Steganalisis Khusus dengan Pendekatan Subjektif dan Statistik pada *Stego Image*

Maria Helena Iwo – NIM: 13503088

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

Email: if13088@students.if.itb.ac.id

Abstrak

Steganografi merupakan teknik dan seni menyembunyikan pesan rahasia di dalam pesan lain atau suatu media sehingga keberadaan pesan tersebut tidak dapat diketahui. Teknik penyisipan pesan ke dalam suatu media dapat dilakukan dalam dua macam ranah, yaitu ranah spasial dan ranah *transform*. Pada ranah spasial, nilai *byte* dari media atau *covertext* dimodifikasi secara langsung. Contoh metode yang tergolong dalam ranah ini adalah LSB (*Least Significant Bit*). Sedangkan pada ranah *transform*, hasil tranformasi frekuensi sinyal dimodifikasi secara langsung. Contoh metode yang tergolong dalam ranah ini adalah *spread spectrum*. Pendeteksian akan adanya suatu pesan tersembunyi pada sebuah media merupakan bidang steganalisis. Steganalisis merupakan teknik dan seni untuk mendeteksi serta mengekstrak pesan tersembunyi yang ada pada suatu media. Media tersebut dapat berupa teks, gambar, dan video. Dari berbagai media yang ada, *steganogram* paling banyak dijumpai pada media berupa gambar dengan berbagai format, misalnya JPEG, GIF, BMP, dan PNG. Salah satu cara untuk melakukan steganalisis adalah dengan pendekatan statistik dan pendekatan subjektif. Steganalisis menjadi lebih mudah apabila teknik atau algoritma untuk menyembunyikan pesan diketahui. Hal ini disebut dengan steganalisis khusus. Pada makalah ini akan dibahas steganalisis khusus dengan pendekatan statistik dan subjektif untuk mendeteksi ada tidaknya pesan tersembunyi pada media gambar, dimana teknik penyisipan pesan tergolong dalam ranah spasial. Kakas-kakas steganografi yang menjadi sasaran steganalisis berupa kakas Eztego dan S-Tools. Metode statistik yang dibahas pada makalah ini meliputi metode *chi-square* dan metode RS Analysis. Sedangkan metode subjektif yang dibahas adalah metode *visual*.

Kata Kunci: steganografi, steganalisis, LSB (*Least Significant Bit*), *spread spectrum,* Eztego, S-Tools, *chi-square*, *RS analysis*.

---

1. Pendahuluan

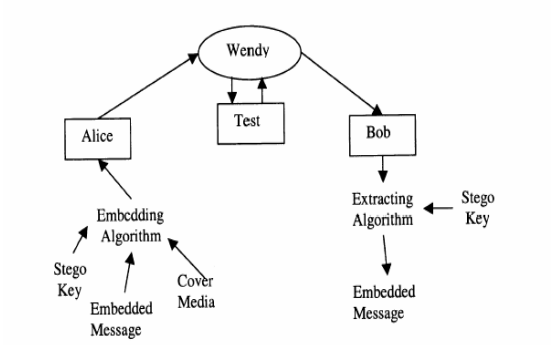
Steganografi berasal dari bahasa Yunani, yaitu ”*steganos*”, yang berarti ”tulisan tersembunyi”. Steganogafi sangat kontras dengan kriptografi. Jika kriptografi merahasiakan makna pesan sementara eksistensi pesan tetap ada, maka steganografi menutupi keberadaan pesan [4]. Dengan demikian, tujuan steganografi adalah mentransmisikan pesan rahasia dengan cara disisipkan pada sebuah media penampung sedemikian rupa sehingga tidak ada pihak ketiga yang sadar akan keberadaan pesan rahasia tersebut. Akibatnya, jika dibandingkan dengan kriptografi, steganografi memiliki keuntungan bahwa pesan yang dikirimkan tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbulkan kecurigaan bagi pihak ketiga.

Terdapat beberapa istilah yang berkaitan dengan steganografi [4], sebagai berikut:

a. *Hiddentext* atau *embedded message*: pesan rahasia yang disembunyikan.
b. *Covertext* atau *cover-object*: pesan yang digunakan untuk menyembunyikan *embedded message*.
c. *Stegotext* atau *stego-object*: pesan yang sudah berisi *embedded message*.

Di dalam steganografi dijital, baik *hiddentext* maupun *covertext* dapat berupa teks, citra, audio, maupun video, yang dapat direpresentasikan sebagai *bit stream.* Jika *cover-object* yang dipilih adalah gambar, *stego-object*-nya dapat juga disebut *stego-image*. Pada kebanyakan makalah, penyebutan s*tego-object* digantikan oleh steganogram. Kedua istilah ini mempunyai arti yang sama. Dalam prakteknya, media penampung yang paling banyak digunakan adalah arsip gambar dijital.

Kerangka kerja steganografi biasanya dijelaskan dengan menggunakan permasalahan klasik tahanan penjara (*prisoner's problem*) [7]. Permasalahan tahanan penjara terjadi ketika dua orang tahanan penjara, yakni Alice dan Bob ingin berkomunikasi untuk merencanakan lari dari penjara. Rencana tersebut tentu saja tidak boleh diketahui oleh pihak ketiga, yang dalam hal ini adalah penjaga penjara, yaitu Wendy. Model umum dari permasalahan steganografi tersebut dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Kerja Steganografi

Pada Gambar 1, Alice ingin menyampaikan pesan rahasia kepada Bob tanpa disadari oleh Wendy. Untuk melakukan hal tersebut, Alice memilih suatu *cover-object* yang tidak akan dicurigai oleh Wendy sebagai tempat untuk menyisipkan pesan rahasianya. Proses menyisipkan pesan bergantung pada *kunci stego*, yaitu informasi rahasia tambahan, misalnya *password*. Proses untuk menyisipkan pesan rahasia dapat dinyatakan sebagai berikut:

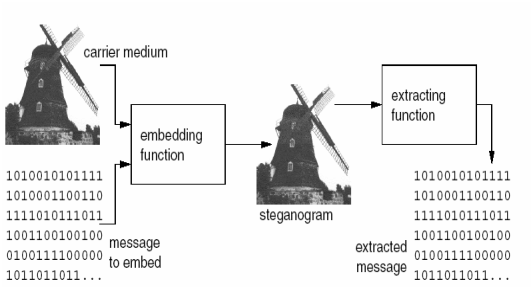
cover-object + embedded message + stego key= stego image

Misalkan, pesan yang ingin dikirim adalah "`Lari jam satu`" dan media penampung yang digunakan adalah teks. Maka, pesan tersebut disembunyikan dalam tulisan lain dengan cara menyisipkan setiap huruf pesan rahasia pada awal setiap kata. sebagai berikut [4]:

"`Lupakan asal rumor itu, jaga agar matamu sehat atau turunkan ubanmu`"

Dengan pesan di atas, Wendy, si penjaga penjara tidak akan curigra dan mengganggap Alice sedang bercanda dengan Bob. Dengan demikian, sistem steganografi dikatakan aman jika pihak ketiga, yaitu Wendy tidak dapat membedakan *covertext* dan *stegotext*.

Dalam steganografi dijital, dimana data direpresentasikan dalam aliran bit, sistem steganografi dapat dimodelkan seperti pada Gambar 2. Pada beberapa sistem steganografi, biasanya ditambahkan kunci (*stego-key*) untuk melakukan penyisipan dan ekstraksi.



