PENGACAKAN POLA STEGANOGRAFI UNTUK MENINGKATKAN KEAMANAN PENYEMBUNYIAN DATA DIGITAL

Nova Hadi Lestriandoko
Lembaga Ilmu Pengetahuan Indonesia, Pusat Penelitian Informatika
Komplek LIPI, Gedung 20, Lantai 3 Jalan Cisitu No. 21/154D, Sangkuriang Bandung 40135
E-mail: ryan@informatika.lipi.go.id

ABSTRAKSI

Penggabungan kriptografi dan steganografi telah dilakukan untuk meningkatkan keamanan data yang disembunyikan dalam image. Metode ini menggunakan 16 karakter atau 128 bits password dan 1 karakter atau 8 bits key untuk menentukan pola penyisipan karakter ke dalam image pada saat proses inisialisasi dan untuk mengacak informasi atau data masukan sebelum digunakan untuk proses steganografi. Hasil dari metode ini adalah stegano image yang serupa dengan image original, sehingga perbedaan dari kedua image tersebut tidak akan terlihat secara visual.

Kata kunci: *kriptografi, steganografi, image.*

1. PENDAHULUAN

Seiring dengan maraknya *hacker* di dunia maya, data-data rahasia terutama data rahasia negara sangat rawan untuk dibongkar oleh para *hacker*. Walaupun sudah bermunculan metode steganografi dan enkripsi data yang rumit dan canggih, namun software untuk membaca dan memecahkan kedua metode tersebut juga terus berkembang [6]. Software untuk mendeteksi tersebut diantaranya adalah *Stegdetect* [http://www.outguess.org/].

Kriptografi merupakan seni untuk mengacak data sehingga tidak dapat dipecahkan atau dibaca oleh orang lain. Kriptografi meliputi enkripsi dan dekripsi. Enkripsi adalah proses mengacak data sehingga tidak dapat dibaca oleh pihak lain. Pada kebanyakan proses enkripsi harus menyertakan kunci sehingga data yang dienkripsi dapat didekripsikan kembali. Gambaran sederhana tentang enkripsi, misalnya mengganti huruf a dengan n, b dengan m dan seterusnya. Model penggantian huruf sebagai bentuk enkripsi sederhana ini sekarang tidak dipergunakan secara serius dalam penyembunyian data. Sedangkan dekripsi merupakan kebalikan dari proses enkripsi, yaitu proses membaca data acak sehingga diperoleh data yang dapat digunakan lagi.

Metode kriptografi saat ini sudah banyak yang dikembangkan dan bervariasi cara enkripsinya. Salah satu yang lazim dipakai saat ini yaitu penggunaan password dan key untuk mengacak data lebih lanjut sehingga keamanan dari enkripsi tersebut juga lebih terjamin. Contoh dari beberapa metode kriptografi yang populer adalah enkripsi blowfish [5], RSA, DES, Triple DES, RC5, dsb [1][2].

Steganografi merupakan seni untuk menyembunyikan informasi di dalam suatu media sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu di dalam media tersebut. Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos*, yang artinya 'tersembunyi/terselubung', dan *graphein*, 'menulis' sehingga kurang lebih artinya "menulis (tulisan) terselubung" [4][6]. Terdapat banyak sekali metode untuk menyembunyikan informasi rahasia. Beberapa contoh metode yang telah ada yaitu tinta yang tidak tampak, *microdots*, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi spektrum lebar [3] [4][6]. Pada gambar 1 diperlihatkan contoh steganografi, yaitu gambar satelit pusat strategi pengebom Uni Soviet yang disamarkan ke dalam gambar lukisan Pierre-Auguste Renoir yang berjudul ”Le Moulin de la Galette”.

Gambar 1. Contoh steganografi[4]

Pada jaman Yunani kuno, manusia menggunakan lempengan kayu yang dilapisi oleh lilin sebagai media untuk mengirim pesan rahasia. Untuk membuat pesan rahasia, seseorang harus mengikis lapisan lilin yang menutupi media, menulis pesan pada dasar media dan melapisi lilin kembali untuk menutupi pesan yang telah ditulis supaya terlihat kosong dan tidak mengandung informasi apapun. Sedangkan tinta yang tidak tampak digunakan sebagai metode steganografi eksklusif pada permulaan perang dunia II. Pada surat biasa yang tidak mencurigakan bisa jadi mengandung suatu informasi rahasia yang tidak tampak diantara baris-baris isi surat [3] [4].

