PENGEMBANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN PROGRAM DELPHI

Supiyanto[1], **Samuel A. Mandowen**[2]
[1,2] Prodi Sistem Informasi Jurusan Matematika FMIPA Universitas Cenderawasih, Jayapura

ABSTRACT

Security of data transmission that contains important messages and secrets become a top priority. Therefore, required an appropriate data security to guarantee confidentiality. Data security or secret messages that can be done in a way to hide the secret data into multimedia data. One of the security of data or messages in the container known as steganography.

This study aims to create an application program that can be used to hide a message to the media or container. Messages used in the form of text data, and the media are used as a carrier of a secret message (carrier files) such as image (images) in BMP format, while developing the application will use the Delphi programming language.

The method used in this research is the Least Significant Bit (LSB). Steganographic techniques with this method are done by modifying the bits belonging to bit LSB in each byte of color in a pixel of an image. An LSB bit is modified to be replaced by any existing LSB bits of the message you want to hide. After all the message bits replace bits LSB in the image file, then the message has been successfully hidden information.

Results from this study are the application program that can be used to hide messages in a container in the form of BMP image format and if seen with the eye, the image of the container before and after insertion of the message there is no difference.

Keywords : Image, LSB, Steganography, Bit, Least Significant Bit, Steganography.

PENDAHULUAN

Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis atau menggambar), sehingga steganografi bias diartikan sebagai "menulis tulisan yang tersembunyi atau terselubung".

Tujuan dari steganografi adalah untuk menyembunyikan pesan di dalam gambar sedemikian rupa yang tidak memungkinkan setiap "musuh" bahkan mendeteksi bahwa ada hadiah pesan rahasia dalam gambar. Steganografi mencoba untuk menyembunyikan keberadaan komunikasi.

Steganografi merupakan seni penyembunyian pesan ke dalam wadah (media digital) sedemikian rupa sehingga orang lain tidak menyadari ada suatu pesan di dalam wadah tersebut.

Steganografi memanfaatkan kelemahan indera manusia seperti indera pendengaran dan indera penglihatan. Dengan adanya kelemahan ini steganografi dapat diterapkan di berbagai media *digital*. Hasil keluaran *file* yang telah disisipi pesan mempunyai persepsi bentuk yang sama dengan *file* aslinya. Penggunaan komputer diperlukan untuk mengetahui keberadaan pesan yang tersembunyi dalam *file digital*.

Terdapat beberapa istilah yang berkaitan dengan steganografi:

1. *Hiddentext* atau *embedded message* : pesan yang disembunyikan.
2. *Covertext* atau *cover-object* : Media yang digunakan untuk menyembunyikan *embedded message.*
3. *Stegotext* atau *stego-object* : Media yang sudah berisi *embedded message.*

Dengan demikian pengamanan data dengan menggunakan steganografi membutuh-kan dua property : wadah penampung dan data rahasia (pesan) yang akan disembunyikan. Media

* *Alamat korespondensi* :
Kampus Uncen Waena, Jurusan Matematika, Program Studi Sistem Informasi, Jayapura
e-mail: supi6976@gmail.com

digital sebagai wadah penampung yang dapat digunakan, misalnya citra (gambar), suara (audio), teks, dan video. Sedangkan pesannya dapat berupa citra, suara, teks, atau video. Penyisipan pesan ke dalam media *covertext* dinamakan *encoding*, sedangkan ekstraksi pesan dari stegotext dinamakan *decoding*

Steganografi pada media digital file gambar sebagaimana yang akan digunakan pada penelitian ini dimaksudkan untuk mengeksploitasi keterbatasan kekuatan sistem penglihatan manusia dengan cara menurunkan kualitas warna pada file gambar yang belum disisipi pesan rahasia. Sehingga dengan keterbatasan tersebut manusia sulit menemukan gradasi penurunan kualitas warna pada file gambar yang telah disisipi pesan rahasia.

Penggunaan steganografi antara lain bertujuan untuk menyamarkan eksistensi (keberadaan) data rahasia sehingga sulit dideteksi, dan melindungi hak cipta suatu produk. Steganografi dapat dipandang sebagai kelanjutan kriptography. Jika pada kriptography, data yang telah disandikan (*ciphertext*) tetap tersedia, maka dengan steganografi ciphertext dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Data rahasia yang disembunyikan harapannnya dapat diekstraksi kembali persis sama seperti keadaan aslinya.

