STEGANOGRAFI PADA CITRA BMP 24-BIT MENGGUNAKAN METODE LEAST SIGNIFICANT BIT

DAVID[1], A. MURTADO[2], UTIN KASMA[3]

[1,2] Program Studi Teknik Informatika,
Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jln. Merdeka No 372 Pontianak, Kalimantan Barat
E-mail[1]: David_Liauw@yahoo.com dan DavidLiauw@gmail.com
E-mail[2]: Amurtado.Eresha@yahoo.com

[3]Program Studi Sistem Informasi,
Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jln. Merdeka No 372 Pontianak, Kalimantan Barat
E-mail: utin.kasma@yahoo.co.id

Abstract : *Steganography is the art and science of hiding messages in a message. In the world of modern technology, the secret messages can be hidden behind the image (image), for example. Messages can be encoded in the low-order bits so as not to interfere with media images (image) is displayed. Research Application System Method of Least Significant Bit Steganography aims to provide data security facilities, particularly in terms of hiding data that is intended for corporate, business or personal. The study was based on increased traffic flow data packet delivery via email or other media that directly impact the growing threats and data theft against itself. The research was shaped by the method of experimental study of literature and data collection techniques using literature and documentation. Approach to problem solving using problem identification approach to identify exactly what problems to look for a solution. Design method used is a Rapid Application Development by applying a 5-step design, namely business modeling, data modeling, process modeling, generation and application testing and turnover. The software performs the analysis by comparing the features and the existing criteria and is built using Borland Delphi 7.0 which make the process of concealment and recall data that is hidden as a solution for end-users in terms of the need for data security warnings.*

Keywords : Data Security, Steganography, Encryption, Decryption, Least Significant Bit (LSB)

1. PENDAHULUAN

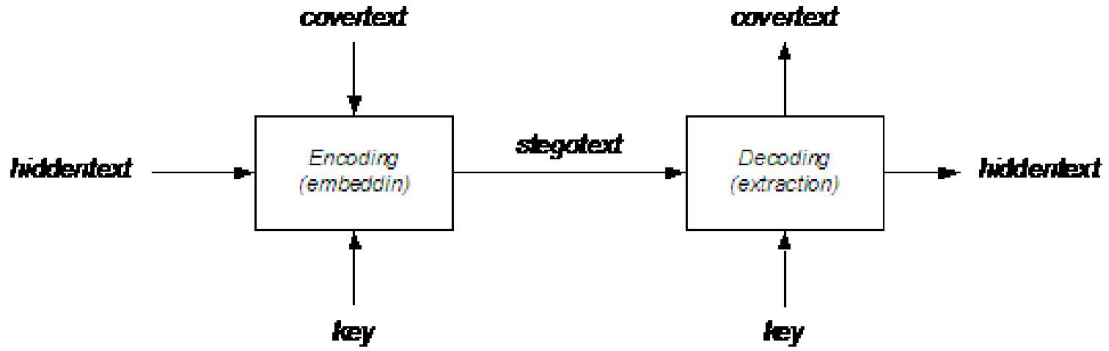
Salah satu teknik penyembunyian pesan yang efektif adalah dengan menggunakan steganografi. Steganografi sebagai suatu seni penyembunyian pesan banyak digunakan untuk mengirim pesan melalui jaringan Internet tanpa diketahui orang lain dengan memanfaatkan media digital. Metode yang digunakan untuk menyembunyikan pesan pada media digital tersebut berbeda-beda. Contohnya pada file citra pesan dapat disembunyikan dengan menggunakan metode penyisipan bit rendah (*least significant bit insertion*) pada data pixel yang menyusun file tersebut. Secara umum dengan metode ini hanya sebagian dari data bit rendah yang diubah, sehingga bila dibutuhkan dapat digunakan bit rendah kedua bahkan ketiga. Teknik di atas bisa diperkuat dengan cara mengubah pola penyembunyian data. Bit-bit data *embedded message* bias tidak disisipkan pada byte-byte *cover* secara berurutan, namun dipilih susunan byte secara acak. Untuk membuat susunan tersebut diperlukan sebuah pembangkit bilangan acak-semu yang disebut sebagai *pseudo-random number generator* (PRNG). Generator ini memerlukan sebuah umpan atau *seed* untuk mulai membangkitkan. *Seed* inilah yang berlaku sebagai kunci.