Gambar 2. Sistem Steganografi Dijital

Gambar 3.a dan Gambar 3.b memperlihatkan *cover image* dan *stego imag*e yang tidak dapat dibedakan secara visual, walaupun Gambar 3.b mengandung pesan rahasia yang disisipkan secara pada bagian atas gambar. Selanjutnya, Gambar 4.a dan Gambar 4.b menunjukkan LSB (*Least Significant Bit*) dari Gambar 3.a dan 3.b. Pada Gambar 4.a dan Gambar 4.b, warna hitam merepresentasikan LSB=0 dan warna putih merepresentasikan LSB=1.

Meskipun terdapat banyak teknik yang dalam melakukan penyisipan pesan agar sulit dideteksi, masih dimungkinkan untuk mendeteksi keberadaan pesan tersembunyi tersebut. Ilmu dan seni untuk mendeteksi keberadaan pesan rahasia inilah yang disebut dengan steganalisis. Cara mendeteksi adanya pesan rahasia dapat melalui inspeksi secara *visual*, *audible,* analisis statistik, dan analisis struktural.

Gambar 3.a. *Cover Image*

Gambar 3.b. *Stego image*

Inspeksi secara *visual* dapat dikenakan pada *steganogram* yang berupa arsip gambar. Sedangkan inspeksi secara *audible* dapat dikenakan pada *steganogram* yang berupa arsip suara. Selain itu, analisis statistika dapat dilakukan melalui metode *chi-square*, *RS analysis*, dan *pairs analysis*. Cara deteksi yang terakhir, yaitu analisis struktur dapat dilakukan dengan membandingkan s*teganogram* dan *cover medium* asal. Perbandingannya dapat dilihat dari beberapa atribut, yaitu ukuran arsip, tanggal dan waktu modifikasi, isi arsip, serta *checksum* [5]. Namun analisis struktur ini tidak aplikatif karena membutuhkan baik *steganogram* maupun *cover medium*-nya dalam proses deteksi.

Gambar 4.a LSB dari *Cover Image*

Gambar 4.b. LSB dari *Stego Image*

Metode-metode dalam steganalisis dapat dikelompokan menjadi dua kategori, yaitu deteksi khusus dan deteksi umum. Deteksi khusus berarti sistem steganografi yang digunakan telah diketahui sebelumnya. Sedangkan deteksi umum berarti sistem steganografi yang digunakan tidak diketahui sebelumnya.

Baik steganalisis khusus maupun steganalisis umum, dapat menggunakan pendekatan statistik, yaitu berupa fungsi-fungsi yang dirancang untuk membedakan spesifikasi statistik antara *cover-object* dan *stego object.* Hasil dari fungsi ini kemudian akan dibandingkan dengan nilai batas

tertentu (*treshold*) untuk menentukan ada tidaknya pesan rahasia dalam *stego object.*

Sementara itu, pendeteksian steganalisis umum biasanya menggunakan pendekatan intelejensia buatan. Dengan pendekatan ini, dilakukan pembelajaran terlebih dahulu terhadap sejumlah *stego image*. Teknik dalam bidang intelejensia buatan yang dapat diadopsi untuk melakukan steganalisis meliputi jaringan saraf tiruan dan *support vector machine*.

2. Teknik Penyisipan Pesan

Teknik penyisipan pesan ke dalam *cover-object* secara dapat dilakukan dalam dua macam ranah, yaitu, ranah spasial dan ranah *transform*. Salah satu teknik steganografi yang terkenal dalam ranah spasial adalah penyisipan LSB (*Least Significant Bit*). Sedangkan pada ranah *transform,* teknik steganografi yang sering digunakan adalah *spread spectrum*. Pada ranah transform, kebanyakan teknik penyisipan menggunakan DCT (*Discrete Cosine Transform*). DCT ini digunakan oleh arsip gambar dengan format JPEG untuk meningkatkan performansi kompresi arsip.

Pada makalah ini hanya akan dibahas teknik penyisipan pesan secara spasial karena metode steganalisis yang dibahas pada makalah ini ditujukan untuk penyisipan pesan secara spasial.

Metode penyisipan pesan yang tergolong dalam ranah spasial adalah metode LSB. Untuk menjelaskan metode ini, digunakan citra dijital sebagai *cover-object.* Setiap piksel dalam citra dijital berukuran 1 sampai 3 *byte.* Pada susunan bit di dalam *byte (*1 *byte*=8 bit*),* terdapat bit yang paling kurang berarti (*Least Significant Bit* atau LSB). Misalnya pada *byte* 11000001**1**, bit LSB-nya adalah 1. Untuk melakukan penyisipan pesan, bit yang paling cocok untuk diganti dengan bit pesan adalah bit LSB, sebab pegubahan bit tersebut hanya akan mengubah nilai *byte*-nya menjadi satu lebih tinggi atau satu lebih rendah [3].

Sebagai contoh, urutan bit berikut ini menggambarkan 3 piksel pada *cover image* 24-bit.

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

Pesan yang akan disisipkan adalah karakter "A", yang nilai binari-nya adalah 10000001, maka akan dihasilkan *stego image* dengan urutan bit sebagai berikut:

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

Penyisipan LSB ini bisa dilakukan secara sekuensial maupun acak. Jika dilakukan secara acak, maka posisi LSB yang diganti menjadi kunci stegano.

3. Pendekatan statistik pada steganalisis

Steganalisis khusus terdiri dari metode subjektif dan metode statistik [7]. Metode subjektif memanfaatkan indera penglihatan manusia untuk mengamati bagian gambar yang dicurigai. Sedangkan metode statistik melibatkan analisis matematis terhadap sebuah gambar untuk menemukan perbedaan antara gambar asal dan *stego image.*

Salah satu metode statistik untuk steganalisis khusus yang didasarkan pada analisis statistik dari pasangan nilai sampel dikenal dengan nama *chi-square attack. Chi-square attack* terbukti handal dalam mendeteksi pesan rahasia yang disisipkan secara sekuensial. Namun, *chi-square attack* gagal untuk menemukan pesan rahasia yang disisipi secara acak.

Metode lainnya adalah *RS Analysis* yang terbukti handal dan akurat dalam mendeteksi pesan rahasia yang disisipkan secara acak. Pada *RS Analysis*, sebuah gambar dipartisi menjadi grup-grup *regular* atau grup-grup *singular*, tergantung pada derajat *noisy* dari sebuah grup. Noisy dapat diartikan sebagai kerusakan yang terjadi pada data. Proporsi grup *regular* dan grup singular nantinya akan membentuk sebuah kurva parabola, yang disebut juga diagram RS. *RS Analysis* ini menggunakan beberapa asumsi dasar, yang jika asumsi tersebut dipenuhi, maka panjang pesan rahasia dapat diestimasi dengan akurat.

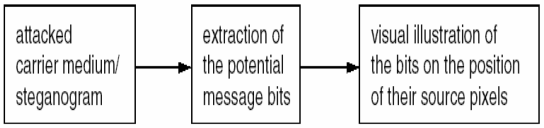
Berikut ini akan dibahas metode-metode steganalisis dengan pendekatan ssubjektif dan statistik.

a. ***Visual Attack***

Serangan visual merupakan serangan terhadap teknik steganografi dengan memanfaatkan indera penglihatan manusia untuk menginspeksi kerusakan-kerusakan pada gambar yang terjadi akibat penyisipan. Ide dari *visual attack* adalah membuang semua bagian gambar yang menampung pesan rahasia.