METODE PENELITIAN

Salah satu algoritma steganografi yang paling popular dan sering digunakan untuk menyembunyikan informasi dalam citra digital metode penyisipan adalah *Least Significant Bit* (LSB). LSB adalah algoritma sederhana yang menukar bit yang paling kecil ke dalam beberapa byte media penyembunyian secara berurutan.
Seperti diketahui untuk *file bitmap* 24 bit maka setiap *pixel* (titik) pada gambar tersebut terdiri dari susunan tiga warna yaitu merah, hijau, dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (*byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111.

Bit (*binary digit*) adalah unit dasar penyimpanan data di dalam komputer, nilai bit suatu data adalah 0 atau 1. Semua data yang ada pada komputer disimpan ke dalam satuan bit ini, termasuk gambar, suara, ataupun video. Jenis-jenis format pewarnaan di dalam media gambar, seperti grayscale, RGB, dan CMY. Sebagai contoh pewarnaan monochrome bitmap (menggunakan 1 bit untuk tiap pixelnya), RGB - 24 bit (8 bit untuk Red, 8 bit untuk Green, dan 8 bit untuk Blue), Grayscale-8 bit (menentukan tingkat keabuan suatu pixel berdasarkan nilai bitnya). Misalnya gambar di bawah ini :

Gambar 1. citra lena dengan tingkat keabuan 8 bit

Gambar di atas menggunakan format pewarnaan grayscale, artinya tiap pixel dari gambar ini direpresentasikan dengan nilai sepanjang 8 bit.
sedangkan sebuah data berupa text "**PINTAR**", kalau direpresentasikan ke dalam binary menjadi seperti pada Tabel 1.

Tabel 1. Tabel representasi kata "pintar"

Karakter	Nilai ASCII	Binary
P	112	01110000
I	105	01101001
N	110	01101110
T	116	01110100
A	97	01100001
R	114	01110010

Misalnya data binary dari citra sebagai berikut :

Tabel 2. Data citra dalam bentuk biner

00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011

Sesuai dengan metodeya, LSB artinya bit yang tidak significant/tidak mempunyai pengaruh yang besar, maka metode ini mengganti nilai bit ke-8. Sehingga, bila kita hendak menyisipkan kata pintar pada citra dengan metode *Least Significant Bit* dapat dilakukan dengan cara mengganti bit terakhir dari citra dengan bit dari

pesan yang kita punya. Sehinga hasilnya akan tampak seperti tabel 3 berikut :

Tabel 3. Tabel Hasil Berkas Stego

00000000	00000001	00000001	00000001	00000000	00000000	00000000	00000000
00000000	00000001	00000001	00000000	00000001	00000000	00000000	00000001
00000000	00000001	00000001	00000000	00000001	00000001	00000001	00000010
00000000	00000001	00000011	00000011	00000010	00000011	00000010	00000010
00000000	00000001	00000011	00000010	00000010	00000010	00000010	00000011
00000000	00000001	00000011	00000011	00000010	00000010	00000011	00000010

Dari Tabel 3, terlihat bahwa hanya beberapa bit rendah saja yang berubah (cetak tebal), untuk penglihatan mata manusia sangatlah mustahil untuk dapat membedakan warna pada file gambar yang sudah diisi pesan rahasia jika dibandingkan dengan file gambar asli sebelum disisipi dengan pesan rahasia.

Untuk memperkuat penyembunyian data, bit-bit data tidak digunakan untuk mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49.