Masih banyak metode yang dapat digunakan dalam penerapan steganografi. Metode *Least Significant Bit* diambil karena dengan penyisipan bit rendah diharapkan tidak terjadi perubahan berarti pada *cover object* sehingga tidak akan berpengaruh pada penglihatan mata manusia. Selain itu juga tidak menimbulkan kecurigaan seperti halnya bila menggunakan kriptografi. Berbeda dengan kriptografi, steganografi menyembunyikan pesan rahasia agar orang awam tidak menyadari keberadaan dari pesan yang disembunyikan. Teknik ini sering digunakan untuk menghindari kecurigaan orang dan menghindari keinginan orang untuk mengetahui isi pesan rahasia tersebut. Walaupun steganografi dapat dikatakan mempunyai hubungan yang erat dengan kriptografi, tapi metode ini sangat berbeda dengan kriptografi. Kriptografi mengacak pesan sehingga tidak dimengerti, sedangkan steganografi bertujuan menyembunyikan pesan sehingga tidak terlihat. Dengan melihat hal tersebut, maka penulis mencoba untuk membuat suatu aplikasi perangkat lunak steganografi yang dapat diterapkan oleh suatu instansi, *home industry* maupun perorangan untuk kebutuhan keamanan data. Tujuan utama dari penelitian ini adalah menghasilkan perangkat lunak Steganografi pada citra BMP 24 bit menggunakan metode *Least Significant Bit*.

2. TINJAUAN PUSTAKA

Steganografi adalah seni dan ilmu menyembunyikan informasi dengan embedding pesan dalam suatu media digital, pesan yang tampaknya tidak berbahaya. Steganografi bekerja dengan mengganti bit tidak berguna atau tidak digunakan, dimana media digital berupa file (seperti grafik, suara, teks, HTML) dengan potongan yang berbeda. Menurut Arubusman (2007), "Steganografi adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya." Menurut Munir (2004), "Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia (hiding message) sedemikian sehingga keberadaan (eksistensi) pesan yang tidak terdeteksi oleh indera manusia".

Skema penyembunyian data dalam steganografi secara umum adalah data atau informasi yang ingin disembunyikan disimpan dalam sebuah wadah (*cover*) melalui suatu algoritma steganografi tertentu (misalnya *Least Significant Bit*). Untuk menambah tingkat keamanan data, dapat diberikan kunci, agar tidak semua orang mampu mengungkapkan data yang disimpan dalam berkas wadah (cover). Hasil akhir dari proses penyimpanan data ini adalah sebuah berkas stego (stego text). Adapun properti steganografi adalah : 1) Embedded message (hiddentext) : pesan yang disembunyikan; 2) Cover-object (covertext) : pesan yang digunakan untuk menyembunyikan embedded message; 3) Stego-object (stegotext) : pesan yang sudah berisi pesan embedded message; dan 4) Stego-key : kunci yang digunakan untuk menyisipan pesan dan mengekstraksi pesan dari stegotext.



Gambar 1. Skema Embedding dan Extraction Data (Munir, 2009)

Pada file citra pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (LSB) pada data pixel yang menyusun file tersebut. Untuk file bitmap 24 bit, 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. Sebagai contoh file gambar BMP 24 bit dengan warna merah murni dalam format biner akan terlihat sebagai berikut :

00000000 00000000 11111111
00000000 00000000 11111111

Sedangkan untuk warna hijau murni dalam format biner akan terlihat sebagai berikut:

00000000 11111111 00000000
00000000 11111111 00000000

Sedangkan untuk warna biru murni dalam format biner akan terlihat sebagai berikut:

11111111 00000000 00000000
11111111 00000000 00000000

Dari uraian di atas dapat dilihat bahwa informasi dari warna biru berada pada bit pertama sampai bit delapan, dan informasi warna hijau berada pada bit sembilan sampai dengan bit 16, sedangkan informasi warna merah berada pada bit 17 sampai dengan bit 24. Dengan demikian pada setiap pixel file bitmap 24 bit dapat disisipkan 3 bit data, dimana bila ingin menyisipi 1 buah karakter (8 bit) maka dibutuhkan 3 buah pixel, dengan catatan hanya menyisipkan 1 bit terendah pada tiap bytenya.