Pada v*isual attack,* terdapat proses penyaringan (*filtering*) yang bertujuan untuk menghilangkan bagian pesan rahasia dari *stego image* [8]. Fungsi yang digunakan pada penyaringan ini bergantung dari sistem steganografi yang digunakan. Struktur umum dari fungsi penyaringan dapat dilihat pada Gambar 5.

Setelah melalui proses penyaringan, maka citra pada bagian gambar yang tidak disisipi pesan akan mendekati bagian gambar semula. Sedangkan bagian gambar yang mengandung pesan rahasia akan menjadi ”rusak” setelah disaring. Dengan demikian, dari gambar yang dihasilkan setelah penyaringan, mata manusia dapat dengan mudah membedakan apakah pada gambar tersebut terdapat pesan rahasia atau tidak.



Gambar 5. Struktur Fungsi Penyaringan

Setelah melalui proses penyaringan, maka citra pada bagian gambar yang tidak disisipi pesan akan mendekati bagian gambar semula. Sedangkan bagian gambar yang mengandung pesan rahasia akan menjadi ”rusak” setelah disaring. Dengan demikian, dari gambar yang dihasilkan setelah penyaringan, mata manusia dapat dengan mudah membedakan apakah pada gambar tersebut terdapat pesan rahasia atau tidak.

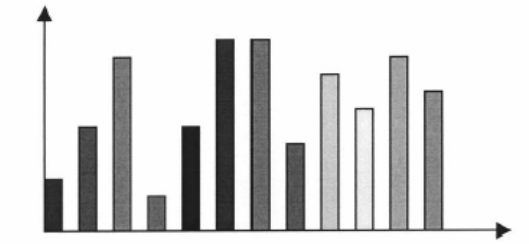
Visual attack ini bekerja dengan baik pada kakas Eztego. Metode ini sangat sederhana, namun sulit untuk dilakukan secara otomatis dan *reliability*-nya masih sangat dipertanyakan.

b. ***Chi-Square Attack***

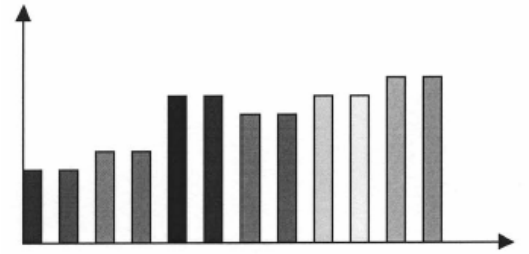
Pfitzman dan Westfeld [8] memperkenalkan steganalisis dengan pendekatan statititstik yang handal. Metode steganalisis tersebut didasarkan pada analisis histogram dari *Pairs of Values* (PoVs) yang ditukar selama proses penyisipan pesan. PoVs dapat berupa pasangan nilai LSB pada piksel, koefisien DCT, indeks palet sebelum dan sesudah digantikan dengan bit pesan.

Jika bit-bit pesan rahasia didistribusikan secara merata, maka kemunculan pasangan nilai PoVs menjadi sama. Dengan kata lain, distribusi kemunculan kedua nilai dari setiap pasangan akan cenderung sama setelah proses penyisipan pesan (hal ini tergantung dari panjang pesan). Histogram dengan enam warna pada Gambar 6 dan Gambar 7 mengilustrasikan kecenderungan tersebut.

Ide dari *chi-square attack* adalah menguji seberapa signifikan kemunculan kedua nilai di setiap pasangan adalah sama. Hal ini berarti membandingkan distribusi frekuensi yang diharapkan secara teori (setelah menerapkan algoritma steganografi pada *cover image*) dan distribusi frekuensi dari beberapa sampel yang diamati pada *stego image*.



Gambar 6. Histogram Warna Sebelum Penyisipan



Gambar 7. Histogram Warna Setelah Penyisipan

Hal penting dalam merancang penyerangan ini adalah menentukan distribusi frekuensi yang diharapkan secara teori. Distribusi

frekuensi yang diharapkan tidak dapat dihitung dari *stego image*. Akan tetapi pada kebanyakan kasus, penyerang tidak memiliki *cover image.* Hal ini dapat diatasi sebagai berikut. Pada gambar asal, distribusi frekuensi yang diharapkan secara teori adalah rata-rata dari dua frekuensi dalam PoV. Karena pertukaran sebuah nilai menjadi nilai yang lain tidak mengubah jumlah kedua nilai tersebut, maka rata-rata dari kedua frekuensi tersebut tidak berubah baik untuk gambar asal dan *stego image*. Fakta inilah yang mendasari penentuan frekuensi yang diharapkan dari *stego image.*

Ketika distribusi sampel yang diamati dan ditribusi frekuensi secara teori ditentukan, maka pengujian dengan *chi-square* dapat diterapkan untuk menentukan derajat kesamaan antara distribusi sampel dan distribusi frekuensi yang diharapkan. Proses kerja *chi-square attack* adalah sebagai berikut:

1. Misalkan terdapat *k* kategori dan terdapat sebuah sampel acak dari hasil obeservasi. Tiap-tiap observasi harus dimasukkan ke dalam satu kategori. Sebagai contoh, untuk sebuah palet gambar, terdapat paling banyak 256 warna c_i pada palet, artinya terdapat maksimal 128 PoV sehingga k=128.

2. Frekuensi yang diharapkan pada kategori ke *i*, dimana *i=1,2,3,...,k* setelah penyisipan bit pesan yang terdistribusi merata, dapat dihitung dengan Persamaan 1.

Persamaan (1)

3. Frekuensi aktual dari sampel dihitung dengan Persamaan 2:

   n_i' = jumlah indeks ke c_{2i}

   Persamaan (2)

4. Nilai chi-square dihitung dengan Persamaan 3:

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n_i')^2}{n_i'}$$

   Persamaan (3)

   dengan derajat kebebasan= k-1

5. Kemungkinan distribusi n_i' dan n_i adalah sama dinyatakan dengan *p* pada Persamaan 4.

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma(\frac{k-1}{2})} \int_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx$$

Persamaan (4)

Jika distribusi n_i' sama dengan $n_{i,}$ maka X_{k-1}^2 akan mendekati 0. Dengan demikian nilai *p* akan mendekati 1.

Chi-square attack hanya cocok untuk penyisipan sekuensial. Akan tetapi, berbagai teknik deteksi yang menggunakan analisis histogram akan sangat mudah dihindari.

3. RS Analysis

RS Analysis dikemukakan oleh Fridrich et al.[1]. Teknik ini memanfaatkan korelasi spasial pada *stego image*. RS Analysis dapat mendeteksi penyisipan secara random dengan akurat. Ide dasarnya adalah menemukan dan menghitung hubungan yang lemah antara kelompok LSB dan *stego image* itu sendiri.

c.1. Terminologi

Diberikan gambar dengan ukuran *MXN* dimana nilai pikselnya merupakan himpunan *P*. Gambar tersebut kemudian dipartisi menjadi kelompok-kelompok *n* piksel yang bertetangga $(x_1,...,x_n)$ sepanjang baris atau kolom. Untuk mendapatkan korelasi spasial, digunakan fungsi diskriminasi f , dimana f merupakan nilai absolut rata-rata dari perbedaan antara piksel-piksel yang bertetangga. Secara matematis, fungsi diskriminasi f dinyatakan dalam Persamaan 5.

$$f(x_1,\ldots,x_n) = \frac{1}{n-1}\sum_{i=1}^{n}\left|x_{i+1} - x_i\right|$$

Persamaan (5)

Fungsi f ini mengukur regularitas dari kelompok piksel K=$(x_1,...,x_n)$. Jika suatu

kelompok semakin *noisy*, maka semakin besar nilai yang dihasilkan oleh fungsi *f*.