HASIL DAN PEMBAHASAN

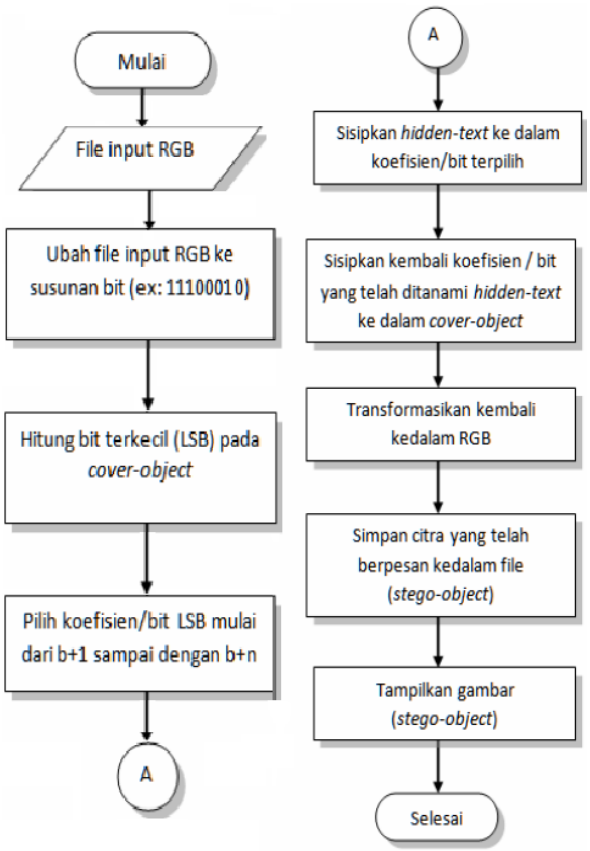
Algoritma

Secara garis besar jalannya aplikasi ini adalah terbagi dua proses utama yaitu hide message atau penyisipan pesan dan extract message atau pendekteksian kembali pesan yang tersembunyi.

Pada proses penyisipan pesan (*embedding message*) dimulai dengan memilih gambar yang akan dijadikan cover object untuk menyisipkan dan menyembunyikan pesan ke dalam gambar. kemudian menuliskan isi pesan text yang akan disisipkan kedalam gambar. Sedangkan pada proses pendeteksian pesan (*extraction message*) dimulai dengan memilih file gambar atau covert object yang akan akan di extract. Kemudian gambar tersebut di extract untuk mendapatkan pesan yang telah disisipkan.

Berikut merupakan digram alir atau flowchart yang akan menjelaskan proses embedding message yaitu bagaimana suatu file gambar dapat disisipkan pesan sehingga menghasilkan stego object atau encoder dan proses extraction message yaitu bagaimana mengekstrak pesan dari suatu file gambar stego object agar dapat terbaca kembali pesan yang dienkripsi sebelumnya.

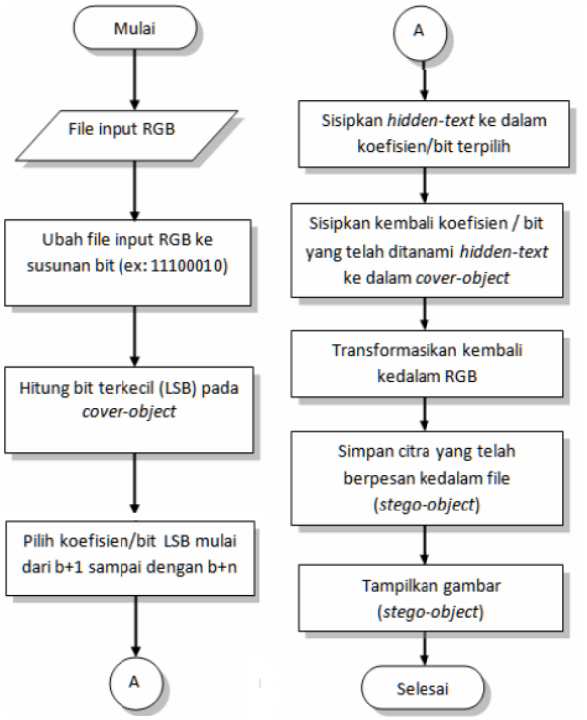
Berikut merupakan digram alir atau flowchart yang akan menjelaskan proses embedding message yaitu bagaimana suatu file gambar dapat disisipkan pesan sehingga menghasilkan stego object atau encoder dan proses extraction message yaitu bagaimana mengekstrak pesan dari suatu file gambar stego object agar dapat terbaca kembali pesan yang dienkripsi sebelumnya.