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

menyembunyikan pesan sehingga tidak terlihat. Pesan dalam cipherteks mungkin akan menimbulkan kecurigaan sedangkan pesan yang dibuat dengan steganografi tidak akan menimbulkan kecurigaan karena secara visual tidak kelihatan. [6]

S-Tools for Windows by Andy Brown merupakan salah satu software steganografi serbaguna yang menggunakan media *image* dan suara. Termasuk di dalamnya program untuk memproses *image* GIF dan BMP(ST-BMP.exe), *file audio* WAV (ST-WAV.exe) dan juga penyembunyian informasi pada area yang tidak digunakan pada disket floppy (ST-FDD.exe). Sebagai tambahan untuk mendukung *image* 24-bit, S-Tools juga memasukkan beberapa rutin enkripsi (IDEA, MPJ2, DES, 3DES and NSEA) dengan beberapa *option* [3].

S-Tools menerapkan metode LSB atau penyisipan pada bit rendah untuk steganografi baik dengan media *image* maupun *audio*. Sebagai pembanding metode yang dikembangkan pada tulisan ini, dibatasi hanya pemakaian dengan media *image* 24 bits. Masukan dari *software* ini adalah *image* 24-bit dan pesan berupa *file text*. S-Tools juga menambahkan ekstra informasi pada permulaan pesan sebelum disembunyikan pada media *image*.

Beberapa *software* lainnya seperti *StegoDos* dan *White Noise Storm* juga telah didiskusikan sebelumnya [4]. Tulisan ini mengambil S-Tools sebagai pembanding karena diantara ketiga software yang ada pada referensi, S-Tools yang paling optimal [4]. *Image* hasil steganografi menggunakan S-Tools lebih tidak kentara perubahannya secara visual dibandingkan kedua metode lainnya.

Pada tulisan ini akan dibahas gabungan metode enkripsi data dan penyamaran data ke dalam *image* atau lebih sering disebut steganografi dengan menggunakan media *image* 24 bits. Dengan demikian diharapkan metode integrasi ini akan menghasilkan *image* stegano yang lebih sulit untuk dipecahkan dan diketahui oleh para *hacker*.

2. DASAR TEORI

2.1 File Image

File Image pada komputer merupakan *array* bilangan yang merepresentasikan nilai intensitas cahaya yang bervariasi (*pixel*). Kumpulan *pixel-pixel* inilah yang membentuk suatu *image*. *Image* yang sering digunakan umum adalah *image* 24 *bits* dan *image* 8 *bits* (256 *colors*) [4].

Pada steganografi, *image* yang biasa digunakan adalah *image* 24 *bits*, karena *image 24 bit*s tersebut dapat menyediakan space yang besar untuk disisipi oleh data. *Pixel* penyusun *image* ini tersusun atas 3 warna primer yaitu merah, hijau, dan biru. Masing-masing warna primer tersusun atas 1 *byte* data. Untuk *image 24 bits* berarti menggunakan *3 bytes per pixel* untuk merepresentasikan nilai warna *pixel*. *3 bytes* data ini dapat berupa hexadesimal, desimal, ataupun biner. Pada kebanyakan webpage menggunakan warna *background* yang direpresentasikan oleh 6 digit bilangan heksadesimal 000000...FFFFFF yang terdiri atas 3 warna primer yang masing-masing tersusun atas 2 digit heksadesimal. Putih bernilai FFFFFF, biru bernilai FF0000, hijau bernilai 00FF00, merah bernilai 0000FF, dan hitam bernilai 000000. Nilai dalam desimal berkisar antara 0..255 untuk masing-masing warna primer dan 00000000..11111111 untuk nilai dalam biner.

