# STUDI DAN DETEKSI STEGANOGRAFI PADA *FILE* BERTIPE *JPEG* DENGAN TIGA *STEGANOGRAPHIC SYSTEM*

Rhesa Adythia – NIM : 13505081

*Program Studi Teknik Informatika, Institut Teknologi Bandung*
*Jl. Ganesha 10, Bandung*
E-mail : if15081@students.if.itb.ac.id

## Abstrak

Makalah ini membahas tentang studi dan deteksi Steganografi (*Steganography*) untuk menyandikan data yang tersimpan dalam format *JPEG*. Steganografi (*Steganography*) merupakan sebuah algoritma untuk membuat orang yang tidak berkepentingan tidak dapat mengetahui isi suatu data seperti kriptografi namun keberadaan data atau pesan tersebut tetap ada, tetapi hanya pesan sebenarnya yang disamarkan. Steganografi (*Steganography*) menjadi cukup populer karena data hasil enkripsinya tidak menarik perhatian karena data yang dikirim masih dapat dilihat dan dibaca secara wajar oleh siapapun, namun pesan sebenarnya yang ada dibalik pesan tersebut yang tidak dapat diketahui. Data yang dimaksudkan adalah *Digital Image*. Implementasi *Steganography* dalam makalah ini meliputi tiga sistem operasi yaitu sistem operasi *Jsteg*, *Outguess*, dan *F5*.

*Jsteg* adalah sistem steganografi yang sudah cukup lama diperkenalkan secara luas di internet. Algoritma *Jsteg* merupakan algoritma penyamaran *Digital Image* yang paling mudah untuk diketahui dan di dekripsikan. Hal ini disebabkan pesan yang disisipkan berada pada awal data.

*Outguess* merupakan algoritma yang lebih sulit untuk diketahui dan di dekripsikan bila dibandingkan dengan *Jsteg*. Sistem enkripsi *Outguess* tersedia di internet dalam bentuk *Source Code* untuk UNIX. Penyisipan pesan yang menyebar pada seluruh data cukup membuat *Outguess* sulit untuk diketahui.

*F5* menyediakan ruang untuk penyisipan data yang besar. *F5* memiliki efisiensi yang besar karena penggunaan sistem *matrix encoding*. *F5* merupakan sistem yang tersulit unutk diketahui dan didekripsikan bila dibandingkan dengan kedua *steganographic system* sebelumnya.

Ketiga *steganographic system* menggunakan media penyipanan berupa Image yang biasanya bertipe *JPEG*. Hal ini disebabkan karena penggunaan yang meluas dari tipe ini di internet sehingga mempermudah pengiriman pesan rahasia.

***Kata kunci***: steganografi, *Steganography*, *Steganographic System*, *Jsteg*, *Outguess*, *F5*, *Image*, enkripsi, dekripsi.

## 1. Pendahuluan

Steganografi adalah seni dan ilmu untuk menyembunyikan fakta dari pertukaran informasi. *Steganographic system* dapat menyembunyikan pesan rahasia dalam gambar digital atau obyek digital lainnya. Bagi orang biasa yang melihat gambar tersebut, pesan rahasianya tidak terlihat.

Pada Februari 2000, USA Today melaporkan bahwa teroris menggunakan steganografi untuk menyembunyikan komunikasi mereka dari pemerintahan. Menurut mereka, pesan rahasia disembunyikan dalam gambar digital yang dikirimkan ke situs internet publik seperti *eBay* atau *Amazon*. Artikel tersebut minim dengan informasi teknis yang memungkinkan pembacanya untuk menguji kebenaran berita tersebut. Namun, artikel tersebut tetap saja di publikasi oleh beberapa kantor berita.

Penyimpanan pesan dengan cara ini menyulitkan orang banyak untuk mengetahui apakah ada pesan rahasia dibalik suatu data (dalam hal ini gambar digital). Steganografi dapat mengelabui siapapun dengan sangat mulusnya. Satu- satunya cara mengungkapkan

pesan rahasia tersebut adalah dengan suatu tools perangkat lunak khusus untuk memeriksa apakah gambar digital tersebut disisipi oleh pesan rahasia.

Setiap *Steganographic system* memiliki keunggulan dan kerugiannya masing-masing. Penggunaan setiap sistem yang berbeda harus disertai tools yang berbeda pula untuk pemeriksaan pesan rahasianya. Hal ini disebabkan karena setiap sistem memiliki algoritmanya sendiri untuk menyisipkan pesan rahasia ke dalam gambar digital. Mulai dari Jsteg yang menyimpannya pada awal data sampai algoritma *F5* yang memakai *Matrix Encoding*.

