Deteksi Steganografi Berbasis *Least Significant Bit* (LSB) Dengan Menggunakan Analisis Statistik

Nur Rokhman dan Juwita Maharanti

Abstrak— Pada penelitian ini dicoba dideteksi penyisipan pesan pada sebuah citra yang dilakukan dengan teknik Least Significant Bit. Deteksi dilakukan dengan teknik Chi-square attack. Pada teknik ini dibandingkan distribusi frekuensi yang diharapkan secara teori dengan beberapa distribusi sampel yang diamati pada stego-image.

Pada penelitian ini dilakukan steganalisis terhadap 10 macam citra yang disisipi dengan sebuah pesan yang berukuran 1kb, 2 kb dan 5 kb. Dicobakan juga 2 penyisipan berbeda, yakni dengan program Stegano dan StegoGraphyBMP.

Dari percobaan yang dilakukan pada sepuluh stego-image dengan panjang pesan 1kb, 2kb, dan 5kb, Chi-square attack berhasil 80% mendeteksi pesan yang disisipkan pada sebuah citra dengan program Stegano dan berhasil 70% mendeteksi pesan yang disisipkan pada sebuah citra dengan program StegoGraphyBMP. Disamping itu diperoleh hasil bahwa ukuran pesan tidak berpengaruh pada proses steganalisis.

Keywords— *Steganalisis, Chi-square.*

I. PENDAHULUAN

Internet banyak digunakan untuk pengiriman data, baik data yang bersifat umum maupun yang bersifat sangat rahasia. Banyak kejahatan yang mengintai dalam proses pengiriman data.

Steganografi merupakan salah satu cara untuk mengamankan data. Namun demikian, tidak tertutup kemungkinan steganografi digunakan untuk mengirimkan suatu pesan rahasia dengan maksud jahat. Marcus [3], menemukan penyisipan file berbau pornografi pada sebuah citra, teks dan file suara yang innocent. Kelley [6], mendapati teroris menggunakan steganografi untuk menyembunyikan komunikasi mereka. Pesan disembunyikan dalam citra yang kemudian dikirimkan melalui internet, khususnya pada situs-situs lelang seperti eBay dan Amazon. Salah satu metode steganografi adalah *Least Significant Bit* (LSB). Cara kerja metode ini adalah dengan mengganti bit terakhir setiap piksel citra pembawa dengan bit-bit pesan yang disembunyikan. Analisis statistik merupakan suatu metode steganalisis sederhana yang dapat digunakan untuk mendeteksi keberadaan suatu pesan dalam suatu citra, meskipun tidak dapat membaca isi pesannya.

Fridrich [2], menguji kehandalan beberapa metode steganalisis yaitu Raw Quick Pairs (RQP), Pairs of Values (PoVs), dan metode yang dapat diaplikasikan ke stego-images dalam suatu format palet. Selain itu diusulkan juga suatu metode baru yaitu RS Analysis. Metode ini dapat mendeteksi dengan akurat penyisipan yang dilakukan secara random. Ide dasar metode ini adalah menemukan dan menghitung hubungan yang lemah antara kelompok LSB dan stego-image.

Maria [4], menganalisis pertahanan terhadap serangan steganalisis yang dilakukan secara analisis probabilistik dan error correcting.

Iwo [5], menganalisis penggunaan pendekatan subjektif dan pendekatan statistik untuk mendeteksi ada tidaknya pesan tersembunyi pada media gambar yang disisipi pesan dalam ranah spasial. Pendekatan statistik yang digunakan adalah metode chi-quare dan metode RS Analysis. Sedangkan metode subjektif yang dibahas adalah metode visual. Iwo [5], menyimpulkan bahwa metode chi-square dapat mendeteksi stego-image dengan akurat pada pesan yang disisipkan secara sekuensial. Metode RS Analysis dapat mendeteksi stego-image dengan akurat pada pesan yang disisipkan secara acak. Pendekatan steganalisis dengan pendekatan subjektif mampu mendeteksi penyisipan pesan secara sekuensial maupun acak.