2.2 Steganografi

Salah satu metode steganografi yang umum digunakan saat ini yaitu informasi rahasia disembunyikan pada *image* dengan cara menyisipkannya pada *bit* rendah pada data *pixel* yang menyusun *image* tersebut (LSB – *Least Significant Bit*). Seperti kita ketahui untuk *image bitmap 24 bit*s maka setiap *pixel* (titik) pada *image* tersebut terdiri atas susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bits dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian pada setiap *pixel image bitmap 24 bits* kita dapat menyisipkan *3 bits* data. Contohnya huruf A dapat kita sisipkan dalam *3 pixel* (Metode LSB [4]), misalnya data raster original adalah sebagai berikut:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Sedangkan representasi biner huruf A adalah 10000011. Dengan menyisipkan-nya pada data *pixel* diatas maka akan dihasilkan:

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001001 00100111 11101001)

Terlihat hanya empat bit rendah yang berubah, untuk mata manusia maka tidak akan tampak perubahannya. Secara rata-rata dengan metoda ini hanya setengah dari data bit rendah yang berubah, sehingga bila dibutuhkan dapat digunakan bit rendah kedua bahkan ketiga.

Untuk jenis media yang lain digunakan metoda yang berbeda-beda pula sesuai dengan format file yang digunakan. Tetapi bagaimanapun dengan steganografi bukan berarti pengiriman pesan menjadi benar-benar aman karena telah dikembangkan juga metoda-metoda untuk mendeteksi keberadaan pesan yang dibuat dengan cara ini [4] [6].

2.3 Pengukuran Image Stegano

Image hasil proses steganografi ini dapat dibandingkan dengan *image* aslinya (*image* sebelum diproses) untuk mengetahui perubahan warna per *pixel* dari *image* tersebut. Ukuran tersebut meliputi total perubahan, rata-rata perubahan per *pixel*, dan perubahan maksimum per *pixel*. Pengukuran ini dilakukan dengan menggunakan rumus Jarak Euclidean (*Euclidean Distance*)[7]:

$$D_i = \sqrt{\sum_{k=1}^{p} (x_{ik} - y_{ik})^2} \quad (1)$$

dimana x_i adalah *pixel* ke i pada *image* hasil steganografi, y_i adalah *pixel* ke i pada *image* asli, dan p adalah jumlah *band* (warna primer RGB).

3. METODE

Metode yang digunakan untuk tulisan ini dibagi menjadi empat macam, antara lain dapat dilihat pada Tabel 1.

Tabel 1. Metode Steganografi

Metode 1	Data teks disisipkan ke dalam *image* menggunakan metode penyisipan 1 karakter ke dalam 1 *pixel*.
Metode 2	Data teks disisipan ke dalam *image* menggunakan metode penyisipan 1 karakter ke dalam 3 *pixel*.
Metode 3	Data teks disisipkan ke dalam *image* menggunakan metode penyisipan 1 karakter ke dalam 9 *pixel*.
Metode 4	Data teks disisipkan ke dalam *image* menggunakan metode penyisipan 1 karakter ke dalam 9 *pixel* yang digabungkan dengan metode enkripsi untuk mengacak pola penyisipan data.

Dari Tabel 1, metode 1 sampai dengan 3 tidak menggunakan enkripsi. Sedangkan metode 4 diintegrasikan dengan enkripsi untuk menambah tingkat keamanan penyamaran data ke dalam *image*.

Metode 1 menggunakan penyisipan 1 karakter ke dalam 1 *pixel*. Langkah pertama yaitu mengkonversi nilai ASCII karakter masukan ke dalam nilai biner yang hasilnya adalah nilai biner 9 bit. Nilai biner tersebut kemudian dibagi menjadi 3 bagian masing-masing 3 bit, yang selanjutnya ketiga bagian tersebut ditambahkan ke dalam nilai *Red*, *Green*, dan *Blue* pada *pixel image*. Metode ini masih mempunyai kelemahan yaitu perubahan warna pada *image* masih terlihat dengan jelas secara visual karena untuk satu *pixel* 24 bit akan mengalami perubahan maksimum 9 bit. Prosentase perubahan maksimumnya adalah 37,50 % per *pixel*.

Metode 2 menggunakan penyisipan 1 karakter ke dalam 3 *pixel*. Nilai biner karakter masukan dibagi menjadi 3 bagian masing-masing 3 bit. Ketiga bagian tersebut ditambahkan ke dalam nilai RGB 3 *pixel image*. Nilai RGB *pixel* pertama, *pixel* kedua dan *pixel* ketiga masing-masing ditambahkan 3 bit. Perubahan maksimum dari masing-masing *pixel* adalah 3 bit per *pixel*. Prosentase perubahan maksimumnya adalah 12,50 % per *pixel*. Hasil dari metode ini juga masih terlihat perubahan warna *image* secara visual.