Steganografi membuat pesan rahasia yang dikirimkan terlihat tidak mencurigakan. Sehingga siapapun yang melihat atau memeriksanya tidak akan menyadari adanya pesan rahasia karena tidak adanya simbol-simbol yang tidak biasa seperti hasil enkripsi kriptografi.

Melihat keunggulan yang ditunjukkan oleh steganografi bila dibandingkan dengan kriptografi biasa membuat pentingnya pengenalan yang lebih mengenai cara kerja steganografi ini. Untuk memperkaya pemahaman mengenai steganografi, perlu juga pelajaran mengenai pendeteksian gambar digital hasil steganografi.

## 2. Latar Belakang Steganografi

Masa dari "Penyembunyian Informasi" berhubungan dengan *watermarking* dan steganografi. *Watermarking* biasanya adalah metode untuk menyembunyikan informasi dalam suatu obyek sehingga informasinya tetap ada tidak terganggu maupun berubah. Hal ini menyatakan bahwa tidak mungkin menghapus cap air tanpa menurunkan kualitas dari obyek tersebut.

Steganografi adalah metode untuk menyembunyikan informasi yang mudah rusak. Modifikasi suatu tahap tertentu dapat menghancurkannya.

*Watermarking* dan steganografi berbeda satu sama lain dalam hal yang mendasar. Dalam steganografi, informasi pasti tidak akan pernah bisa diketahui oleh orang yang melihatnya tanpa mengetahui adanya informasi yang tersembunyi. Sedangkan untuk *watermark* hal ini tidak pasti.

Keamanan dari *Steganographic system* klasik bergantung pada kerahasiaan dari *encoding system*. Sekali *encoding system* diketahui, *Steganographic system* tidak dapat menyembunyikan apa-apa. Contoh yang terkenal dari sistem klasiknya adalah Jenderal Roma yang mencukur kepala seorang budak dan membuat tato yang berisi pesan rahasia di kepalanya. Setelah rambut budak tersebut tumbuh kembali, budak tersebut dikirimkan untuk menyampaikan pesan. Jika sistemnya sudah diketahuia, siapapun dengan mudah mencukur kepala budak pengirim pesan tersebut dan membaca pesan rahasianya.

Encoding system yang lain mungkin untuk menggunakan kata terakhir dari tiap kalimat atau bit signifikan yang paling sedikit pada suatu gambar digital.

Steganografi modern dapat diketahui hanya jika informasi rahasianya diketahui, disebut sebagai kunci rahasia. Hal ini sangat mirip dengan "Kerckhoffs Principle" dalam kriptografi.

Karena "*invasive nature*"nya, *Steganographic systems* meninggalkan jejak yang dapat terdeteksi pada karakteristik suatu medium. Hal ini memungkinkan pendeteksian pada media yang dimodifikasi, dan mengungkapkan komunikasi rahasia yang berlangsung. Meskipun isi rahasianya tidak terlihat, keberadaanya dapat diketahui. Hal ini membuat tujuan utama dari staganografi terbongkar.

Modifikasi dari bit yang berlebihan dapat merubah keterangan statistik dari medium yang menyebunyikannya. Jadi, analisis statistik dapat mengungkapkan data yang tersembunyi.

## 3. Gambar digital dengan format JPEG

Gambar *JPEG* (Gambar digital dalam format *JPEG*) sangat umum digunakan dalam website di internet. Format gambar *JPEG* mengunakan *discrete cosine transform* (*DCT*) untuk mengubah blok 8 x 8 piksel dari gambar ke 64 *DCT* koefisien masing-masing. Bit signifikan yang paling sedikit dari jumlah koefisien *DCT* digunakan sebagai bit yang paling banyak sebagai tempat pesan rahasia disisipkan.

Dalam beberapa format gambar digital seperti *GIF*, struktur visual dari gambar berada di seluruh *bit-layer* dari gambar digital.

*Steganographic system* yang memodifikasi bit signifikan yang paling sedikit seringkali mudah terlihat oleh pengelihatan manusia.

Hal ini berbeda dengan format *JPEG*. Modifikasi yang dilakukan pada koefisien *DCT* tunggal mempengaruhi ke 64 piksel dari gambar. Karena alasan tersebut, tidak ada yang dapat mengetahui keberadaan pesan rahasia dengan steganografi hanya dengan melihat suatu file gambar dengan format *JPEG*.