Penyisipan LSB meningkatkan *noisy* pada gambar, sehingga nilai yang dihasilkan oleh fungsi diskriminasi *f* akan meningkat setelah membalikkan LSB dari sekumpulan piksel yang tetap di dalam tiap-tiap kelompok. Penyisipan dengan LSB dapat dideskripsikan dengan menggunakan fungsi *flipping* F_1 dan fungsi *dual flipping* F_{-1}.

Contoh:

F_1: $0 \leftrightarrow 1, 2 \leftrightarrow 3, ..., 254 \leftrightarrow 255$
F_{-1}:$-1 \leftrightarrow 0, 1 \leftrightarrow 2, ..., 255 \leftrightarrow 256$

$$F_0: \; F_0(x) = x \quad \forall x \in P$$

Persamaan (6)

Selanjutnya kelompok piksel *K* dapat diklasifikasikan ke dalam tiga tipe berbeda, yaitu R,S,dan U. Pengklasifikasian ini bergantung pada bagaimana operasi *flipping* mengubah nilai yang dihasilkan oleh fungsi diskriminasi *f*. Secara formal, ketiga tipe tersebut dapat dinyatakan sebagai berikut:

- Kelompok *Regular* $R \leftrightarrow f(F(K)) > f(K)$
- Kelompok *Singular* $S \leftrightarrow f(F(K)) < f(K)$
- Kelompok *Unchanged* $U \leftrightarrow f(F(K)) = f(K)$

dimana $F(K)$ merupakan fungsi *flipping* untuk tiap-tiap komponen kelompok $K=(x1,..,x3)$.

Agar sebuah kelompok diklasifikasikan ke dalam kelompok *regular*, maka piksel yang mengandung *noisy* dalam kelompok tersebut harus ditambahkan setelah operasi *flipping* dilakukan. Hal yang sama juga tejadi pada kelompok *singular*, dimana piksel yang mengandung *noisy* berkurang setelah operasi *flipping* dilakukan.

Secara umum, operasi *flipping* yang berbeda diaplikasikan pada piksel-piksel yang berbeda dalam kelompok *K*. Pola piksel-piksel untuk di-*flip* disebut "*mask*". "*Mask*" terdiri dari nilai-nilai 0 dan 1. "M*ask*" berukuran sama dengan ukuran kelompok yang akan di-*flip*. Hal ini dimaksudkan unruk menentukan berapa banyak piksel di setiap kelompok yang telah di-*flip*. Oleh karena itu, kelompok $F(G)$ yang telah di-*flip* didefinisikan sebagai $(F_{M(1)}(x_1), ..., F_{M(n)}(x_n))$, dimana $M(_i)$ dengan $i=1,2,..,n$ adalah elemen dari "*mask*" *M* dan nilainya dapat berupa -1,0,1. Sebagai contoh jika $M = (0,1,1,0)$ maka $F(G)= F_0(x_1), F_1(x_2), F_1(x_3), F_0(x_4)$.

c.2. Prinsip RS Analysis

Karena *flipping* pada LSB mengakibatkan piksel yang mengandung *noisy* bertambah, maka nilai yang dihasilkan oleh fungsi diskriminasi *f* akan meningkat. Akibatnya, jumlah kelompok *regular* akan melebihi jumlah kelompok *singular*. Misalkan R_M dan S_M merupakan nilai relatif dari kelompok regular dan kelompok singular untuk "*mask*" yang tidak negatif. Maka hipotesis bahwa tidak ada pesan rahasia adalah benar untuk suatu *cover image*, dimana

$R_M \cong R_{-M}$ dan
$S_M \cong S_{-M}$

yang berarti nilai R_M hampir sama dengan R_{-M} jika tidak ada pesan rahasia. Demikian juga pada hubungan antara S_{+M} dan $S_{-M..}$

Akan tetapi, asumsi di atas tidak bisa digunakan jika bidang LSB acak. R_M dan S_M akan saling seiring dengan bertambahnya panjang pesan yang disisipkan. Perbedaan R_M dan S_M akan mendekati nol jika LSB dari 50% piksel di-*flip*. Pada kasus ini, R_M dan S_M mempunyai hibungan

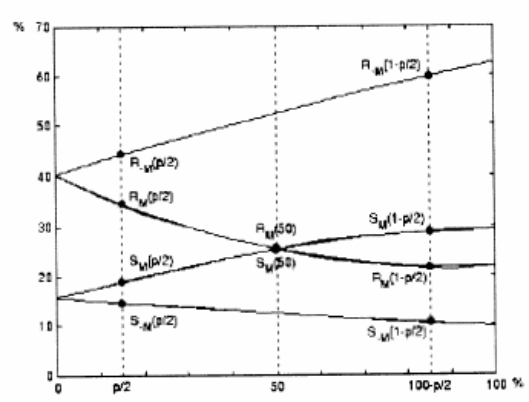
: $R_M \cong S_M$.

Secara mengejutkan pengacakan LSB mengakibatkan perbedaan antara R_{-M} dan S_{-M} I meningkat sering dengan panjang pesan.

Percobaan [3] menunjukkan bahwa R_{-M} dan S_{-M} mempunyai hubungan aproksimasi linear sesuai dengan jumlah piksel dan LSB yang di-*flip*. Karena itu, kurva R_{-M} dan S_{-M} berbentuk garis lurus, sedangkan kurva R_M dan S_M berbentuk parabola.

Gambar 8 memperlihatkan R_{-M},S_M , R_{-M} dan S_{-M} sebagai fungsi dari jumlah piksel dengan LSB yang telah di-*flip*. Graf pada Gambar 8 ini disebut diagram RS dimana

sumbu-x (absis) merepresentasikan persentasi piksel dengan LSB yang di-*flip*, sedangkan sumbu-y (ordinat) merepresentasikan jumlah relatif dari kelompok regular dan kelompok singular dengan "*mask*" M dan –M,M = [0,1,1,0].



Gambar 8. Diagam RS

Misalkan panjang pesan pada *stego image* adalah *p* (dalam persen piksel). Titik $R_M(p/2)$, dan $S_M(p/2)$, berkoresponden dengan jumlah kelompok R dan kelompok S untuk mask *M* yang tidak negatif. Hal yang sama juga terjadi dimana diperoleh titik $R_{-M}(p/2)$ dan $S_{-M}(p/2)$ untuk *mask M* yang negatif. Alasan keempat koordinat dari empat titik ini hanya satu setengah dari panjang pesan p adalah rata-rata kemungkinan bahwa sebuah piksel akan di-*flip* adalah 0.5 jika pesan tersembunyi diasumsikan berupa *bit-stream* yang acak.