Metode 3 menggunakan penyisipan 1 karakter ke dalam 9 *pixel image*. 8 bit nilai biner karakter masukan ditambahkan nilai '0' didepannya sehingga menjadi 9 bit. Masukan ini kemudian ditambahkan ke dalam 9 *pixel image* sehingga masing-masing *pixel* hanya memiliki perubahan 1 bit per *pixel*. Prosentase perubahan maksimumnya adalah 4,167 % per *pixel*. Metode ini menghasilkan *image* yang perubahannya tidak tampak secara visual.

Metode 4 merupakan fokus dari tulisan ini. Metode ini hampir sama dengan metode 3 yaitu menyisipkan 1 karakter ke dalam 9 *pixel*. Perbedaannya adalah metode ini mengintegrasikan enkripsi untuk meningkatkan keamanan data. Metode ini menggunakan password yang terdiri atas maksimum 16 karakter atau 128 bit dan key yang terdiri atas 1 karakter atau 8 bit. Pada proses initialisasi, password dan key ini digunakan untuk mengacak pola penyisipan karakter ke dalam *pixel image*. Penjelasan lebih detil tentang pengacakan pola steganografi ini akan dibahas pada subbab berikutnya. Metode ini juga menambahkan ekstra informasi di akhir pesan masukan sebelum disembunyikan. Setelah pola penyisipan ditentukan, karakter masukan dienkripsi dengan menambahkan atau mengurangkan masing-masing karakter dengan password dan key. Nilai biner hasil dari pengacakan data masukan tersebut kemudian disisipkan ke dalam *image* mengikuti pola yang telah ditentukan pada proses initialisasi. Metode ini menghasilkan *image* dengan perubahan maksimum 1 bit per *pixel* atau 4,167 % per *pixel*. Sama dengan hasil metode 3, perubahan ini tidak akan tampak secara visual. Perubahan minimum dari metode ini adalah 0 % yaitu jika pola yang terbentuk dari hasil enkripsi dan steganografi sama persis dengan pola *image* aslinya.

Keempat metode steganografi diatas dibuat sedemikian rupa sehingga untuk membaca informasi yang tersembunyi tidak diperlukan perbandingan dengan file *image* asli. Hal ini untuk menghilangkan kecurigaan bahwa *image* mengandung suatu informasi rahasia. Setelah dihasilkan *image* stegano, file *image* asli bisa langsung dihapus sehingga keamanan lebih terjamin.

3.1 Pengacak Pola Steganografi

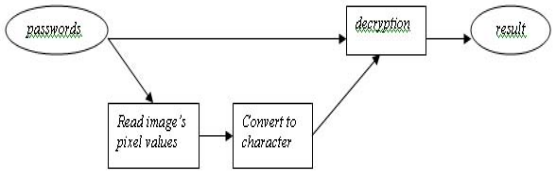
Proses mengacak pola steganografi memerlukan beberapa langkah yang melibatkan kriptografi untuk mengolah password. Elemen yang diacak meliputi ukuran matrik *pixel* dan lokasi penyisipan.

- Langkah 1: Pengolahan password dengan memanfaatkan kriptografi sehingga didapatkan nilai x.
- Langkah 2: Nilai x tersebut digunakan untuk menentukan ukuran matrik. Misalkan, jika $x = x_1$ maka digunakan matrik 9x1, jika $x = x_2$ maka digunakan matrik 3x3, jika $x = x_3$ maka digunakan matrik 1x9, dst. x_1, x_2, x_3, ... , x_n adalah variabel kriptografi.
- Langkah 3: Pemetaan nilai x untuk menentukan lokasi penyisipan pada nilai RGB *pixel*. Misalkan, x_1 untuk penyisipan di warna merah, x_2 untuk penyisipan di warna hijau, dst.
- Langkah 4: Menyisipkan data yang telah dienkripsi ke dalam bit terakhir dari nilai RGB yang telah ditentukan dengan ukuran matrik *pixel* yang telah dipilih pada langkah 2 dan 3.

Pengacakan pola steganografi ini bertujuan untuk mempersulit pendeteksian sehingga dapat meningkatkan keamanan data yang disembunyikan.