Gambar 1 menunjukkan 2 gambar dengan resolusi 800 x 600 dan *24-bit color depth*. Gambar digital asli tanpa kompresi berukuran 12 Mb, namun kedua gambar JPEG yang ditunjukan hanya berukuran 0,3 Mb. Gambar yang diatas belum dimodifikasi. Gambar di bawah berisi bab pertama dari "The Hunting of the Snark" yang ditulis oleh Lewis Carroll. Setelah kompresi, bab tersebut memiliki ukuran sekitar 14,700 bit. Sangat tidak mungkin untuk pengelihatan mata manusia dapat menemukan perbedaan dari keduanya.

***Gambar 1. Gambar atas tidak dimodifikasi, Gambar bawah berisi bab pertama dari "The Hunting of the Snark" yang ditulis oleh Lewis Carroll.Tidak ada perbedaan visual yang terlihat.***

**4. Tipe *Steganographic system***

Steganographic system yang biasa digunakan adalah :

1. *Jsteg*
   Algoritma Steganografi yang menyisipkan pesan rahasia pada bagian awal data. Lebih mudah untuk diketahui keberadaannya.
2. *Outguess*
   Algoritma Steganografi yang menyisipkan pesan rahasia secara menyebar pada seluruh bagian data. Lebih sulit untuk diketahui keberadaannya dibandingkan dengan *Jsteg*.
3. *F5*
   Algoritma yang lebih kuat menghadapi pemeriksaan statistik karena menggunakan *Matrix Encoding*. Sistem ini merupakan sistem yang paling baru bila dibandingkan dengan kedua sistem yang lain.

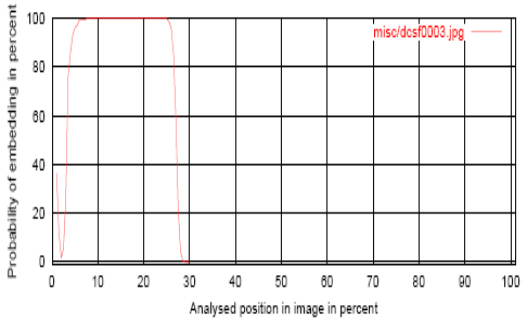
**4.1 *Jsteg***

*Jsteg* adalah penyempurnaan *Independent JPEG Group's JPEG Software library* oleh Derek Upham. Koefisien *DCT* dimodifikasi secara kontinu dari awal suatu gambar digital. *Jsteg* tidak mendukung enkripsi dan tidak ada pemilihan bit acak.

Data dari suatu pesan di gabung di awal dengan ukuran header yang bervariasi. Lima bit pertama dari header menyatakan ukuran *field* dalam bit. Lima bit selanjutnya menyatakan ukuran *field* yang menyatakan ukuran data yang disisipkan.

Perangkat lunak *Jsteg-Shell* bekerja dalam lingkungan Windows dan dibuat oleh Korejwa. Perangkat lunak ini mendukung enkripsi dan penyisipan data dengan *Jsteg*.

Menurut hasil tes $\chi$2-test terhadap gambar digital yang berisi informasi tersembunyi dengan sistem *Jsteg*, Kemungkinan terbesar dari data yang disisipkan berada pada awal data gambar digital. Tidak ada kemungkinan yang lain. Lihat Gambar 2.




**Gambar 2. Gambar digital berisi pesan rahasia dengan sistem *Jsteg* menunjukkan kemungkinan terbesar disisipkan di awal gambar. Hasil menunjukkan 0 ketika test mencapai bagian yang tidak dimodifikasi dari koefisien DCT.**

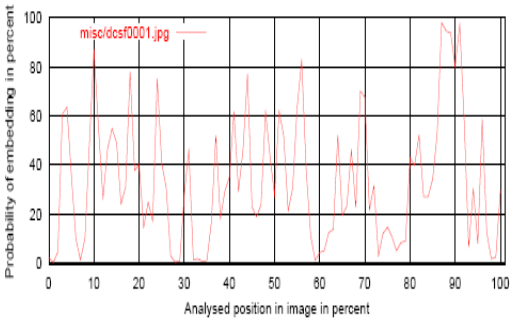
### 4.2 *Outguess*

*Outguess* adalah *Steganographic system* yang tersedia dalam *UNIX source code*. Telah ada dua versi yaitu : OutGuess 0.13b, yang mudah diketahui oleh analisis statistik, dan OutGuess 0.2, yang memiliki kemampuan untuk mengelabui keterangan statistik dan tidak dapat dideteksi dengan statistik tes.

*Outguess* berbeda dengan sistem yang dibahas pada bagian sebelumnya (*Jsteg*) yang memilih koefisien *DCT* dengan sebuah *pseudo-random number generator*. Penggunanya harus memasukkan kata masuk yang menginisialisasi sebuat *stream cipher* dan sebuah pseudo-*random number generator*.