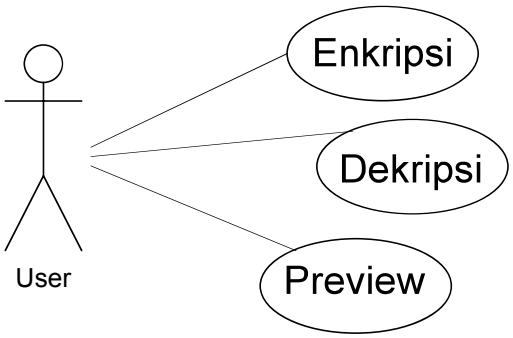
3. METODOLOGI PENELITIAN

Dalam penelitian ini, penulis menggunakan bentuk penelitian studi literatur dan metode penelitian eksperimen murni. Penulis melakukan kajian yang berkaitan erat dengan permasalahan yang hendak dipecahkan serta mendefinisikan masalah dengan melakukan eksperimen. Selain itu penulis juga mencari referensi dan informasi yang diperlukan dari buku-buku dan artikel-artikel di Internet. Referensi dan informasi tersebut merupakan dasar pembuatan aplikasi oleh penulis. Adapun instrumen atau alat (*tools)* yang digunakan penulis dalam penelitian yaitu menggunakan algoritma, *flowchart* (bagan alir). Penulis menggunakan metode perancangan RAD (*Rapid Application Development*) karena proses perkembangan perangkat lunak ini menekankan pada siklus perkembangan yang singkat dan pemanfaatan fungsi yang telah ada sebelumnya.

4. HASIL PENELITIAN

4.1 Perancangan Aplikasi

Metode yang digunakan untuk penyembunyian data adalah dengan cara memprioritaskan penyisipan ke dalam bit terendah (least significant bit) sesuai dengan nilai Bit Per Channel (BPC) yang diinput user. Media gambar yang digunakan adalah media gambar berformat bitmap (BMP) 24 bit sebagai sarana media penampung data (stego image), sedangkan data yang akan disembunyikan (plain data) dapat berupa sembarang data dengan ekstensi apa saja selama besar ukuran aktual data lebih kecil dari ukuran aktual bitmap. Pada perancangan aplikasi ini terdiri dari 3 bagian proses utama yaitu Enkripsi, Dekripsi dan Preview yang dapat dilihat pada diagram use case berikut:



Gambar 2. Use Case Diagram Aplikasi Steganografi

4.2 Proses Enkripsi

Proses penyisipan file dimulai dengan pemilihan nilai BPC oleh user, dilanjutkan dengan proses pemilihan data (*plain data*) dan berlanjut pada proses pengaksesan file bitmap (*cover image*). Sebelum proses sisip bit dimulai, dilakukan beberapa validasi dimana jenis bitmap harus dalam format 24 bit, kemudian dilanjutkan dengan pengecekan ukuran file yang harus lebih besar dari 0 kb dan berlanjut pada proses pengecekan ukuran file yang harus lebih kecil dari ukuran bitmap penampung. Setelah melalui proses validasi dan pengecekan di atas, proses sisip dimulai dengan menyisipkan bit BPC pada pixel pertama bitmap. Karena nilai biner BPC adalah dari 0000 (0) sampai 1000 (8), maka dibutuhkan 4 bit untuk menyisipkan nilai BPC. Dalam perangkat lunak steganografi ini, proses penyisipan bit pertama dan kedua BPC disisip pada channel biru pixel pertama, dilanjutkan dengan bit ketiga BPC disisip pada channel hijau pixel pertama dan penyisipian bit keempat BPC disisip pada channel merah pixel pertama. Setelah proses penyisipan BPC berhasil, maka dilanjutkan dengan penyisipan informasi ukuran file. Untuk menyisipkan informasi besar ukuran file, dialokasikan nilai statis sebesar 32 bit (LongInt [4 byte x 8]) dari bitmap. Dengan kata lain, telah dialokasikan pixel sekitar 10,6 pixel (32 bit) untuk menyisip informasi ukuran file. Setiap selesai menyisipkan 1 bit, dilakukan prosedur pengecekan pixel berikutnya (*CheckNextPixel*) guna mengetahui posisi index bit, index channel RGB dan posisi lainnya saat proses sisip berlangsung. Setelah proses penyisipan informasi ukuran file berhasil, dilanjutkan dengan penyisipan data yang sesungguhnya ke dalam bitmap. Proses penyisipan bit dilakukan secara sekuensial dalam sebuah file bitmap baru.