3.2 *Unhide* Data

Pertanyaan yang pasti muncul jika data telah disembunyikan yaitu ”Bagaimana cara kita membaca data yang telah disembunyikan?”. Proses untuk membaca data tersembunyi ini memerlukan beberapa tahapan. Metode untuk membaca *image* hasil steganografi dijelaskan pada Gambar 2.



Gambar 2. Diagram alur proses membaca steganografi

Langkah pertama yaitu membaca nilai *pixel-pixel* pada *image* dengan menggunakan password untuk menentukan pola pembacaan. Langkah selanjutnya yaitu mengkonversi nilai biner yang didapat ke dalam nilai ASCII agar didapatkan sebuah karakter. Setelah didapatkan nilai karakter yang acak, dilakukan dekripsi supaya data yang acak tersebut bisa dibaca. Pengolahan password dan key diperlukan untuk proses membaca karakter dari image dan untuk mendekripsi karakter.

4. HASIL DAN ANALISA

Data sampel yang digunakan dalam tulisan ini adalah *image* dengan format 24 bit dan data masukan berupa teks. Pada tabel 2 ditampilkan kapasitas maksimum data yang dapat disembunyikan dalam *image* menggunakan beberapa metode steganografi dengan ukuran *image* yang bervariasi.

Tabel 2. Kapasitas Maksimum *Image*

Metode	***Ukuran Image***	***Kapasitas Data (bits)***	***Kapasitas Data (bytes)***
1	454 x 336	1.372.896	171.612
	800x600	4.320.000	540.000
	500x375	1.687.500	210.937
2	454 x 336	457.632	57.204
	800x600	1.440.000	180.000
	500x375	562.500	70.312
3 dan 4	454 x 336	152.544	19.068
	800x600	480.000	60.000
	500x375	187.500	23.437

Dari tabel 2 dapat diketahui bahwa kapasitas data maksimum yang dapat dimasukkan ke dalam *image* dipengaruhi oleh ukuran *image* dan metode steganografi yang digunakan untuk menyisipkan informasi ke dalam *image*. Semakin besar ukuran *image* maka semakin besar pula data yang dapat disisipkan. Sedangkan untuk metodenya, semakin banyak *pixel* yang dipakai untuk menyisipkan satu karakter, maka semakin kecil kapasitas data yang dapat disisipkan ke dalam *image*.

Data yang digunakan untuk uji coba ada 2 macam yaitu pesan teks (*Messages*) untuk masukan dan media steganografi berupa *image* 24 bit (*Container*). Pesan Teks masukan 1 yang berupa *plain-text* untuk selanjutnya disebut M1:

> *Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the “enemy” is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other “harmless” messages in a way that does not allow any “enemy” to even detect that there is a second secret message present. [Markus Kuhn 1995-07-03].*

Sedangkan media steganografi yang berupa *image* 24 bit diambil 3 sampel *image* yang masing-masing memiliki karakter yang berbeda. C1 yang memilki variasi warna tinggi dengan ukuran 454 x 336 diperlihatkan pada gambar 3a dan contoh *image* stegano hasil metode 4 (penggabungan M1 dengan C1) diperlihatkan pada Gambar 3b.

(a)

(b)

Gambar 3. Container C1(a) dan hasil metode 4(b)

Media C2 yang memiliki variasi warna lebih rendah dengan ukuran 800x600 diperlihatkan pada gambar 4a dan contoh *image* stegano hasil metode 4 (penggabungan M1 dengan C2) diperlihatkan pada Gambar 4b.

Media C3 dengan ukuran 500x375 diperlihatkan pada gambar 5a dan contoh *image* stegano hasil metode 4 (penggabungan M1 dengan C3) diperlihatkan pada Gambar 5b.

Media C2 yang diperlihatkan pada gambar 6a dan contoh *image* stegano hasil metode 2 (penggabungan M1 dengan C2) diperlihatkan pada Gambar 6b.

Pengukuran variabel-variabel perubahan *image* stegano diperlukan untuk menguji tingkat kesempurnaan penyamaran. Hasil pengukuran perubahan *image* dengan menggunakan rumus 1 dapat dilihat pada Tabel 3 sampai dengan 5.