Yitnanto [7], menggabungkan teknik LSB insertion, fungsi hash, kompresi dan kriptografi untuk menyembunyikan dan mengekstrak suatu pesan rahasia dalam citra. Format citra yang digunakan berupa BMP, PNG, dan JPEG.

I.I STEGANALISIS MELALUI CHI-SQUARE ATTACK

Steganalisis merupakan suatu seni dan ilmu mendeteksi pesan-pesan yang disembunyikan dengan menggunakan steganografi [3]. Steganografi bekerja dengan mengganti bit-bit data yang tidak berguna dalam berkas digital seperti citra, suara, teks, atau HTML (disebut *cover data*) dengan bit-bit informasi yang disembunyikan (*embedded message*).

Metode LSB mengambil keuntungan dari keterbatasan indera manusia dengan cara menurunkan kualitas suatu obyek. Metode ini bekerja dengan cara menyisipkan *embedded data* ke dalam bit rendah pada *cover data*.

Sebagai contoh, terdapat data raster original *cover data* berupa citra :

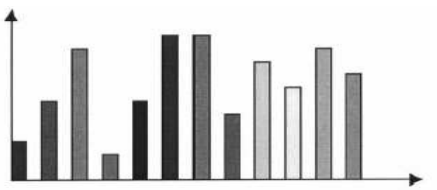
00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001

Representasi biner huruf A adalah 01000001. Dengan menyisipkannya ke dalam pixel di atas maka akan dihasilkan :

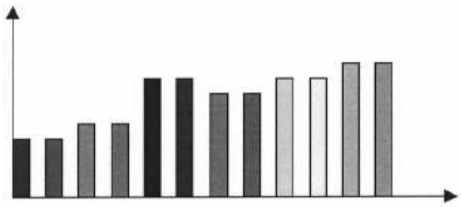
00100111 11101001 11001000
00100111 11001000 11101000
11001000 00100111 11101001

Terlihat pada bit ke-8, 16 dan 24 diganti dengan representasi biner huruf A, dan hanya tiga bit rendah yang berubah (cetak tebal).

Prinsip Chi-square attack dalam mendeteksi keberadaan *embedded message* adalah dengan menguji seberapa signifikan kemunculan pasangan dengan nilai yang sama. Hal ini berarti membandingkan distribusi frekuensi yang diharapkan secara teori dan distribusi frekuensi dari beberapa sampel yang diamati (*stego-object*). Gambar 1 menunjukkan histogram warna untuk citra yang belum disisipi pesan sedang Gambar 2 untuk citra yang telah disisipi pesan.



Gambar 1.
Contoh Histogram warna sebelum penyisispan



Gambar 2.
Contoh Histogram warna sesudah penyisipan

Masalah utama dalam merancang pendeteksian ini adalah menentukan distribusi frekuensi yang diharapkan secara teori (frekuensi kejadian yang diharapkan sebelum adanya perubahan steganografi). Distribusi frekuensi ini tidak dapat diperoleh dari *stego-image* karena sampel random sudah dimodifikasi dengan operasi steganografi. Untuk itu diambil rata-rata dari dua frekuensi dalam pasangan nilai (*Pairs of Values*/PoVs). Hal ini berdasar asumsi bahwa pertukaran sebuah nilai menjadi nilai yang lain tidak mengubah jumlah kedua nilai tersebut, sehingga rata-rata dari kedua frekuensi tersebut tidak berubah baik untuk citra asal dan *stego-image*. Hal inilah yang mendasari penentuan frekuensi yang diharapkan dari *stego-image*.

Ketika distribusi sampel yang diamati dan distribusi 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:

Misalkan terdapat *k* kategori dan terdapat sebuah sampel acak dari hasil observasi. 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.