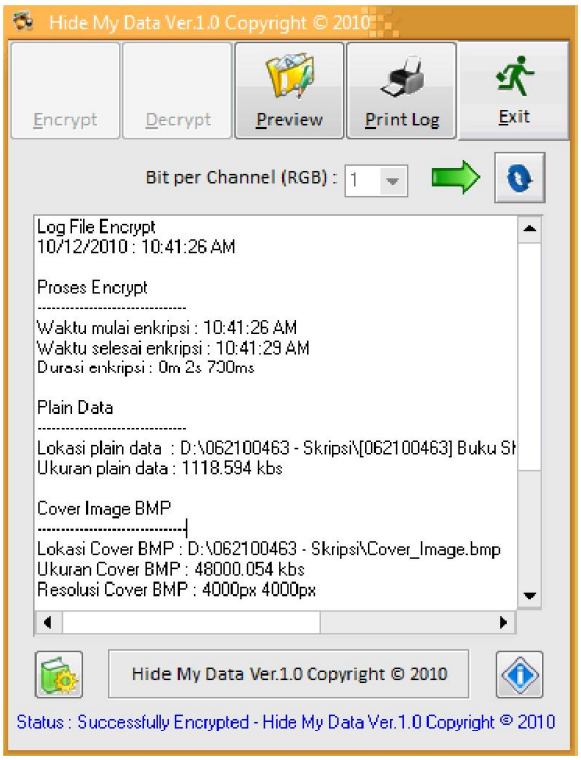
4.3 Proses Dekrip

Proses decrypt data dimulai dengan pengaksesan file gambar (*stego image*) yang telah disisipi, dilanjutkan dengan proses proses penyimpanan file data baru hasil ekstraksi. Sama halnya pada proses encrypt, sebelum proses ekstrak bit dimulai juga dilakukan beberapa validasi dimana file bitmap harus dalam format 24bit, kemudian dilakukan pengecekan kandungan nilai BPC (1 sampai 8) pada *stego image* dan berlanjut pada proses pengecekan data hasil ekstrak yang harus lebih besar dari 0 kb. Nilai BPC adalah kunci (*key*) yang dibutuhkan perangkat lunak untuk dapat mendekripsi bitmap, karena jumlah bit yang akan diambil per channel warna pixel oleh perangkat lunak menyesuaikan pada nilai BPC itu sendiri. Setelah proses pembacaan BPC selesai, maka dilanjutkan dengan proses pembacaan informasi ukuran file sesuai alokasi LongInt pada proses encrypt. Jadi program akan mengecek hingga sekitar 10,6 pixel untuk mendapatkan nilai informasi ukuran file yang disembunyikan. Setelah proses pembacaan informasi ukuran file selesai, maka dimulai proses pengambilan bit setiap channel RGB yang tentunya mengacu pada nilai BPC. Proses pembacaan

informasi ukuran file dilakukan agar perangkat lunak dapat menentukan batas akhir dari file gambar yang disisipi oleh file. Dengan demikian, proses pengambilan bit akan berhenti hingga mencapai nilai ukuran file.