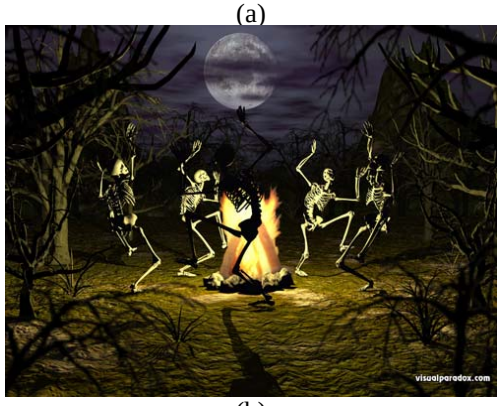
(a)

(b)

Gambar 4. Container C2(a) dan hasil Metode 4(b)

Dari data-data tersebut, yang paling berpengaruh terhadap perubahan *image* secara visual adalah nilai maksimum perubahan per *pixel*. Pada Gambar 3 s.d. 5 perbedaan pada kedua *image* tidak dapat dilihat secara visual, hal ini disebabkan perubahan maksimum metode 3 dan 4 adalah 1,0 sehingga tidak memungkinkan mata untuk mendeteksi perbedaan warna ini. Sedangkan pada gambar 6, bila diperhatikan secara seksama, masih terlihat perubahan warna *image* sehingga akan menimbulkan kecurigaan jika digunakan untuk menyembunyikan informasi rahasia. Hal ini disebabkan metode 1 dan 2 dengan media C2 memiliki maksimum perubahan paling besar yaitu 53 untuk metode 1 dan 50 untuk metode 2 (Tabel 4). Jadi semakin rendah nilai maksimum perubahan per *pixel* maka perubahan *image* tersebut akan semakin tidak kelihatan secara visual. Nilai perubahan maksimum ini juga dipengaruhi oleh pola dan variasi warna pada image. Jika kebetulan pola yang terbentuk oleh steganografi sama atau mirip dengan pola media yang digunakan, maka akan didapatkan nilai maksimum perubahan yang kecil seperti terlihat pada Gambar 7.

(a)



(b)

Gambar 5. Container C3(a) dan hasil metode 4(b)

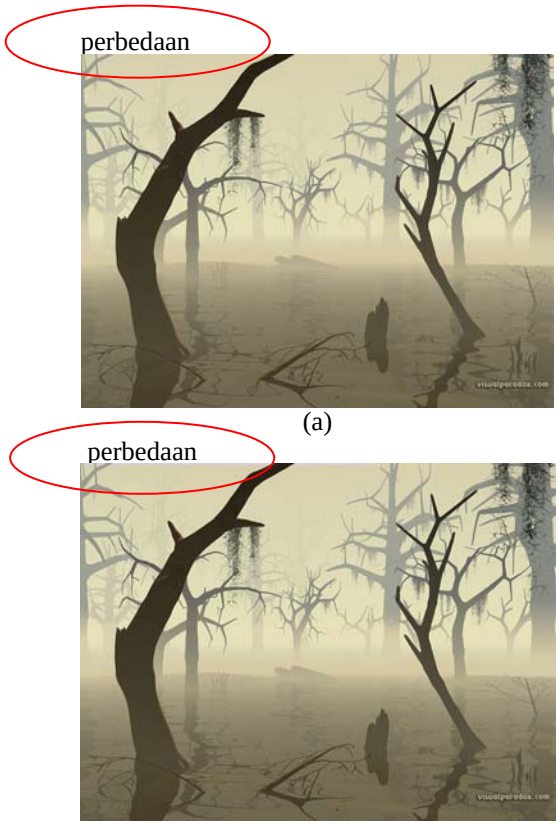
Pada C1 dan C3 seharusnya nilai perubahan maksimum pada metode 2 lebih kecil dari metode 1, tetapi ternyata lebih besar. Hal ini disebabkan oleh karena pola yang dihasilkan metode 1 lebih mirip dengan media C1 dan C3 dibandingkan metode 2. Sedangkan nilai total perubahan merupakan jumlah total dari nilai perubahan per *pixel image*. Nilai ini digunakan untuk mengetahui seberapa besar perubahan image secara keseluruhan.