Empat titik lainnya, yaitu $R_M(1-p/2)$, dan $S_M(1-p/2)$, $R_{-M}(1-p/2)$, dan $S_{-M}(1-p/2)$, didapatkan dari operasi *flipping* LSB dari semua piksel pada *stego image*. Titik tengah $R_{-M}(p/2)$, $R_M(1-p/2)$ dan $S_{-M}(p/2)$, $S_{-M}(1-p/2)$ mendefinisikan dua buah garis lurus. Sedangkan titik-titik $R_M(p/2)$, $R_M(1/2)$, $R_M(1-p/2)$, dan $S_M(p/2)$, $S_M(1/2)$, $S_M(1-p/2)$ menghasilkan dua buah parabola.

Dengan *RS Analysis*, dimungkinkankan juga mendapatkan sebuah formula untuk menghitung panjang pesan, jika kedua asumsi dibawah ini dipenuhi.

a. Kurva R_M dan R_{-M} saling beririsan pada koordinat x yang sama dengan kurva S_M dan S_{-M}.

b. Titik perpotongan dari kurva R_M dan S_M memiliki koordinat x yang sama dengan 50%, sehingga $R_M(1/2) = S_M(1/2)$.

Untuk menurunkan formula, sumbu-x mula-mula di disesuaikan ukurannya sehingga $p/2$ menjadi 0 dan $100-p/2$ menjadi 1. Koordinat x dari titik perpotongan kemudian menjadi akar dari persamaan kuadrat di bawah ini:

$$2(d_1+d_0)x^2+(d_{-0}-d_{-1}-d_1-3d_0)x+d_0-d_{-0}=0$$

Persamaan (7)

dimana
$d_0 = R_M(p/2) - S_M(p/2)$
$d_1 = R_M(1-p/2) - S_M(1-p/2)$
$d_{-0} = R_{-M}(p/2) - S_{-M}(p/2)$
$d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2)$

Dengan demikian panjang pesan *p* dapat dihitung dengan Persamaan 8.

$$p = x/(x-1/2)$$

Persamaan (8)

c.3 Akurasi *RS Analysis*

Ada beberapa faktor yang mempengaruhi akurasi dari pendeteksian dengan menggunakan metode *RS Analysis*, yaitu bias awal, pemilihan "*mask*", level *noisy cover image*, dan penempatan bit-bit pesan dalam gambar.

Bias awal merupakan estimasi panjang pesan ketika tidak ada pesan yang disembunyikan secara aktual. Secara teori, panjang pesan rahasia seharusnya bernilai 0 unruk *cover image* asal (yang belum disisipi pesan). Akan tetapi *RS Analysis* mungkin akan menunjukkan suatu bias awal yang tidak 0 dan bisa berupa berupa bilangan positif ataupun negatif. Hal ini membatasi keakuratan *RS Analysis* secara teori. Gambar berukuran kecil memiliki jumlah kelompok *regular* dan *singular* yang kecil dan karena itu, dapat memiliki variasi yang relatif lebih tinggi pada bias awal. Sebaliknya, bias biasanya bernilai kecil untuk gambar dengan format JPEG maupun gambar-gambar yang berasal dari kamera dijital dan tidak dikompres. Secara umum, gambar bewarna memiliki variasi bias awal yang lebih tinggi daripada gambar tidak berwarna.

RS Analysis sangat bergantung pada bagaimana proses partisi gambar menjadi kelompok-kelompok. *Mask* menentukan pola piksel-piksel dalam kelompok untuk di-*flip*, dan karena itu berefek pada efektivitas metode deteksi. Pada percobaan yang telah dilakukan [1], berbagai *mask* telah dicobakan untuk menginvenstigasi kehandalannya dalam melakukan steganalisis terhadap 10000 gambar berformat JPEG. Dari 10000 gambar tersebut, sebanyak 5% merupakan *stego image* dengan penyisipan LSB. *Mask-mask* yang digunakan meliputi *mask* linear dan *mask* matriks. Mask linear meliputi [0,1].[0,1,0],[0,1,1,0],[0,1,1,1,0], [0,1,0,1,0]. Sedangkan *mask* matriks meliputi

$$M_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, M_{3a} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, M_{3b} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$, M_{4a} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, M_{4a} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Hasil dari percobaan [1] yang menyatakan bahwa tedapat perbedaan kecil namun signifikan antara hasil yang didapatkan dengan menggunakan berbagai *mask. Mask* [0,1,0] dan M_{3a} memberikan hasil terbaik. Sedangkan penggunaan M_2, [0,1,1,1,0], $M_{3b,}$ M_{4a}, dan M_{4b} memberikan hasil terburuk.

Untuk gambar yang mengandung tingkat *noisy* yang tinggi, perbedaan antara jumlah piksel-piksel regular dan singular adalah kecil. Konsekuensinya, kurva pada diagram RS berpotongan pada sudut yang kecil. Hal ini berarti tingkat akurasi metode RS Analysis menurun.

Secara keseluruhan, *RS Analysis* akan menyediakan deteksi yang handal untuk pesan rahasia yang disisipkan secara acak hanya jika memenuhi beberapa asumsi. Jika asumsi-asumsi ini tidak dipenuhi, maka pendeteksian akan menemui kegagalan.

4. Pengujian Steganalisis pada Beberapa Kakas Steganografi

Dewasa ini terdapat banyak kakas steganografi yang tersebar di pasaran, misalnya Steganos, StegHide, JPHide, Invisible Secrets, Eztego, S-Tools, Camouflage, dan Hiderman. Dari berbagai kakas steganografi yang ada, kakas yang digunakan untuk menguji metode steganalisis adalah Eztego dan S-Tools. Pembahasan pada Bab 4 ini akan diawali dengan uraian singkat mengenai kakas Eztego dan S-Tools, lalu dilanjutkan dengan pengujian *visual attack* dan *chi-square attack* spesifik terhadap kedua kakas tersebut.

Kakas-kakas steganografi:

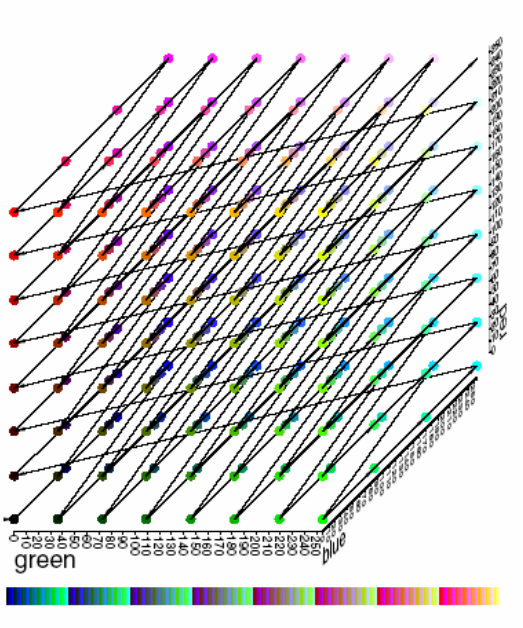
a. Eztego

Algoritma penyisipan pesan Eztego dikembangkan oleh Romana Machado [8]. Algoritma ini hanya dapat digunakan untuk menyisipkan pesan rahasia pada arsip dengan format GIF. Eztego akan menyisipkan pesan secara sekuensial ke dalam indeks palet dari gambar berformat GIF. Gambar berformat GIF merupakan gambar berindeks yang terdiri dari sebuah *array* dan sebuah palet warna. Palet warna tersebut mengandung 256 warna berbeda dengan nilai berada dalam cakupan [0,1].