4.4 Pengujian

Penulis membagi proses pengujian menjadi 2 bagian, yaitu pengujian dengan bitmap dan file berukuran kecil (sekitar 6 MB) serta bitmap dan file berukuran besar (sekitar 61 MB). Adapun data bitmap yang digunakan adalah sebagai berikut : 1) Cover_Image_24Bit_Small.bmp dengan resolusi 24 bit (1920 x 1080 pixel) dan ukuran sebesar 6.076 kilobytes; 2) Cover_Image_24Bit_Large.bmp dengan resolusi 24 bit (7000 x 3000 pixel) dan ukuran sebesar 61.524 kilobytes; dan 3) Red_Green_Blue_24Bit.bmp dengan resolusi 24 bit (900 x 1200 pixel) dan ukuran sebesar 3.186 kilobytes.



Gambar 3. Hasil Pengujian

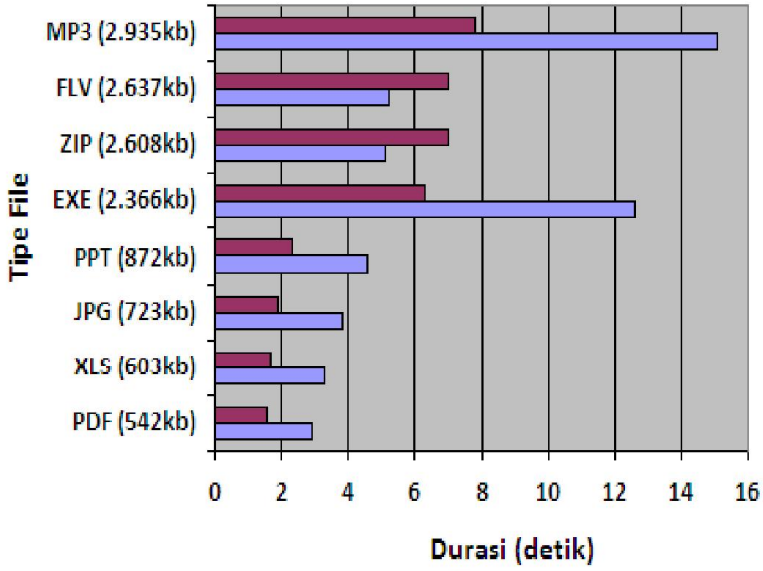
Adapun spesifikasi perangkat keras dari computer yang digunakan dalam pengujian program aplikasi ini antara lain: Prosessor Intel DualCore E5300 2,6 Ghz; Memori RAM 4 GB, Motherboard Asus P5KPL-AM, Kapasitas HardDisk 3 TB, VGA Ati HD4850 512MB, dan Sistem Operasi Windows 7 32bit. Berikut adalah tabel-tabel yang menunjukkan hasil pengujian proses sisip data dan ekstrak data dengan berbagai macam variasi pengujian.

Tabel 1
Hasil Pengujian File Kecil

Nama File	Ukuran (kbs)	File Bitmap	BPC	Sisip (m:s:ms)	Ektrak (m:s:ms)
Winzip 9.0.exe	2.366	Cover_Image_24Bit_Small.bmp	4	00:12:655	00:06:380
iPad Commercial.flv	2.637	Cover_Image_24Bit_Small.bmp	4	00:05:243	00:07:072
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	4	00:03:816	00:01:990

Sleeping Child.mp3	2.935	Cover_Image_24Bit_Small.bmp	4	00:15:182	00:07:872
Konsep WLAN.pdf	542	Cover_Image_24Bit_Small.bmp	4	00:02:920	00:01:573
Leadership.ppt	872	Cover_Image_24Bit_Small.bmp	4	00:04:648	00:02:386
Hdef Movie List.xls	603	Cover_Image_24Bit_Small.bmp	4	00:03:270	00:01:656
Ubuntu.zip	2.608	Cover_Image_24Bit_Small.bmp	4	00:05:189	00:07:001

Pada Tabel 1, aplikasi steganografi dapat melakukan proses enkripsi maupun dekripsi dalam waktu yang relatif singkat. Dari tabel di atas, rata-rata proses sisip untuk ukuran kecil berkisar antara 2,9 sampai 12,6 detik. Sedangkan proses ektrak untuk ukuran kecil berkisar antara 1,5 detik sampai 7,8 detik. Pada pengujian Tabel 1 digunakan nilai 4 pada BPC dan 8 tipe file berbeda, mulai dari file exe, flv, jpg, mp3, pdf, ppt, xls dan zip. Setelah proses ekstrak, semua data ekstrak tidak mengalami perubahan ukuran maupun integritas data.