5. KESIMPULAN

Semakin banyak pixel yang digunakan untuk menyisipkan sebuah karakter ke dalam *image*, akan semakin tidak tampak perubahan *image* tersebut secara visual karena nilai perubahan maksimum per *pixel image* akan semakin kecil. Hal ini terbukti pada metode 3 dan 4 yang menghasilkan *image* dengan maksimum perubahan *pixel* paling kecil yaitu 1,0 dibandingkan metode 1 dengan nilai 22 s.d. 53, metode 2 dengan nilai 35 s.d. 50 dan S-Tool dengan nilai 1,414 (diperlihatkan pada gambar 7). Kapasitas data yang dapat dimasukkan ke dalam *image* tergantung pada ukuran *image* yang digunakan dan jumlah *pixel* yang digunakan untuk menyisipkan suatu karakter pesan (metode steganografi). Semakin besar ukuran pesan yang

 ISSN: 1907-5022

ingin dimasukkan akan membutuhkan ukuran *image* yang semakin besar juga. Dengan menggunakan metode penyisipan 1 karakter ke dalam 9 *pixel*, mengacak pola penyisipan, dan mengacak data masukan terbukti lebih meningkatkan keamanan informasi yang disembunyikan dalam *image* sehingga selain lebih sulit untuk dilacak, juga lebih sulit untuk dipecahkan.



Gambar 6. Contoh perbedaan Container C2 (a) dan hasil Metode 2 (b)

Tabel 3. Tabel Perubahan *Image* C1 dan M1

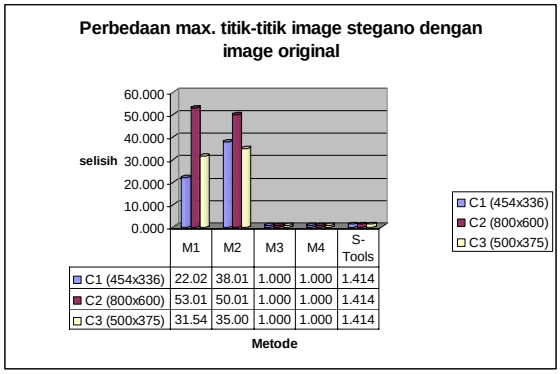
Method/ Software	***Total Perubahan***	***Rata-rata***	***Max. Perubahan***
M1	2.687,232	0,018	22,023
M2	5.962,429	0,039	38,013
M3	1.875,000	0,012	1,000
M4	1.704,000	0,011	1,000
S-Tools	1.730,899	0,011	1,414

Tabel 4. Tabel Perubahan *Image* C2 dan M1

Method / Software	***Total Perubahan***	***Rata-rata***	***Max. Perubahan***
M1	21.004,190	0,044	53,019
M2	61.666,036	0,128	50,010
M3	2.132,000	0,004	1,000
M4	2.322,000	0,005	1,000
S-Tools	1.763,828	0,004	1,414

Tabel 5. Tabel Perubahan *Image* C3 dan M1

Method / Software	***Total Perubahan***	***Rata-rata***	***Max. Perubahan***
M1	4.481,086	0,024	31,544
M2	10.875,663	0,058	35,000
M3	1.722,000	0,009	1,000
M4	1.725,000	0,009	1,000
S-Tools	1.774,899	0,009	1,414



Gambar 7. Grafik perbedaan maksimum.

DAFTAR PUSTAKA

[1] William Stallings, *Cryptography and Network Security: Principles and Practice Second Edition*, Prentice Hall, 1999.

[2] C.A. Deavours, David Kahn, Louis Kruh, Greg Mellen and Brian Winkel, *Cryptologi: Yesterday, Today, and Tomorrow*, Artech House, 1987.

[3] Neil F. Johnson, Steganography, George Mason University, Information System and Software Engineering, *www.jjtc.com/stegdoc/steg1995.html*

[4] Neil F. Johnson and Sushil Jajodia, Steganography: Seeing the Unseen. *IEEE Computer*, February 1998: 26-34.

[5] Budi Sukmawan, Metode Enripsi Blowfish,*http://bdg.centrin.net.id/~budskman/blowfish.htm*, April 2000.

[6] Budi Sukmawan, Steganografi, *bdg.centrin.net.id/~budskman/stegano.htm*, February 2002.

[7] Arto Kaarna, Pavel Zemcik, Heikki Kalviainen, and Jussi Parkkinen, Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction, *IEEE Trans. Geosci. Remote Sensing*, vol.38, no.2, March 2000.