1. Frekuensi yang diharapkan pada kategori ke *i*, dimana *i=1,2,3,...,k* setelah penyisipan bit pesan yang terdistribusi merata, dapat dihitung dengan :
n_i' = *jumlah indeks ke* c_i

2. Frekuensi aktual dari sampel dihitung dengan :
n_i= *jumlah indeks ke* c_{2i}

3. Nilai chi-square dihitung dengan :

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

dengan derajat kebebasan= k-1

4. Kemungkinan distribusi n_i' dan n_i adalah sama dinyatakan dengan :

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

di mana Γ adalah fungsi Gamma Euler,

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

Jika distribusi n_i' sama dengan n_i

maka X_{k-1}^2 akan mendekati 0. Dengan demikian nilai *p* akan mendekati 1. Jika nilai *p* semakin mendekati 1, maka semakin besar kemungkinan pixel disisipi pesan. Sebaliknya, jika nilai *p* semakin mendekati 0, maka semakin besar kemungkinan piksel tidak disisipi pesan [8].

Dalam analisis statistik, fungsi-fungsi dirancang untuk membedakan spesifikasi statistik antara *cover-object* dan *stego-object*. Hasil dari fungsi ini kemudian dibandingkan dengan nilai batas tertentu (*threshold*) untuk menentukan ada tidaknya pesan rahasia dalam *stego-object*.

I.II CITRA DIGITAL

Citra (*image*) digital adalah kumpulan baris angka-angka yang menunjukkan intensitas warna pada berbagai titik (*pixel*). Variasi warna berasal dari tiga warna dasar, yaitu merah, hijau, dan biru. Setiap warna dasar diwakili 1 byte. Citra 24-bit menggunakan 3 byte tiap piksel untuk menggambarkan satu nilai warna. Misalnya sebuah warna putih memiliki nilai FFFFFF, yang berarti 100 persen merah (FF), 100 persen hijau (FF), dan 100 persen biru (FF) [1].

Warna merah murni, hijau murni, dan biru murni dalam format biner, masing-masing adalah :

00000000 00000000 11111111
00000000 11111111 00000000
11111111 00000000 00000000

Dari uraian di atas dapat dilihat bahwa informasi dari warna biru berada pada bit 1 sampai bit 8, informasi warna hijau berada pada bit 9 sampai dengan bit 16, dan informasi warna merah berada pada bit 17 sampai dengan bit 24..

II. DESAIN PERCOBAAN

Ide dasar dari chi-square attack adalah membandingkan distribusi frekuensi yang diharapkan secara teori dengan beberapa distribusi sampel yang diamati pada *stego-image* Oleh karena itu, kunci pokok pendeteksian *stego-image* pada *Chi-square attack* adalah pencarian nilai probabilitas *stego-image*. Nilai probabilitas dapat diketahui melalui pengujian chi-square terhadap *stego-image* yang dilakukan melalui pengambilan beberapa sampel data dari *stego-image*. Output dari pengujian chi-square berupa histogram yang menampilkan probabilitas per persen data.

Berikut ini langkah-langkah pendeteksian *stego-image* :

1.Tentukan *Pair of Values* (PoV) beberapa sampel data dari stego-image. Pada penyisipan pesan acak, kedua nilai PoV memiliki frekuensi sama.

a.Untuk 2 bit terakhir, berarti ada 4 nilai yang mungkin dengan 2 PoV yang berbeda.

No	PoV
1	00 01
2	10 11

b.Untuk 3 bit terakhir, berarti ada 8 nilai yang mungkin dengan 4 PoV yang berbeda.

No	PoV
1	000 001
2	010 011
3	100 101
4	110 111

c.PoV tersebut diteruskan hingga 8 bit tiap pixel. Jadi, dalam 1 byte maksimal ada 256 nilai yang mungkin dengan 128 PoV.

2. Menghitung frekuensi aktual PoV dari *stego-image*. Sebagai contoh, pada populasi 500, muncul nilai **01011100** sebanyak 100 kali dan **01011101** sebanyak 400 kali.