Karena modifikasinya tersebar secara acak di seluruh koefisien *DCT*, tes $\chi$2-test tidak dapat dipakai untuk memeriksa keberadaan data tersembunyi dengan metode *Outguess*.

Gambar 3 menunjukkan kemungkinan adanya penyisipan data pada gambar contoh. Kurva yang naik-turun menunjukkan area dari gambar yang dimodifikasi menyebabkan penyimpangan dari frekuensi koefisien DCT yang diperkirakan.




**Gambar 3. Karena pemilihan bit secara acak yang dilakukan oleh *Outguess*, tidak ada penanda yang jelas mengenai data yang disisipkan.**

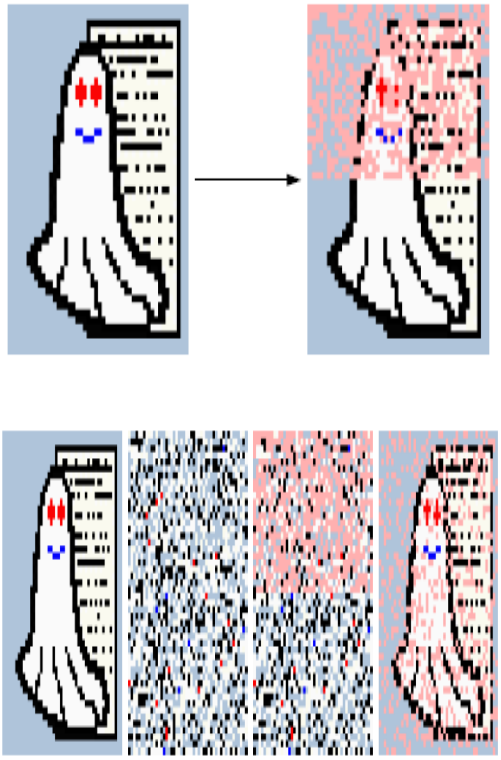
### 4.3 *F5*

*F5* adalah *Steganographic system* yang menawarkan kapasitas penyimpanan data yang besar. Dengan *F5*, penggunanya dapat memasukkan data yang berukuran sampai 13% dari ukuran gambar keseluruhan.

Selain itu *F5* menawarkan efisiensi yang sangat baik dalam perubahan bit. Efisiensinya dapat mencapai 3,7 bit untuk setiap perubahan. Dan jika *F5* diimplementasi tanpa *Matrix Encoding* efisiensinya dapat mencapai dua kali lipat.

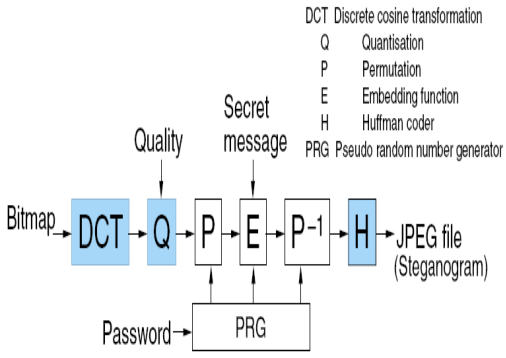
Penggunaan Matrix Encoding membuat tes $\chi$2-test tidak dapan memeriksa keberadaan dari data yang disisipkan.

Implementasi Permutative Straddling menyebabkan penyisipan data yang “terlalu” banyak menyebabkan gambar digital “rusak” secara merata. Sehingga tidak mudah dikenali dengan melihatnya.



**Gambar 4. Gambar atas contoh dari continuous embedding tanpa *Permutative Straddling* menyebabkan gambar rusak pada bagian awalnya. sedangkan Gambar bawah dengan *Permutative Straddling* pada *F5* yang menyebabkan gambar menjadi "rusak" secara merata.**

*F5* merupakan penyempurnaan dari sistem *F4*. Pada pelaksanaan *F5*, permutasi dikenakan kata lewat yang dimasukkan pengguna. Kemudian *Pseud one time pad* untuk distribusi pesan secara rata, Matrix Encoding dengan *embedding rate* yang minimal. Terakhir dilakukan *Core Embedding* seperti pada *F4*.