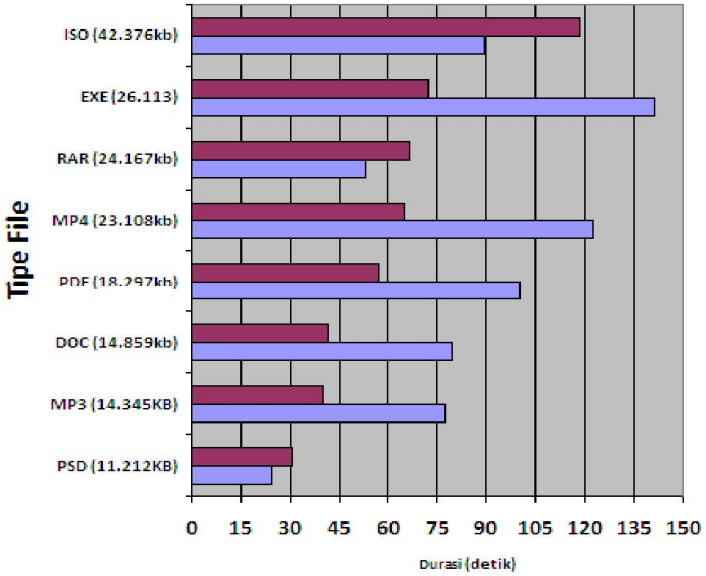


Gambar 4. Diagram Pengujian File Kecil (Enkripsi, Dekripsi)

Tabel 2
Hasil Pengujian File Besar

Nama File	Ukuran (kbs)	File Bitmap	BPC	Sisip (m:s:ms)	Ektrak (m:s:ms)
Pengantar TI.doc	14.859	Cover_Image_24Bit_Large.bmp	4	01:19:450	00:41:314
Adobe Reader 9.exe	26.113	Cover_Image_24Bit_Large.bmp	4	02:21:300	01:12:212
Kina Grannis.iso	42.376	Cover_Image_24Bit_Large.bmp	6	01:29:731	01:58:525
One Winged Angel.mp3	14.345	Cover_Image_24Bit_Large.bmp	4	01:17:635	00:39:809
Adrian Sanny Prewed.mp4	23.108	Cover_Image_24Bit_Large.bmp	4	02:02:841	01:05:126
Ps CS Type Effects.pdf	18.297	Cover_Image_24Bit_Large.bmp	4	01:40:401	00:57:092
SkyDrive Flyer.psd	11.212	Cover_Image_24Bit_Large.bmp	4	00:24:141	00:30:777
Volume 1.rar	24.167	Cover_Image_24Bit_Large.bmp	4	00:52:072	01:06:785

Pada Tabel 2, terjadi penurunan efisiensi durasi enkripsi maupun dekripsi bila dibanding dengan pengujian sebelumnya (file kecil). Tampak waktu sisip untuk file besar berkisar antara 52 detik hingga 2 menit 2 detik, sedangkan waktu ekstrak berkisar antara 30 detik hingga 1 menit 12 detik. Pengujian dilakukan dengan variasi nilai BPC 4 dan 6 dengan 8 variasi tipe file yaitu tipe file doc, exe, iso mp3, mp4, pdf, psd dan rar. Sama halnya pada pengujian sebelumnya, setelah proses ekstrak semua data hasil ekstraksi juga tidak mengalami perubahan berarti baik dari sisi ukuran file maupun integritas data.