3. Menghitung frekuensi PoV yang diharapkan. Penghitungan frekuensi yang diharapkan ini bersamaan dengan penghitungan frekuensi aktual. PoV mempunyai frekuensi masing-masing 50%. Dengan menggunakan contoh pada nomor 2, maka kemunculan nilai **01011100** sebanyak 250 kali dan kemunculan nilai **01011101** sebanyak 250 kali.

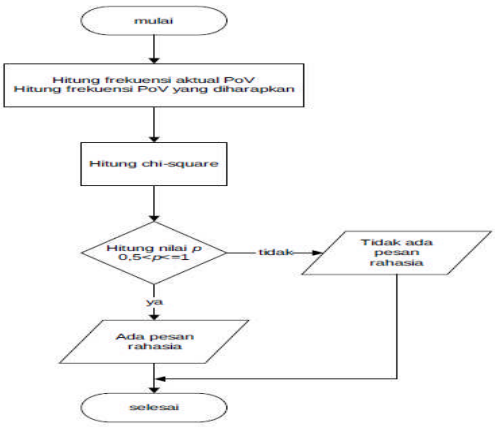
4. Menghitung nilai chi-square x^2. Penghitungan dilakukan dengan membandingkan distribusi frekuensi aktual dan frekuensi yang diharapkan dengan rumus :

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

dengan n_i adalah frekuensi aktual dan n_i ' adalah frekuensi yang diharapkan.

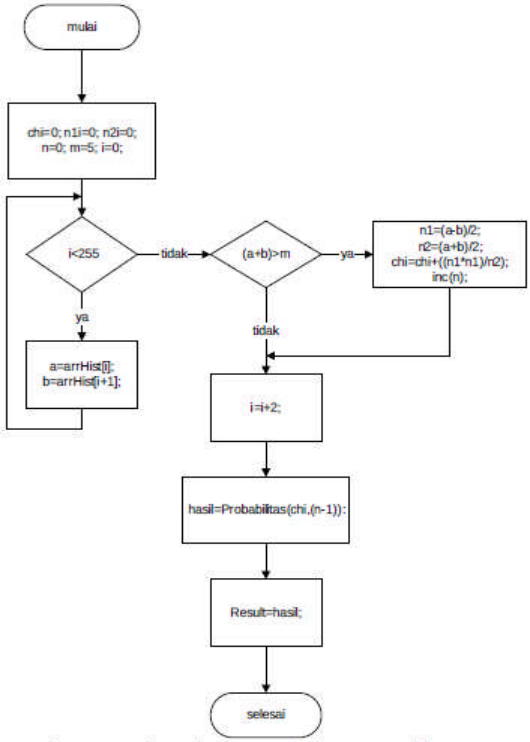
5. Menghitung nilai probabilitas (*p*). Jika dua populasi (aktual dan yang diharapkan) berbeda jauh sehingga nilai *p* mendekati 0, maka distribusi LSB tersebut tidak acak, sehingga besar kemungkinannya tidak ada pesan yang disisipkan dalam LSB. Sebaliknya jika hampir sama sehingga nilai *p* mendekati 1, maka distribusi LSB tersebut acak, sehingga besar kemungkinannya ada pesan yang disisipkan dalam LSB.

Flowchart pendeteksian *stego-image* dengan chi-square attack diperlihatkan pada Gambar 3.