**Gambar 5. Implementasi Steganografi dengan sistem *F5***

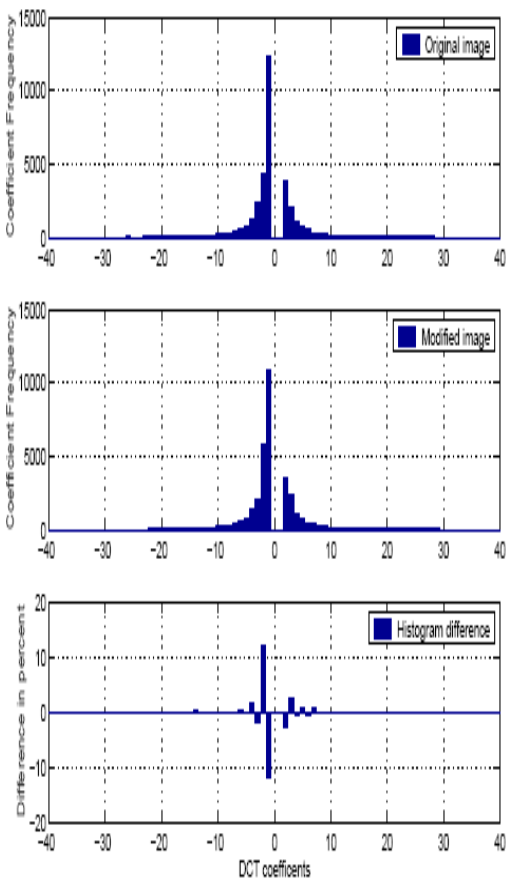
**5. Analisis Statistik**

Tes statistik dapat memeriksa apakah suatu gambar digital sudah dimodifikasi dengan steganografi atau belum. Caranya dengan memeriksa apakah keterangan statistik suatu gambar digital menyimpang dari kondisi normalnya. Beberapa tes tidak tergantung dari format data dan hanya mengukur *entropy* dari data yang berlebih.

Tes yang paling sederhana mengukur hubungan terhadap salah satu. Tes yang lebih baik adalah *"Universal Statistical Test for Random Bit Generators"* yang dibuat oleh Ueli Maurer. Kita memperkirakan Gambar Digital dengan data yang tersembunyi memiliki *entropy* yang lebih besar daripada Gambar Digital tanpa data yang tersembunyi.

Suatu tes yang sederhana tidak dapat memutuskan secara otomatis apakah suatu Gambar Digital memiliki data tersembunyi atau tidak. Westfeld dan Pfitzmann sudah meneliti bahwa data terenkripsi yang disisipkan pada gambar dengan format *GIF* mengubah histogram dari frekuensi warnanya. Satu bagian dari data yang terenkripsi sama dengan bit 1 dan bit 0. Ketika menggunakan metode bit signifikan yang paling sedikit untuk menyisipkan data terenkripsi ke dalam gambar digital yang mengandung warna yang lebih sering dipakai dari yang lain. Hasilnya, perbedaan dari frekuensi warna antara kedua warna tersebut berkurang karena penyisipan.

Kejadian yang sama berlaku untuk gambar *JPEG*. Selain mengukur frekuensi warna, kita dapat menganalisis frekuensi dari koefisien *DCT*. Gambar 6 menunjukkan contoh ketika adanya data tersembunyi yang disisipkan menyebabkan perbedaan histogram dari koefisien DCT yang terlihat.




**Gambar 6. Data tersembunyi yang disisipkan menyebabkan perubahan dari histogram koefisien *DCT*.**

Kita dapat menggunakan tes $\chi$2-test untuk menentukan apakah suatu gambar digital menunjukan gangguan karena data tersembunyi yang disisipkan. Karena tes ini hanya menggunakan *stego medium*, distribusi dari $y_i$* untuk tes $\chi$2-test harus dikalkulasikan dengan komputer dari suatu gambar digital. Jika $n_i$ adalah frekuensi dari koefisien *DCT* dari suatu gambar digital. Kita dapat beranggapan bahwa suatu gambar digital yang telah disisipi data tersembunyi memiliki frekuensi yang sama untuk kedekatan dari koefisien *DCT*.