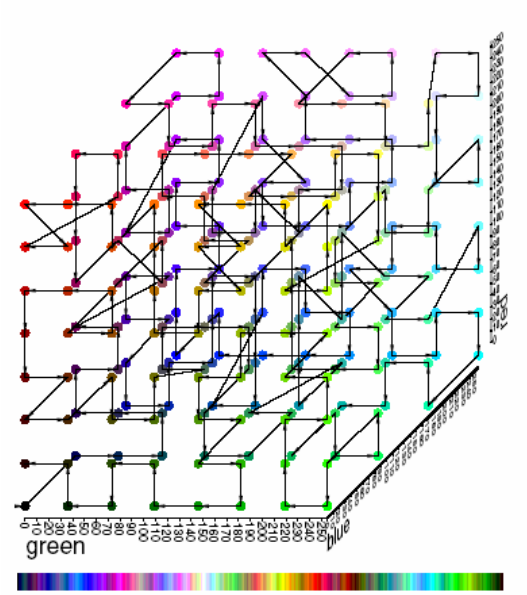
Berikut ini akan dijelaskan format gambar berindeks terlebih dahulu. Setiap baris dari palet menyatakan komponen merah (*Red*), hijau (*Green*), dan biru (*Blue*) dari suatu warna tunggal, seperti pada Gambar 9. Gambar berindeks menggunakan pemetaan langsung dari nilai-nilai piksel ke warna pada palet. Warna dari tiap-tiap piksel gambar ditentukan dengan menggunakan nilai piksel yang berkoresponden sebagai indeks palet. Gambar 10 mengilustrasikan struktur dari gambar berindeks. Pada Gambar 10, dapat dilihat bahwa nilai piksel 5 merujuk pada baris kelima dari palet yang mengindikasikan kombinasi dari komponen warna R=0.2920, G=0.0627, B=0.0627.

Langkah pertama yang dilakukan oleh algoritma Eztego adalah menciptakan salinan palet yang terurut dimana perbedaan antara dua warna yang bertetangga akan diminimalkan. Palet terurut ini diilustrasikan pada Gambar 11. Selama proses penyisipan,

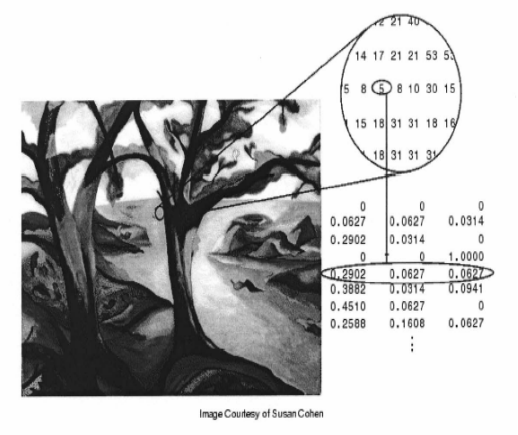
bit-bit pesan disisipkan sebagai LSB dari indeks warna ke palet tadi. Kemudian, algoritma ini akan memasangkan bit pesan dengan LSB dari indeks yang akan disisipkan, lalu menggantikan warnanya dengan warna tetangganya pada palet terurut jika diperlukan. Proses penyisipan ini didemonstrasikan pada Gambar 12.



Gambar 9. Urutan Warna pada Palet

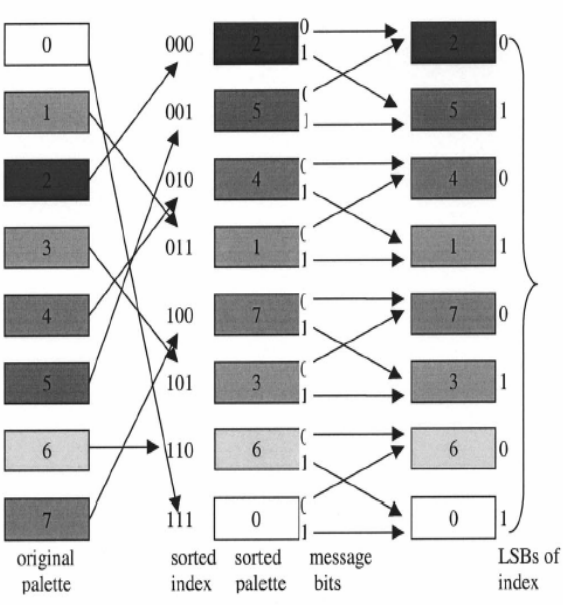


Gambar 11. Urutan Warna pada Eztego



Gambar 10. Struktur Gambar Berindeks

Pada Gambar 11, digunakan palet dengan delapan warna berbeda. Angka 0 hingga 7 didalam kotak merepresentasikan nilai piksel dalam arsip berformat GIF. Misalkan kita ingin menyisipkan "0" ke dalam piksel bernilai 5 yang berkorespondensi dengan indeks 1 (001) pada palet terurut (*sorted palette*). Indeks 1 menjadi indeks 0 dengan bit LSB dari indeks 1 digantikan oleh bit pesan "0" (00**1**→00**0**). Warna dari piksel 5 kemudian digantikan oleh tetangganya yang berindeks 0 pada palet terurut. Karena perbedaan antara kedua warna tersebut sangat kecil, maka perubahan yang terjadi hampir tidak dapat ditangkap oleh mata manusia.



Gambar 12. Penyisipan pada Eztego

b. S-Tools

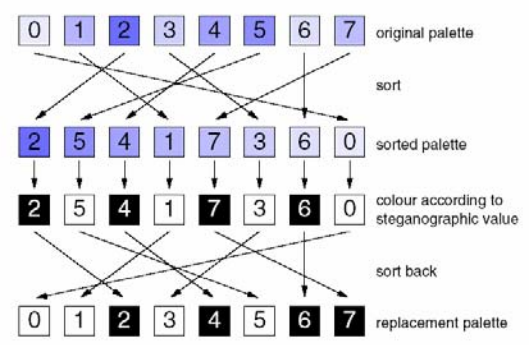
S-Tools merupakan kakas steganografi yang menyisipkan pesan secara acak pada *cover object*. S-Tools mampu menyisipkan pesan pada arsip-arsip dengan format GIF, BMP,dn WAV[5].

Selanjutnya, akan dibahas cara mendeteksi pesan yang disembunyikan dengan menggunakan kedua kakas tersebut. Metode steganalisis yang diujicobakan adalah *visual attack* dan *chi-square attack.*

a. *Visual Attack*

a.1. *Visual Attack* pada Eztego

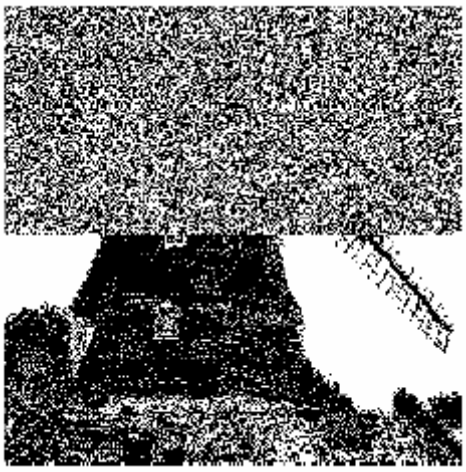
Eztego menggunakan piksel-piksel warna yang ada di palet untuk menentukan bit-bit yang akan disisipkan. Dengan demikian, rancangan fungsi penyaringan yang digunakan pada Eztego akan menggantikan palet asal dengan palet yang hanya terdiri dari dua warna, yaitu hitam dan putih. Warna yang memiliki indeks genap pada palet terurut diganti dengan warna hitam, sedangkan sisanya diganti dengan warna putih. Proses penyaringan ini diilustrasikan pada Gambar 13.