Gambar 5. Diagram Pengujian File Besar (Enkripsi, Dekripsi)

Tabel 3
Hasil Pengujian Durasi Enkripsi Dengan Variasi BPC

Nama File	Ukuran (kbs)	File Bitmap	BPC	Sisip (m:s:ms)
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	1	00:03:858
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	2	00:03:815
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	3	00:03:863
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	4	00:03:851
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	5	00:03:848
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	6	00:03:861
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	7	00:03:845
AquaScape.jpg	723	Cover_Image_24Bit_Small.bmp	8	00:03:839

Pada Tabel 3, dilakukan pengujian lama waktu sisip dengan variasi nilai BPC mulai dari 1 sampai dengan 8. Tampak bahwa durasi enkripsi tercepat adalah 3 detik 815 milidetik dan waktu terlama adalah 3 detik 863 milidetik. Hasil pengujian dengan variasi nilai BPC hanya terpaut maksimal 48 milidetik, sehingga dapat ditarik

kesimpulan bahwa nilai BPC tidak berpengaruh signifikan pada efisiensi durasi enkripsi.

Tabel 4
Hasil Pengujian Stego Bitmap Dengan Variasi BPC

Nama File	Ukuran (kbs)	Cover Bitmap	BPC	Stego Bitmap
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	1	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	2	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	3	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	4	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	5	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	6	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	7	
Ball.jpg	300	Red_Green_Blue_24Bit.bmp	8	

Pada Tabel 4, tampak perubahan warna RGB pada Stego Bitmap bila dilakukan enkripsi dengan variasi BPC dari 1 hingga 8. Secara kasat mata, perubahan warna mulai terjadi pada saat BPC bernilai 3, namun perubahan sangat minimal (tidak signifikan). Pada pengujian berikutnya, perubahan warna RGB semakin signifikan mengikuti

peningkatan nilai BPC. Stego bitmap tampak mengalami perubahan warna RGB paling maksimal saat BPC bernilai 8. Jadi dapat ditarik kesimpulan bahwa semakin tinggi nilai BPC, semakin tinggi kemungkinan terjadi perubahan warna RGB pada stego bitmap.

Pada pengujian kali ini, penulis membandingkan dari sisi efisiensi waktu sisip dan ekstrak antara perangkat lunak steganografi dengan sebuah perangkat lunak keamanan data sejenis, yaitu Hide In Picture 2.1.

Tabel 5
Tabel Hasil Perbandingan Dengan Perangkat Lunak Sejenis

Nama Aplikasi	Nama file Cover BMP	Ukuran Cover BMP	Proses Sisip	Proses Ekstrak	Integritas Data Sisip
Hide In Picture 2.1	iPad Commercial.flv	2.637	00:10:542	00:11:042	100%
HideMyData 1.0	iPad Commercial.flv	2.637	00:05:521	00:06:649	100%

Dari Tabel 5 di atas, hasil pengujian proses enkripsi untuk perangkat lunak HideMyData 1.0 (aplikasi penulis) lebih cepat sekitar 5 detik bila dibandingkan dengan perangkat lunak Hide In Picture 2.1. Begitu pula halnya dengan proses ekstrak, dimana perangkat lunak HideMyData 1.0 lebih cepat sekitar 4.4 detik. Bila dilihat dari sisi efisiensi durasi proses sisip maupun proses ekstrak, perangkat lunak HideMyData 1.0 memiliki efisiensi yang lebih tinggi jika dibanding dengan perangkat lunak Hide In Picture 2.1. Sedangkan pada pengujian hasil ekstrak kedua perangkat lunak, keaslian dan integritas data sebelum sisip tidak mengalami perubahan jika dibanding keaslian dan integritas data sesudah ekstrak.

4.5 Evaluasi Perangkat Lunak

Setelah perangkat lunak stegnografi ini melalui proses implementasi, didapatkan bahwa kelebihan perangkat lunak ini adalah sebagai berikut: 1) Dapat digunakan untuk melakukan penyisipan berbagai jenis data dengan ekstensi yang berbeda; 2) Ukuran file bitmap setelah disisip (Stego Bitmap) tidak mengalami perubahan dari ukuran file bitmap sebelumnya (Cover Bitmap); 3) Dapat melakukan manipulasi BPC (Bit Per Channel) untuk meningkatkan daya tampung cover bitmap semaksimal mungkin; 4) Efisiensi waktu enkripsi dan dekripsi yang relatif tinggi (cepat); 5) Integritas data sebelum dan sesudah proses ekstrak tidak mengalami perubahan sama sekali.