Gambar 2. Algoritma Penyisipan pesan

Pada gambar 2. diatas adalah flowchart proses embedding message kedalam file citra (cover-object) dimulai dengan membaca file citra ke RGB, Seperti kita ketahui untuk file bitmap 24 bit maka setiap pixel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Setelah membaca pixel dari file citra langkah selanjutnya menentukan bit terkecil (LSB) pada cover-object. Setelah menentukan

bit terkecil dari cover-object yang akan digunakan. Proses selanjutnya adalah memilih koefisien bit terpilih mulai dari b+1 sampai b+n untuk disisipkan hidden-text kedalamnya, Selanjutnya adalah setelah memilih koefisien atau bit-bit terpilih maka proses berikutnya adalah menyisipkan hidden-text ke dalamnya koefisien atau bit-bit tersebut sehingga akan dihasilkan koefisien atau bit-bit yang baru yang telah mengandung pesan, dan menyisipkannya kembali kedalam cover-object, yang kemudian koefisien tersebut selanjutnya akan di transformasikan kembali kedalam nilai RGB yang baru dan menyimpan citra yang telah berpesan ke dalam cover-object sehingga diperoleh atau dapat ditampilkan sebuah gambar baru yang telah disisipkan pesan atau stego-object.



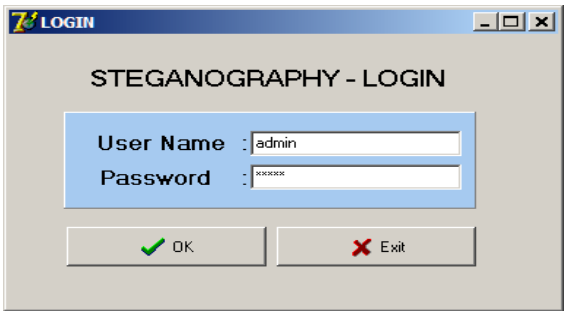
Gambar 3. Algoritma Ekstraksi Pesan

Pada gambar 3 di atas adalah flowchart proses extraction message dari stego-object menghasilkan hidden-text yang terdapat didalamnya atau untuk mengungkap kembali pesan yang disisipkan kedalam file citra, proses awalnya dimulai dengan membaca file citra ke RGB, dan mengubah file input RGB kedalam format biner. Setelah diperoleh byte yang tersembunyi pada cover-object maka proses berikutnya adalah mengekstrak kembali pesan yang tersembunyi (hidden-text) yang terdapat didalamnya sehingga pesan dapat ditampilkan kembali.

Program Aplikasi

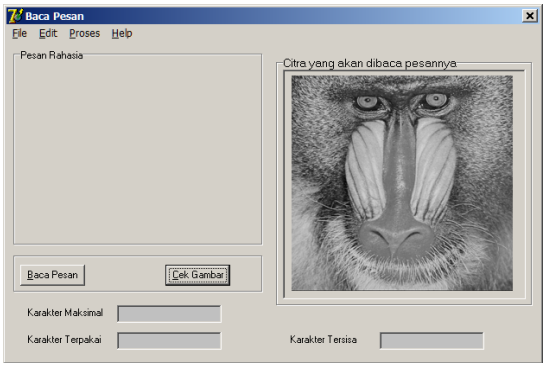
Aplikasi "aplikasi pengamanan data" ini dibuat dengan menggunakan bahasa pemrograman Delphi 7. Tahap implementasi merupakan tahap yang akan membangun sebuah sistem berdasarkan atas analisis kebutuhan sistem yang telah dirancang sehingga akan dihasilkan sistem yang dapat menghasilkan tujuan yang akan 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.

Tampilan program aplikasi pengamanan data yaitu sebagai berikut:



Gambar 4. Tampilan Menu Login

Gambar 5. menu untuk penyisipan pesan.



Gambar 6. menu untuk pembacaan pesan

Analisis Hasil

Setelah melakukan proses penyisipan dan eksraksi file image, maka untuk melihat apakah hasil dari penyisipan dan ekstraksi file image telah berhasil maka akan disbandingkan citra sebelum dan sesudah disisipi pesan baik dari segi size, perubahan warna dan histogram.

a. Ditinjau dari size citra

Analisis ini dilakukan bertujuan untuk mengetahui apakah *file image* dapat menampung pesan tanpa adanya perubahan ukuran pada *file image*. Kemudian apakah *file* pesan tersebut dapat diambil kembali seperti semula tanpa adanya perubahan.