Gambar 13. Skema Fungsi Penyaringan untuk Eztego

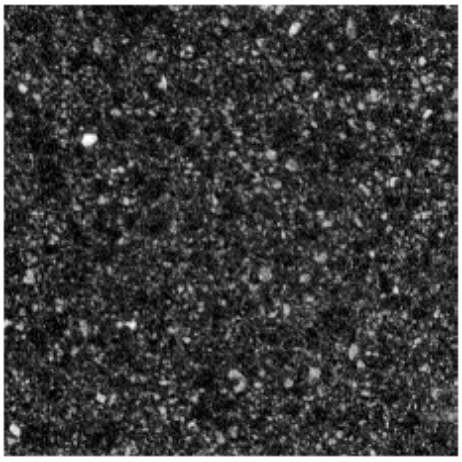
Jika fungsi penyaringan ini diterapkan pada *stego image* yang terdapat pada Gambar 3.b, maka akan terlihat hasilnya pada Gambar 14.

Pada Gambar 14, sangat mudah dideteksi dengan mata manusia, bahwa pesan rahasia terletak pada bagian atas gambar. Hasil penyaringan menunjukkan bahwa 50% dari *cover image* digunakan untuk menyisipkan pesan.



Gambar 14. *Stego Image* (Gambar 3.b) Setelah Disaring

Jika penyaringan diterapkan pada gambar yang tidak mengandung pesan rahasia, maka akan dihasilkan keseluruhan bagian gambar mendekati gambar semula. Hal ini diilustrasikan pada Gambar 15.a dan Gambar 15.b

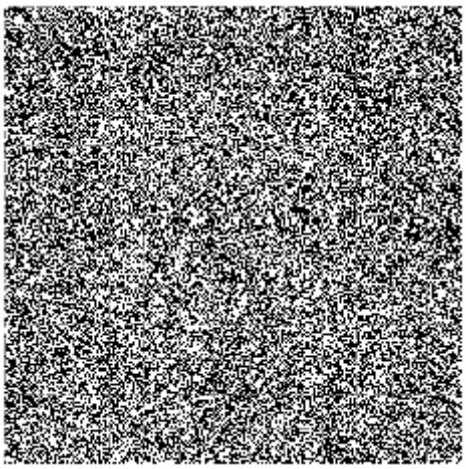


Gambar 15.a *Cover Image*

a.2. *Visual Attack* pada S-Tools

Jika dibandingkan dengan *stego image* yang dihasilkan Eztego, hasil penyaringan pada *stego image* yang dihasilkan oleh S-Tools tidak mempunyai garis pemisah yang tegas antara bagian gambar yang mengandung pesan dan yang tidak mengandung pesan. *Stego image* yang

digunakan dalam percobaan ini merupakan arsip gambar dengan format BMP.



Gambar 15.b Gambar 15.a Setelah Disaring

Pada percobaan ini, *Visual attack* dikenakan pada sebuah gambar dengan tiga jenis ukuran pesan, yaitu ukuran pesan yang relatif pendek (< 50% ukuran *cover image*), ukuran pesan sepanjang ukuran *cover image*, dan ukuran pesan sama dengan setengah ukuran *cover image.* Baik *stego image* maupun gambar yang dihasilkan dari percobaan ini dapat dilihat pada Gambar 16a. dan Gambar 16.b, Gambar 17.a dan Gambar 17.b, serta Gambar 18.a dan Gambar 18.b.

Gambar 16.a *Stego Image* yang Dihasilkan S-Tools dengan Ukuran Pesan < 50% Ukuran Gambar

Dari ketiga gambar hasil penyaringan dapat dilihat bahwa bagian gambar yang mengandung pesan akan disaring menjadi bagian gambar yang "rusak". Banyaknya bagian gambar yang "rusak" sebanding dengan ukuran pesan yang disisipkan.

2. Chi-Square Attack

2.a *Chi-Square Attack* Eztego

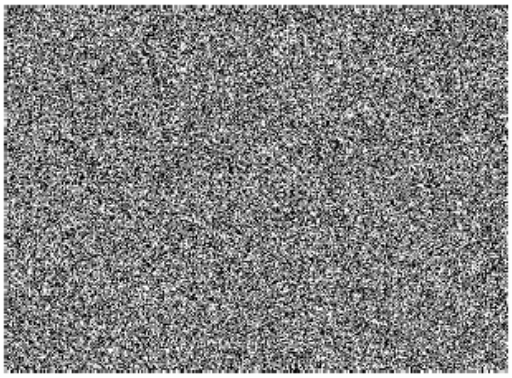
Pada percobaan ini dilakukan pengujian *chi suare* terhadap *stego image* pada Gambar 19. Pesan yang disisipkan berukuran setengah dari ukuran *cover image* dan terletak pada *stego image* bagian atas. Nilai-*p* yang dihasilkan dari pengujian ini diilustrasikan pada Gambar 20.

Gambar 16.b. Gambar 16.a Setelah Disaring

Gambar 17.a *Stego Image* yang Dihasilkan S-Tools dengan Ukuran Pesan = Ukuran Gambar

Dari Gambar 20, untuk sampel pertama, nilai-p yang dihasilkan adalah 0.8826. Sampel berikutnya menambahkan nilai-p menjadi 0.9808, dan seterusnya hingga mencapai setengah dari ukuran sampel sebelum nilai-p menurun drastis. Nilai-*p* =0 pada sisa setengah dari ukuran sampel menunjukkan bahwa bagian gambar tersebut tidak disisipi pesan. Dengan demikian terbukti bahwa Gambar 19 merupakan *stego image* dengan estimasi panjang pesan mendekati setengah

panjang *stego image* serta pesan rahasia disipkan pada setengah bagian pertama dari *stego image.*



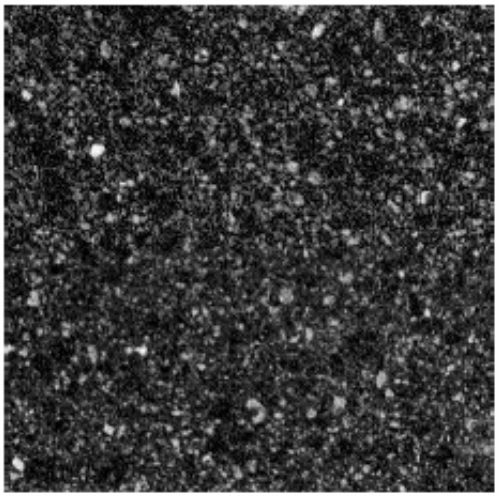
Gambar 17.b. Gambar 17.a Setelah Disaring

Gambar 18.a *Stego Image* yang Dihasilkan S-Tools dengan Ukuran Pesan = 50% Ukuran Gambar