Jadi kita dapat menghitung rata-rata aritmatika,

$$y_i^* = \frac{n_{2i} + n_{2i+1}}{2},$$

untuk menentukan distribusi yang diharapkan. Distribusi yang diharapkan dibandingkan dengan distribusi sesuai penelitian

$$y_i = n_{2i}.$$

Nilai dari $\chi$2 sebagai perbedaan dari kedua distribusi adalah

$$\chi^2 = \sum_{i=1}^{v+1} \frac{(y_i - y_i^*)^2}{y_i^*},$$

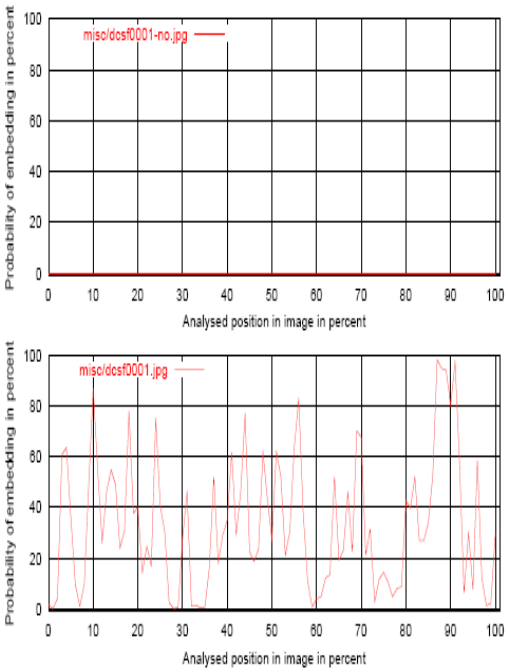
dengan v adalah derajat kebebasan. Jumlah dari katagori yang berbeda dalam histogram dikurangi satu.

Peluang dari penyisipan p diberikan dengan distribusi fungsi kumulatifsebagai berikut :

$$p = 1 - \int_0^{\chi^2} \frac{t^{(v-2)/2} e^{-t/2}}{2^{v/2}\Gamma(v/2)} dt,$$

Dengan $\Gamma$ adalah Fungsi Gamma Euler.

Kita dapat menghitung peluang dari penyisipan untuk tiap bagian yang berbeda dari gambar digital. Pemilihannya bergantung pada *steganographic system* apa yang kita coba deteksi. Untuk suatu gambar digital yang tidak mengandung data tersembunyi yang disisipkan, kita mengharapkan peluangnya bernilai nol disemua tempat. Gambar 7 menunjukkan hasil pemeriksaan pada suatu gambar digital.




**Gambar 7. Peluang dari data tersembunyi yang disisipkan dihitung berdasarkan area gambar digital yang berbeda. Gambar atas menunjukkan hasil untuk image yang tidak dimodifikasi, Gambar bawah menunjukkan hasil gambar digital dengan data tersembunyi yang disisipkan.**

## 6. Perangkat Lunak Pendeteksi

Salah satu perangkat lunak yang cukup baik adalah "*Stegdetect*". Perangkat lunak ini merupakan *utility* yang bekerja secara otomatis untuk menganalisis gambar digital dalam format *JPEG* memiliki data tersembunyi yang disisipkan atau tidak.

### 6.1 *Stegdetect*

*Stegdetect* dapat menganalisis gambar digital yang memiliki data tersembunyi yang disisipkan dengan sistem Jsteg, JPHide, dan OutGuess. Untuk setiap sistem yang mau dideteksi, kita memilih koefisisen DCT dengan tujuan dimodifikasi dan melakukan tes $\chi$2-test.

Hasil dari *Stegdetect* menampilkan list *steganographic system* ang ditemukan di setiam gambar digital, atau "negative" jika tidak ada data tersembunyi yang disisipkan dalam gambar digital yang terdeteksi oleh *Stegdetect*.

*Stegdetect* memiliki kecepatan pemeriksaan yang tinggi. *Stegdetect* memiliki kemungkinan kesalahan yang cukup besar dalam pemeriksaan jika data tersembunyi yang disisipkan berukuran kecil. Untuk sistem *Jsteg* persentase kesalahannya hanya 2%, untuk JPHide 15%-60%, sedangkan untuk Outguess mencapai 60%.

### 6.2 *Stegbreak*

Karena tes statistik hanya dapat menunjukkan peluang adanya suatu data tersembunyi yang disisipkan, *Stegdetect* tidak dapat menjamin keberadaan dari suatu data tersembunyi yang disisipkan. Untuk memeriksa sekali lagi kebenaran suatu gambar digital memiliki data tersembunyi yang disisipkan, sangat penting untuk melakuakan "pemeriksaan kamus" terhadap file dengan format *JPEG*.

Semua *Steganographic system* yang ada menyembunyikan data berdasarkan kata lewat yang dimasukkan *user*. Karena itu, kita dapat mencoba untuk menerka kata lewat untuk mengungkapkan data tersembunyi yang disisipkan pada suatu gambar digital.

Salah satu perangkat lunak untuk melakukan pemeriksaan seperti ini adalah *Stegbreak*. *Stegbreak* memeriksa setiap kemungkinan dari keta lewat sesuai dengan keinginan user.