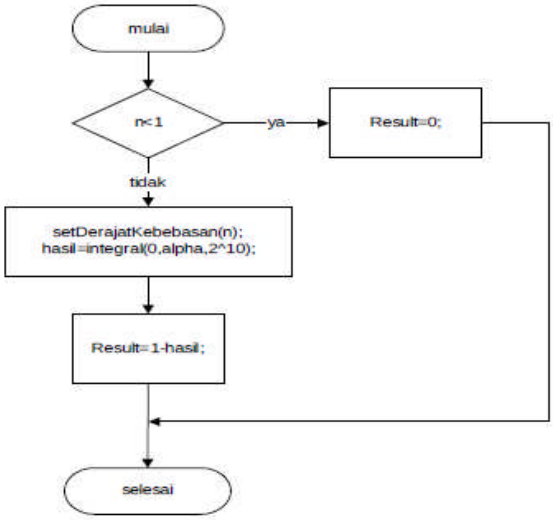
Gambar 3. Flowchart pendeteksian stego-image dengan chi-square attack

Gambar 4 menunjukkan flowchart penghitungan chi-square. Penghitungan chi-square dilakukan selama i ≥ 0 dan i < 255. Pada flowchart ini terdapat fungsi *probabilitas* untuk menghitung probabilitas *stego-image*.

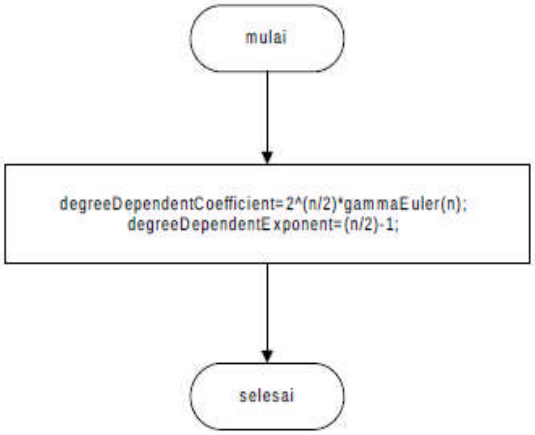


Gambar 4. Flowchart penghitungan chi-square

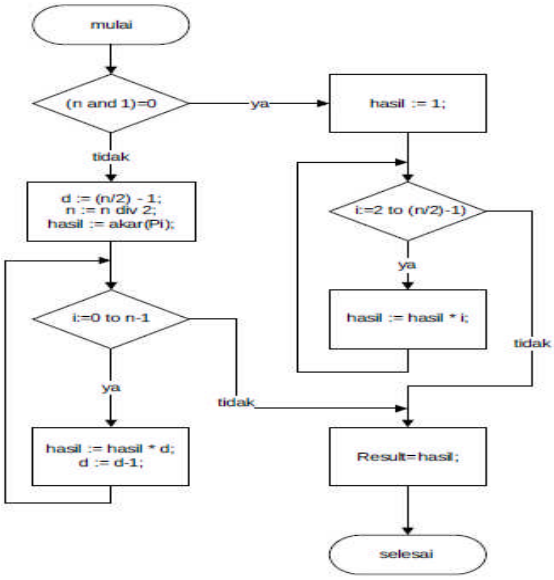
Flowchart fungsi Probabilitas ditunjukkan pada Gambar 5. Dalam fungsi Probabilitas, fungsi *set Derajat Kebebasan* (Gambar 6) digunakan untuk menentukan derajat kebebasannya, sedang fungsi *integral* (Gambar 7) digunakan untuk menghitung rata-rata distribusi frekuensi aktual dan distribusi frekuensi yang diharapkan.



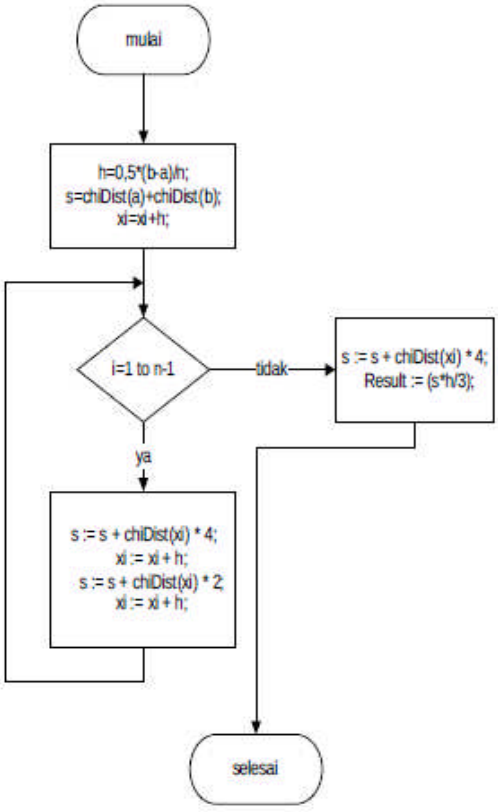
Gambar 5. Flowchart penghitungan probabilitas.



