	N	O	W
Plaintext	A	T	T	A	C	K

Gambar 4.2 Hasil plaintext dengan kunci yang benar.

Pesan *Error* / Tidak bisa dibaca.

Berikut hasil jika penerima salah memasukkan password:

Cipherteks : "NHPNQG"

Kunci : "WON"

Plainteks : "8T]8CT"

5. Kesimpulan

Kesimpulan yang dapat diambil dari penulisan laporan tugas akhir ini adalah sebagai berikut :

1. Semakin banyak karakter yang disisipkan maka semakin berkurang kualitas citra yang dihasilkan. Hal ini ditandai dengan berkurangnya nilai PSNR yang dihasilkan oleh masing-masing file citra uji, dimana dari uji coba pengujian penyisipan karakter dengan jumlah karakter yang berbeda-beda, diperoleh hasil PSNR yang semakin berkurang sesuai dengan banyaknya karakter yang disisipkan, seperti pada citra uji A.jpg dimana pada penyisipan 100 karakter diperoleh nilai PSNR 78,09 dB dan nilai PSNR yang semakin menurun sesuai dengan banyaknya karakter yang disisipkan sampai dengan 71,14 dB pada penyisipan 500 karakter.
2. Pada proses ekstraksi, pesan atau informasi yang disisipkan pada file citra uji dalam aplikasi Steganografi ini, dapat diperoleh kembali secara utuh atau dengan kata lain pesan yang disisipkan sebelum proses penyisipan dan setelah proses ekstraksi sama tanpa ada perubahan kecuali pengguna memasukkan *password* yang salah atau jika sebelum proses ekstraksi citra yang telah disisipi pesan (*stegoimage*) telah mengalami perubahan kompresi citra.

Referensi

- [1] Menezes Alfred, Oorschot Paul Van and Vanston Sean, 1996. "HandBook of Applied Cryptography", CRC Press.
- [2] Ariyus, Dony, Kriptografi – Keamanan Data Dan Komunikasi, Graha Ilmu, Yogyakarta, 2006.
- [3] Triputra Safei, Timotius, Pengukuran dan Pengujian Kekuatan Algoritma Auto-key Vigenere Cipher, ITB, Bandung, 2012
- [4] Leonardo, Kevin Handoyo, Modifikasi Vigenère Cipher dengan Metode Penyisipan Kunci pada Plaintext, ITB, Bandung, 2012
- [5] Suranta, Ricardo Pramana, Perbandingan Ketahanan Algoritma LSB dan F5 dalam Steganografi Citra, ITB, Bandung, 2012
- [6] B.Tjaru, Setia Negara, Modifikasi Full Vigenere Chipher dengan Pengacakan Susunan Huruf pada Bujur Sangkar Berdasarkan Kunci, ITB, Bandung, 2012
- [7] Alatas, Putri, Implementasi Teknik Steganografi dengan Metode LSB pada Citra Digital, Universitas Gunadarma, Jakarta, 2009
- [8] Suhartana, I Ketut Gede, Pengamanan Image True Color 24 Bit Menggunakan Algoritma Vigenere Cipher Dengan menggunakan Kunci Bersama, Universitas Udayana, Bali.
- [9] Ramadhani, Budi, Steganografi pada Citra GIF menggunakan bahasa pemrograman Delphi, UII, Yogyakarta, 2006
- [10] www.cctv-information.co.uk/constant2/sn_ratio.html