Tabel 3. Tabel uji penyembunyian pesan

Original Image	Size image	pesan	Size image	Image Output	Size image
artis-2.bmp	149 kb	Alamat.txt	5 kb	artis-2 hasil.bmp	149 kb
Baboon256.bmp	193 kb	Alamat.txt	5 kb	Baboon256 hasil.bmp	193 kb
bunga 320.bmp	301 kb	pesan.txt	45 kb	Bunga 320 hasil.bmp	301 kb

Table 3.1 menunjukkan bahwa *file* citra dapat menampung *file* pesan tanpa adanya perubahan ukuran pada *file* citra, dan *file* pesan yang telah disisipi ke dalam *file* citra dapat diambil kembali tanpa adanya perubahan pada size *file* pesan.

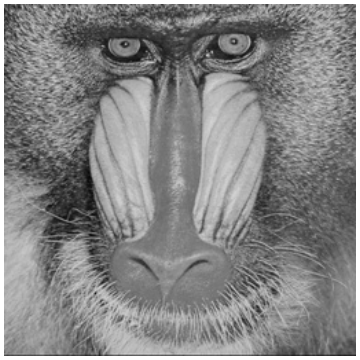
b. Ditinjau dari perubahan citra

Analisis ini dilakukan bertujuan untuk mengetahui apakah citra setelah dan sebelum disisipi pesan mengalami perubahan bila ditinjau dari segi tampilannya. Untuk mengetahui hal ini, berikut ini diberikan empat citra yakni dua buah citra sebelum disisipkan pesan (artis-2 dan baboon256) dan dua buah citra setelah disisipkan pesan (artis-2 hasil dan baboon256 hasil).

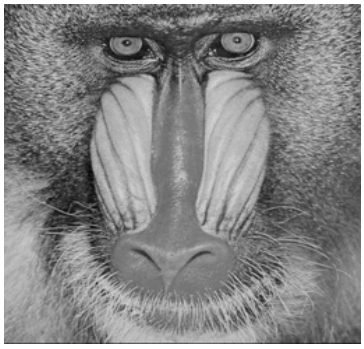
Gambar 7, 8, 9 dan 10. menunjukkan bahwa citra sebelum dan sesudah disisipi pesan terilhat sama atau tidak mengalami perubahan. Ini dikarenakan *file* pesan yang disisipkan mengalami proses steganografi yang menggunakan metode LSB (Least Significant Bit), dimana file pesan disisipkan pada bit terakhir pada *file image*.

Gambar 7. citra artis-2, citra sebelum disisipkan File Pesan

Gambar 8. Citra artis-2 hasil, citra sesudah disisipkan File Pesan



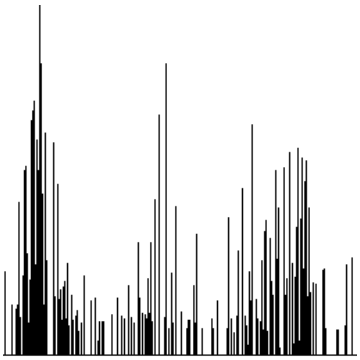
Gambar 9. Citra baboon256, citra Sebelum disisipkan File Pesan



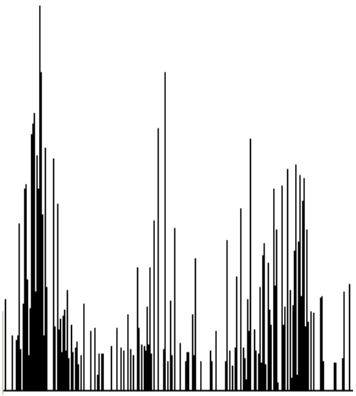
Gambar 10. Citra baboon256 hasil, citra sesudah disisipkan File Pesan

c. Ditinjau dari Histogram citra

Analisis ini dilakukan bertujuan untuk mengetahui apakah citra setelah dan sebelum disisipi pesan mengalami perubahan bila ditinjau dari histogram. Untuk mendukung hal ini, berikut ini diberikan salah contoh histogram dari citra yang sebelum dan setelah disisapi pesan.



Gambar 11. Histogram Intensitas Warna Sebelum Penyisipan Pesan



Gambar 12. Histogram Intensitas Warna setelah Penyisipan Pesan

Gambar 11 dan 12 menunjukkan bahwa Histogram sebelum dan sesudah disisipi pesan terilhat sama atau tidak mengalami perubahan.