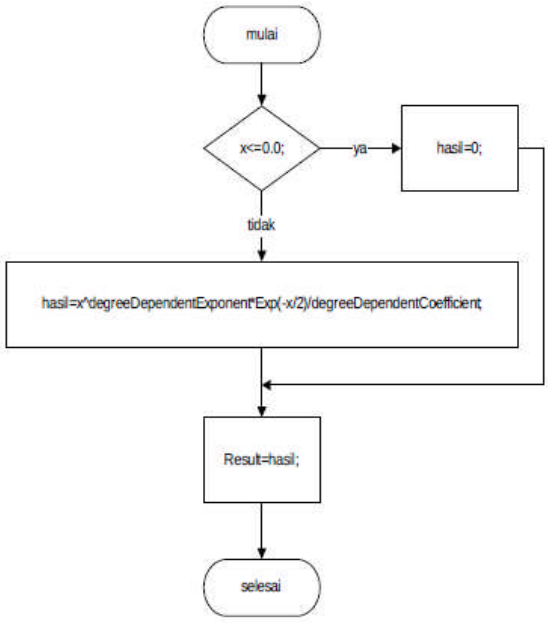
Gambar 6. Flowchart fungsi Derajat kebebasan



Gambar 8. Flowchart fungsi GammaEuler

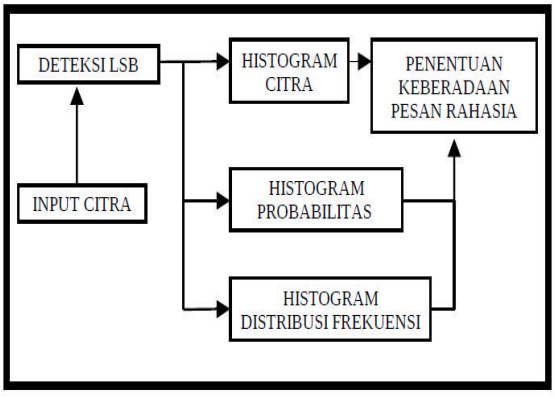


Gambar 7. Flowchart penghitungan integral



Gambar 9. Flowchart fungsi chiDist

Fungsi gammaEuler (Gambar 8) merupakan rumus yang diperlukan untuk menghitung derajat kebebasan, sedangkan fungsi chiDist (Gambar 9) digunakan untuk menghitung distribusi frekuensi sampel yang diambil dari citra yang dideteksi. Pada Gambar 10 diperlihatkan sistem steganalisis secara menyeluruh.



Gambar 10. Sistem Steganalisis

III. PEMBAHASAN

Pada penelitian ini percobaan dilaksanakan dengan tujuan untuk mendeteksi keberadaan pesan yang disisipkan pada 10 citra. Pada percobaan I, penyisipan dilakukan dengan menggunakan program Stegano dan pada percobaan II dilakukan dengan menggunakan program StegoGraphyBMP. Pada Tabel 1, dari 10 citra yang disisipi pesan dengan menggunakan Stegano, 8 diantaranya berhasil dideteksi oleh chi-square attack. Sedangkan dua *stego-image* yang lain tidak berhasil dideteksi. Dengan kata lain, 80% dari *stego-image* yang diuji berhasil dideteksi oleh chisquare attack.