*Stegbreak* memiliki perbedaan kecepatan pemeriksaan yang cukup besar untuk setiap sistem. Dengan 1200MHz prosesor Pentium III, *Stegbreak* melakukan pemeriksaan sampai 112.000 kata per detik untuk sistem Jsteg, menurun menjadi 47.000 kata per detik untuk sistem Outguess. *Stegbreak* hanya dapat melakukan pemeriksaan dengan kecepatan 15.000 kata per detik saja untuk sistem JPHide. Suatu tes dengan 300 gambar digital dan kata yang akan dicoba sebanyak 557.000 kata memakan waktu smencapai 10 hari untuk JPHide, 3 hari untuk Outguess, dan hanya beberapa jam saja untuk Jsteg.

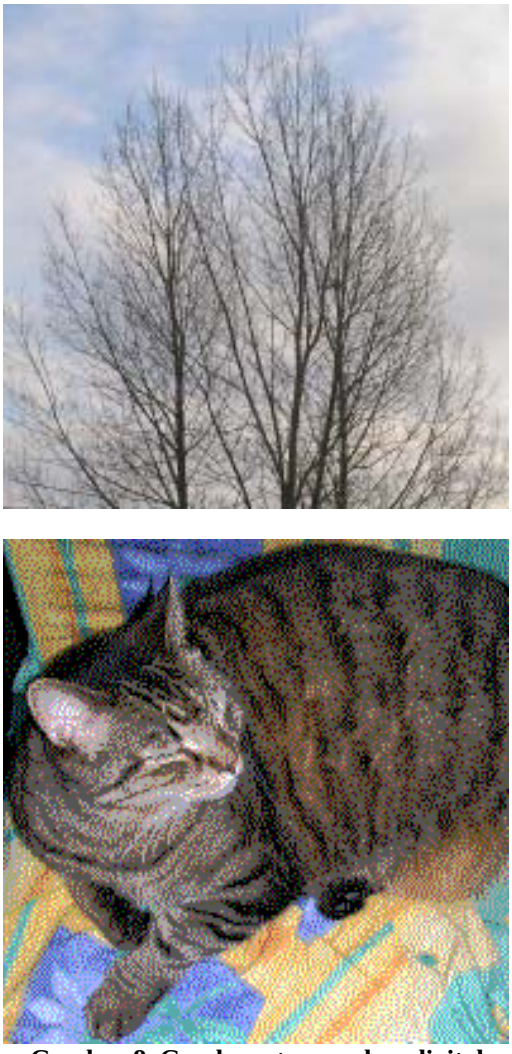
Melihat waktu yang dihabiskan oleh *Stegbreak*, sangat dianjurkan kata yang dipakai untuk mencoba berdasarkan kamus, atau biasa disebut dengan *distributed dictionary attack*.

Cara seperti ini dapat membantu mengurangi waktu kerja cukup lama. Kekurangan dari *distributed dictionary attack* adalah ketidakmampuan untuk menangani kata lewat

yang rumit. *distributed dictionary attack* biasanya ditujukan untuk *weak password*.

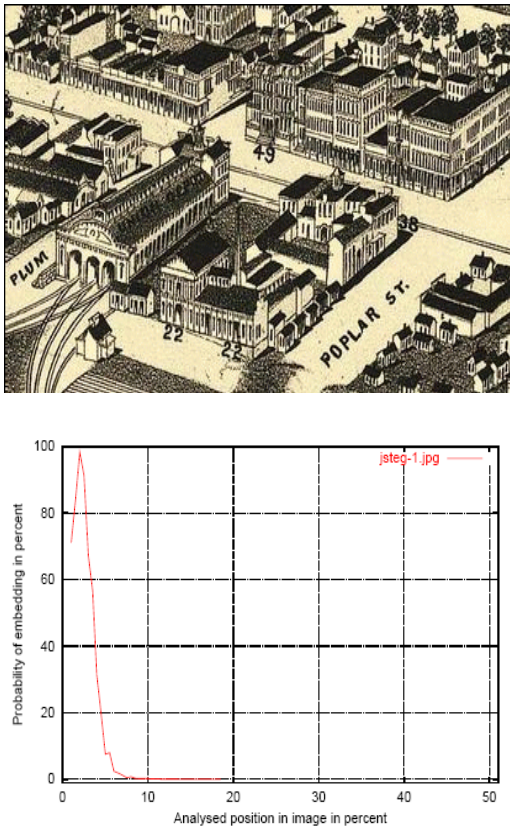
### 7. Contoh Aplikasi Steganografi

Gambar 8 menunjukkan perubahan suatu gambar dari gambar atas ke gambar bawah. Perubahan ini dapat dilakukan dengan semua kecuali 2 bit terakhir dari tiap komponen warna, kemudian membuat tingkat cahayanya 85 kali lipat.