KESIMPULAN

A. Kesimpulan

Dari hasil pengujian sistem yang dilakukan pada bab sebelumnya, maka dapat disimpulkan ke dalam beberapa hal antara lain:

1. Aplikasi pengamanan data dengan mengimplementasikan teknik steganografi yang menggunakan algoritma LSB dalam mengamankan pesan menggunakan program Delphi berhasil dibuat dengan baik. Hal ini dibuktikan melalui hasil uji analisa pada Tabel 3. bahwa pesan dapat disisipkan ke dalam citra dan pesan dapat diambil kembali atau dibaca dari citra tersebut.
2. Berdasarkan hasil uji analisis, penyisipan pesan ke dalam citra tidak mempengaruhi kualitas warna pada citra tersebut.
3. Implementasi algoritma LSB (*Least Significant Bit*) dapat digunakan cukup baik untuk menyembunyikan pesan di dalam pesan sebuah berkas citra digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut.

B. Saran

1. Aplikasi pengamanan seperti ini dapat diimplementasikan di instansi yang membutuhkan pengamanan *file* seperti bank, kantor pemerintahan, militer dan sebagainya.
2. Aplikasi ini hanya menggunakan citra atau gambar sebagai media penampung, diharapkan dapat dikembangkan sehingga dapat menggunakan *file* teks, audio, video dan lain-lain sebagai media penampungnya.
3. Aplikasi ini masih dikembangkan untuk perangkat *computer desktop*. Akan lebih praktis apabila dapat dikembangkan lebih lanjut untuk dapat digunakan dalam lingkungan perangkat keras *mobile* seperti telepon genggam.

DAFTAR PUSTAKA

Achmad, B., dan Kartika Firdausy, *Teknik Pengolahan Citra Digital Menggunakan Delphi*, Ardi Publishing, Cetakan pertama, Yogyakarta, 2005

Ahmad, Usman., *Pengolahan Citra Digital*, Graha Ilmu, Cetakan pertama, Yogyakarta, 2005

Alfatwa, F. D. 2005. *Watermarking Pada Citra Digital Menggunakan Discrete Wavelet Transform*. Bandung: Institut Teknologi Bandung.

Hakim A, Muhammad, *Makalah Studi dan Implementasi Steganografi Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah*, Teknik Informatika ITB, 2007

http://haryanto.staff.gunadarma.ac.id/Downloads/files/7272/9.Steganografi.ppt

Ibrahim, R. N., *Keamanan citra digital dengan Menggunakan metode discrete wavelet Transformation,* Jurnal Computech & Bisnis, Vol. 6, No. 2, Desember 2012, 82-95

Johnson, Neil F.; Duric, Zoran; Jajodia, Shushil: *"Information Hiding Steganography and Watermaking-Attacks and Countermeasures"*, Advanced in Information Security, Kluwer Academic Publisher, United State, 2001.

Maya, *Steganografi LSB* (http://maya9luthu.-blogsome.com/2006/12/11/steganografi-lsb/, diakses 28 Mei 2013 pukul 10.40

Munir, Rinaldi, 2006. *Kriptografi Steganografi dan Watermarking,* Bandung : Institut Teknologi Bandung.

Neil F. Johnson, Sushil Jajodia, "*Steganography: Seeing the Unseen"*.

N.F. Johnson, Z. Duric, and S. Jajodia, "Information hiding: Steganography and watermarking - attacks and countermeasures," Kluwer Academic Publishers, 2000

Sukmawardhani, D.,Y. *Soepriyanto dan Taufik R. Penyembunyian File Pada Citra Digital Dengan Menggunakan Metode Modifikasi 2 Bit*., 2009.

Tjong, Andreas, "Steganografi : LSB (Least Significant Bit)", http://andreastjong_.wordpress.com/2008/09/22/steganografi-2-lsb-least-significant-bit/Tanggal akses : 28 Mei 2013 pukul 10.06

Warsito, A. Budi, L. Fajarita dan Nazori A.Z., *Proteksi Keamanan Dokumen Sertifikat File Jpeg pada Perguruan Tinggi dengan Menggunakan Steganografi dan Kriptografi*, Jurnal TELEMATIKA MKOM Vol.4 No.1, Maret 2012.