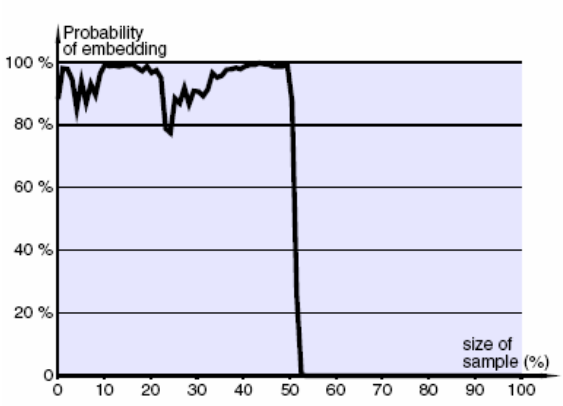
Gambar 18.b. Gambar 18.a Setelah Disaring

2.b. Chi-Square Attack pada S-Tools

Percobaan deteksi dengan metode *chi square attack* pada *stego image* yang dihasilkan dengan S-Tools dapat dilihat pada Tabel 1. Pada tabel tersebut, dilakukan pengujian dengan berbagai kasus, yaitu tidak ada pesan yang disisipkan, ukuran pesan yang disisipkan sebanyak 50% dari ukuran *cover image*, dan ukuran pesan yang disisipkan sebanyak 99.5% dari ukuran *cover image*. ε pada Tabel 1 menyatakan probabilitas *error* yang mungkin terjadi dimana $\varepsilon = 0.5$.



Gambar 19. Stego Image yang Dihasilkan oleh Eztegos



Gambar 20. Grafik Nilai-*p* dari Gambar 19

Tabel 1. Probabilitas Penyisipan pada S-Tools

file	size of embedded text	p-value
jungle.bmp	0	$0+\varepsilon$
bavarian.bmp	0	$0+\varepsilon$
soccer.bmp	0	$0+\varepsilon$
groenemeyer.bmp	0	$0+\varepsilon$
pudding.bmp	0	$0+\varepsilon$
jungle50.bmp	18 090 bytes/50 %	$0+\varepsilon$
jungle100.bmp	36 000 bytes/99.5 %	$1-\varepsilon$
bavarian100.bmp	36 000 bytes/99.5 %	$1-\varepsilon$
soccer100.bmp	36 000 bytes/99.5 %	$1-\varepsilon$
groenemeyer100.bmp	36 000 bytes/99.5 %	$1-\varepsilon$
		$\varepsilon < 10^{-16}$

Dari hasil percobaan pada Tabel 1, dapat dilihat bahwa untuk gambar yang tidak disisipi pesan, nilai probabilitasnya = $0+\varepsilon$ atau nilai-p = 0+0.5 = 0.5. Nilai-p=0.5 ini seharusnya mengindikasikan adanya pesan tersembunyi. Namun ternyata hal tersebut bertentangan dengan keadaan yang sebenarnya. Hal yang demikian juga terjadi pada *stego image* yang disisipi pesan. Nilai-p yang dihasilkan ternyata tidak berpengaruh terhadap panjang pesan. Kedua hal tersebut menunjukkan bahwa *chi-square attack* tidak cocok untuk diterapkan pada S-Tools, dimana penyisipan pesan dilakukan secara acak. Dengan demikian dapat disimpulkan bahwa chi-square attack hanya cocok untuk untuk melakukan steganalisis pada penyisipan pesan secara sekuensial.

5. Kesimpulan

Beberapa kesimpulan yang dapat diambil dari seluruh uraian di atas adalah sebagai berikut:

1. Steganalisis bertujuan untuk mengestimasi panjang pesan rahasia, menentukan lokasi pesan pada *stego-object*, mengungkap *kunci stego,* dan mengekstrak isi pesan rahasia. Namun, hingga saat ini metode-metode steganalisi yang ada hanya dapat mendeteksi ada tidaknya pesan rahasia dalam suatu *steganogram* (meliputi estimasi panjang pesan dan lokasi pesan).

2. Teknik steganografi dengan LSB menghasilkan frekuensi kemunculan yang sama, dimana kesamaan frekuensi ini akan sangat mudah dideteksi.

3. Untuk menghindari kesamaan frekuensi, maka sebaiknya pada teknik steganografi dengan LSB harus dimodifikasi agar tidak ditemukan kesamaan frekuensi. Hal ini bisa dilakukan dengan operasi *increment* pada LSB.

4. Untuk menghindari steganalisis dengan pendekatan statistik, maka sebuah *steganogram* seharusnya memiliki karakteristik statistik yang sama dengan *cover medium*-nya.

5. Performansi steganalisis dengan pendekatan statistik sangat bagus jika diterapkan spesifik pada teknik penyisipan pesan (secara sekuensial atau secara acak).

6. Metode *chi-square* dapat mendeteksi *steganogram* dengan handal pada pesan yang disisipkan secara sekuensial.

7. Metode RS Analysis mampu mendeteksi *steganogram* dengan akurat pada pesan yang disisipkan secara acak.

8. Pendekatan steganalisis dengan pendekatan subjektif mampu mendeteksi penyisipan pesan secara sekuensial maupun acak.

9. Steganalisis sangat sulit dilakukan jika teknik steganografi-nya tidak diketahui.

10. "Perang" antara steganografi dan steganalisis tidak akan pernah berakhir. Seiring dengan ditemukannya algoritma-algoritma steganografi yang baru, maka metode-metode steganalisis pun akan semakin berkembang. Hal ini merupakan tantangan yang membangun bagi kedua bidang tersebut.

Daftar Pustaka

[1] Fridrich, J, dan M. Goldjan. Practical Steganalysis of Digital Images. http://wwwiti.cs.uni-magdeburg.de/iti_amsl/lehre/02_SoSem/mmsec/reference/2002.pdf#search=%22Practical%20Steganalysis%20of%20Digital%20Images.%22
Tanggal Akses: 26 September 2006 pukul 18:30

[2] Jhonson, F Neil, dan Sushil Jajodia. Exploring Steganography: Seeing the Unseen
http://www.jjtc.com/pub/r2026.pdf
Tanggal Akses: 26 September 2006 pukul 19:00

[3] Krenn, Robert. Steganography and Steganalysis.
http://www.krenn.nl/univ/cry/steg/article.pdf
Tanggal Akses: 26 September 2006 pukul 18:05

[4] Munir, Rinaldi. Diktat Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung

[5] Raggo, Michael T. Steganography, Steganalysis, Cryptanalysis.
http://althing.cs.dartmouth.edu/secref/resources/defcon12/dc-12-raggo.ppt
Tanggal Akses: 27 September 2006 pukul 17:05

[6] Wang, Huaiqing, dan Shuozhong Wang. Cyber Warfare: Steganography vs. Steganalysis
http://sunsite.online.globule.org/dblp/db/indices/a-tree/w/Wang:Huaiqing.html
Tanggal Akses: 27 September 2006 pukul 17:40

[8] Westfeld, dan Pfitzmann. Attacks On Steganographic System.
http://www.ece.cmu.edu/~adrian/487-s06/westfeld-pfitzmann-ihw99.pdfs
Tanggal Akses: 27 September 2006 pukul 18:00

[7] Wen Chen. Study Of Steganalysis Methods
http://www.library.njit.edu/etd/2000s/2005/njit-etd2005-006/njit-etd2005-006.html
Tanggal Akses: 27 September 2006 pukul 17:25