5. KESIMPULAN

Dari hasil penelitian dan pembahasan, maka dapat diambil kesimpulan mengenai aplikasi steganografi dengan metode Least Significant Bit antara lain : 1) Dapat digunakan untuk melakukan penyisipan berbagai jenis data dengan ekstensi yang berbeda; 2) Ukuran file bitmap setelah disisip (Stego Bitmap) tidak mengalami perubahan dari ukuran file bitmap sebelumnya (Cover Bitmap); 3) Dapat melakukan manipulasi BPC (Bit Per Channel) untuk meningkatkan daya tampung cover bitmap semaksimal mungkin; 4) Efisiensi waktu enkripsi dan dekripsi yang relatif cepat; dan 5) Integritas data sebelum dan sesudah proses ekstrak tidak mengalami perubahan sama sekali.

DAFTAR PUSTAKA

Anggraini, Ema U, 2007, *Analisis Penyisipan Data Pada Citra Bitmap Menggunakan Metode Bit Plane Complexcity Segmentation*, http://digilib.unsri.ac.id/-download/22%20%20analisis%20penyisipan%20data%20pada%20citra%20bitmap%20menggunakan%20metode%20bit%20plane%20complexcity%20segmentation%20-%20amikom14082009.pdf diakses tanggal 26 Maret 2010.

Arubusman, Yusrian Roman, 2007, *Audio Steganografi*, http://iwayan.info/FileMahasiswa/SkirpsiYusAudioSteganografi_2007.pdf, diakses tanggal 27 Maret 2010.

Basuki, Dwi Kurnia., Nadhori, Isbat Uzzin., Maulana, Ahmad Mansur., 2009, Data Hiding Steganograph Pada File Image Menggunakan Metode Least Significant Bit, *Jurnal Teknik Informatika*

Fauzy, Gita Atika, 2009, Pengolahan Data Keluaran DTMF Decoder Untuk Mengendalikan Peralatan Listrik, http://repository.usu.ac.id/bitstream/123456789/7889/1/09E00874.pdf diakses tanggal 29 Juli 2010.

Hakim A., 2007, *Studi dan Implementasi Steganografi Metode LSB dengan Preprocessing Kompresi data dan Ekspansi Wadah*, http://www.informatika.org/~rinaldi/Kriptografi/20072008/Makalah1/MakalahIF5054-2007-A077 .pdf diakses tanggal 30 Maret 2010.

Munir, Rinaldi, 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, Bandung.

Munir, Rinaldi, 2004, *Steganografi dan Watermarking*, http://www.informatika.org/~rinaldi/Kriptografi/Steganografi%20dan%20Watermarking.pdf, diakses tanggal 26 Maret 2010.

N.F. Johnson, J. Suhil, 2006, *Exploring Steganography : Seeing the Unseen*, Computing Practices, http://www.jjtc.com/pub/r2026.pdf, diakses tanggal 26 Maret 2010.

Pakereng, M.A., Ineke., Beeh, Yos Richard., Endrawan, Sonny., 2010, Perbandingan Steganografi Metode Spread Spectrum dan Least Significant Bit (LSB) Antara Waktu Proses dan Ukuran File Gambar, *Jurnal Informatika*, Vol. 6 No. 1, April 2010

Ray Rizaldy, M, 2009, *Teknik Penyembunyian Pesan Rahasia Pada Berkas Video*, http://www.informatika.org/~rinaldi/Kriptografi/2008-2009/Makalah1/MakalahIF30581-2009-a037.pdf diakses tanggal 7 Agustus 2010.

Rude, Thomas, Jhonson, Neil F, 2001, *Introduction To Steganography : Hidden Information*, George Manson University.

Sofyanti, Ledy., 2011, Steganography with Least Significant Bit Methods For In Their Use Confidential Information Storage Media In Figure, *Gunadarma University E-Paper*, Faculty of Industrial Technology