Tabel 1. Deteksi pesan terhadap penyisipan dengan program Stegano

No	Citra	Ukuran Pesan Yang Disisipkan		
		1 kb	2 kb	5 kb
1	balaikota_solo	√	√	√
2	pasar_gede	--	--	--
3	bank_indonesia	√	√	√
4	MIPA	√	√	√
5	road_to_paris	√	√	√
6	sunset_paris	--	--	--
7	stadion	√	√	√
8	imogiri	√	√	√
9	dilarang_memotret	√	√	√
10	yellow_gate	√	√	√

Pada Tabel 2, dari 10 citra yang disisipi pesan dengan menggunakan StegoGraphyBMP, 7 diantaranya berhasil dideteksi oleh chi-square attack. Sedangkan 3 *stego-image* yang lain tidak berhasil dideteksi. Dengan kata lain, 70% dari *stego-image* yang diuji berhasil dideteksi oleh chi-square attack.

Tabel 2. Deteksi pesan terhadap penyisipan dengan program StegoGraphyBMP

No	Citra	Ukuran Pesan Yang Disisipkan		
		1 kb	2 kb	5 kb
1	balaikota_solo	√	√	√
2	pasar_gede	--	--	--
3	bank_indonesia	√	√	√
4	MIPA	√	√	√
5	road_to_paris	--	--	--
6	sunset_paris	--	--	--
7	stadion	√	√	√
8	imogiri	√	√	√
9	dilarang_memotret	√	√	√
10	yellow_gate	√	√	√

Dari Tabel 1 dan Tabel 2 juga dapat dilihat bahwa ukuran pesan tidak berpengaruh terhadap steganalisis.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada sistem steganalisis,ada beberapa kesimpulan yang dapat diambil, yaitu :

1. Sistem steganalisis dapat mendeteksi pesan yang disembunyikan dalam citra BMP.

2. Dari percobaan yang dilakukan pada sepuluh *stego-image* dengan panjang pesan 1kb, 2kb, dan 5kb, chi-square attack berhasil 80% mendeteksi pesan rahasia yang disisipkan pada citra dengan program Stegano dan 70% untuk penyisipan dengan StegoGraphyBMP.

3. Ukuran pesan tidak berpengaruh terhadap proses steganalisis.

V. DAFTAR PUSTAKA

[1] Gonzalez, R.C dan Woods, R.E 2002, *Digital Image Processing*. Prentice- Hall, New Jersey.

[2] Fridrich, J., Goljan M., dan Du, Rui. *Detecting LSB Steganography in Color and Gray-Scale Images.* http://csdl.computer.org/dl/mags/mu/2001/04/u4022. pdf , Tanggal Akses : 18Oktober 2006

[3] Marcus, Ilana. *Steganography Detection* http://www.uri.edu/personal2/imarcus/stegdetect.htm , Tanggal Akses : 18 Februari 2007

[4] J.S., Ana Maria. *Analisis dan Studi Kasus Pertahanan Terhadap Serangan Steganalisis yang Menggunakan Teknik Analisis Statistik*. http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-024.pdf , Tanggal Akses : 19 Februari 2007

[5] Iwo, Maria Helena. *Steganalisis Khusus dengan Pendekatan Subjektif dan Statistik pada Stego Image*. http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-030.pdf , Tanggal Akses : 19 Februari 2007

[6] Kelley, Jack. *Terror Groups Hide Behind Web Encryption.* http://www.usatoday.com/tech/news/2001-02-05-binladen.htm , Tanggal Akses : 21 Februari 2007

[7] Yitnanto, Deddy Sulistyawan Dwi. 2004, *Aplikasi Steganografi Pada File Citra*. Fakultas FMIPA, Yogyakarta.

[8] Westfeld, Andreas dan Pfitzmann, Andreas. *Attacks on Steganographic Systems*. http://www.ece.cmu.edu/~adrian/487- s06/westfeld pfitzmann-ihw99.pdf , Tanggal Akses : 12 Juni 2007