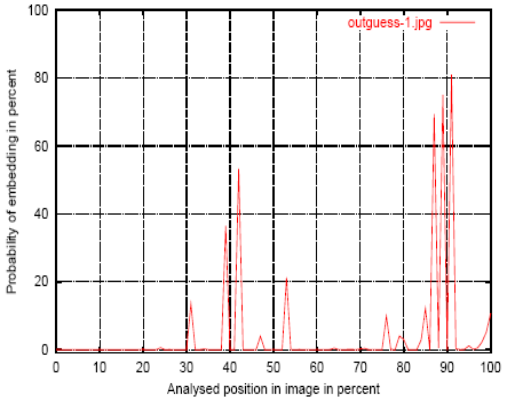
**Gambar 8. Gambar atas gambar digital pembawa data yang disisipkan, Gambar bawah setelah diekstrak dari gambar atas menunjukkan hasilnya.**

Gambar 9 menujukkan suatu gambar dan hasil tes $\chi$2-test yang menunjukkan gambar digital tersebut memiliki data tersembunyi yang disisipkan. Karena peluang penyisipan jelas ada pada bagian awal, maka dapat disimpulkan *steganographic system* yang digunakan adalah *Jsteg*.




**Gambar 9. *Stegdetect* menunjukkan kemungkinan gambar atas memiliki data tersembunyi yang disisipkan dengan sistem *Jsteg***

Gambar 10 menujukkan suatu gambar dan hasil tes $\chi$2-test yang menunjukkan gambar digital tersebut memiliki data tersembunyi yang disisipkan. Karena peluang penyisipan ada tersebarpada seluruh bagian, maka dapat disimpulkan kemungkinan besar *steganographic system* yang digunakan adalah Outguess.




**Gambar 10. *Stegdetect* menunjukkan kemungkinan gambar atas memiliki data tersembunyi yang disisipkan dengan sistem *Outguess***

## 8. Kesimpulan

Kesimpulan yang dapat diambil dari Studi dan Deteksi Steganografi pada file bertipe *JPEG* dengan tiga *Steganographic System* ini adalah :

1. Steganografi dapat digunakan untuk komunikasi rahasia tanpa mencurigakan karena media penyimpanannya berupa gambar digital yang masih dapat dilihat dengan mata tanpa ada suatu kejanggalan.
2. *Jsteg* merupakan *steganographic system* yang paling mudah diketahui penggunaannya dalam suatu gambar digital. Data yang disisipkan dengan sistem ini selalu disimpan pada awal gambar digital.
3. *Outguess* lebih sulit dilacak daripada *Jsteg*. Data yang disisipkan dengan sistem *Outguess* disimpan secara merata diseluruh bagian dari gambar digital.
4. Sistem *F5* menawarkan kapasitas penyimpanan data yang besar, efisiensi yang besar, *Matrix Encoding* yang lebih aman dan *Permutative Straddling* yang membuat *F5* menjadi sistem yang dapat diunggulkan dibandingkan dengan kedua sistem yang lainnya.
5. Penggunaan Analisis Statistik dapat menentukan suatu gambar digital memiliki data tersembunyi yang disisipkan atau tidak. Analisis Statistik adalah cara yang cukup sederhana dan membantu dalam pemeriksaan. Pemakaian Analisis Statistik hanya untuk menentukan apakah suatu image memiliki data tersembunyi yang disisipkan atau tidak.

**DAFTAR PUSTAKA**


[1] Fridrich Jessica. Breaking the Outguess. *wwwiti.cs.unimagdeburg.de/itiamsl/acm/acm02/outguess.pdf*
Tanggal akses: 3 Januari 2007 pukul 15:00

[2] Provos Niels. (2001). Detecting Steganographic Content on the Internet. *http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf*
Tanggal akses: 2 Januari 2007 pukul 20:00

[3] Provos Niels. (1998) OutGuess – Universal Steganography. *http://www.outguess.org/*
Tanggal akses: 3 Januari 2007 pukul 16:00

[4] Westfeld Andreas. F5 – a Steganographic Algorithm *os.inf.tudresden.de/~westfeld/publikationen/f5.pdf.*
Tanggal akses: 3 Januari 2007 pukul 15:00.

[5] Wikipedia. Steganography. *en.wikipedia.org/wiki/Steganography*
Tanggal akses: 3 Januari 2007 pukul 15:00