Program Steganalisis Metode *LSB* pada Citra dengan *Enhanced LSB*, Uji *Chi-Square*, dan *RS-Analysis*

Yulie Anneria Sinaga – 13504085 [1)]
1) Program Studi Teknik Informatika ITB, Bandung 40132, email: if14085@students.if.itb.ac.id

Abstract – *Pada makalah ini dibangun suatu perangkat lunak yang dapat melakukan steganalisis atau mendeteksi ada tidaknya pesan rahasia pada suatu citra. Citra merupakan media yang paling sering digunakan untuk menyisipkan pesan rahasia, karena dapat menyembunyikan pesan dengan sangat baik dan banyak tersedia. Perangkat lunak yang dibangun mengimplementasikan steganalisis terhadap metode LSB, metode ini merupakan teknik penyisipan pesan yang paling banyak digunakan.*

Perangkat lunak yang dibangun menggunakan algoritma enhanced LSB, uji chi-square, dan RS-analysis untuk melakukan steganalisis. Algoritma-algoritma tersebut merupakan algoritma steganalisis yang diciptakan khusus untuk penyisipan dengan metode LSB. Enhanced LSB merupakan teknik steganalisis secara visual yang melibatkan indra penglihatan manusia. Sedangkan uji chi-square dan RS-analysis merupakan teknik steganalisis secara statistik, dimana dibutuhkan perhitungan statistik terhadap pixel citra. Apabila ditemukan pesan rahasia maka perangkat lunak dapat menghancurkan pesan rahasia yang dikandung.

Perangkat lunak dibangun menggunakan bahasa C# di atas framework .net. Berdasarkan pengujian perangkat lunak yang dilakukan, dapat dilihat bahwa perangkat lunak dapat berjalan dengan baik dalam mendeteksi ada tidaknya pesan rahasia dan menghancurkan pesan tersebut apabila ditemukan. Setiap algoritma steganalisis yang diimplementasi hanya akurat untuk kasus-kasus yang sesuai, enhanced LSB sesuai untuk citra dengan kontras tinggi, uji chi-square sesuai untuk penyisipan pesan secara sekuensial, dan RS-analysis sesuai untuk penyisipan pesan secara acak.

Kata Kunci: *citra, enhanced-LSB, metode LSB, RS-analysis, steganalisis, uji chi-square*

1. PENDAHULUAN

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui [MUN06]. Metode yang paling banyak digunakan untuk melakukan steganografi adalah *LSB* (*Least Significant Bit*) [MIT03]. Metode ini banyak digunakan karena metode ini paling sederhana dan mudah diimplementasikan.

Media penampung yang paling sering digunakan dalam melakukan steganografi adalah citra digital karena jumlahnya yang besar di internet [MOR05]. Kehandalan penggunaan citra dibandingkan dengan media lain adalah kualitas citra yang telah disisipi pesan rahasia tidak berbeda jauh dengan kualitas citra aslinya.

Kelebihan dari steganografi adalah pesan yang dikirim tidak menarik perhatian. Hal inilah yang membuat steganografi sering digunakan dalam melakukan komunikasi rahasia antar teroris atau kriminal [GUP05]. Oleh karena itu dibutuhkan suatu teknik untuk mendeteksi apakah terdapat pesan rahasia pada suatu objek yang disebut steganalisis.

Steganalisis dilakukan untuk mendeteksi ada-tidaknya pesan tersembunyi dalam suatu objek [KHA06]. Apabila ditemukan pesan tersembunyi, maka umumnya para penegak hukum akan melakukan ekstraksi pesan tersembunyi tersebut atau paling tidak menghancurkannya sehingga pesan tersebut tidak berguna lagi bagi penerima pesan [GUP05].

Karena kemudahan penggunaan metode LSB sebagai teknik penyisipan pesan dan banyaknya citra digital sebagai media penampung, maka terdapat banyak kakas steganografi untuk citra menggunakan metode LSB. Oleh karena itu dilakukan implementasi steganalisis metode LSB untuk mendeteksi keberadaan pesan tersembunyi pada citra tidak terkompresi. Dan untuk memperkuat aplikasi ini maka aplikasi ini dilengkapi dengan tindak lanjut yang perlu dilakukan apabila ditemukan pesan tersembunyi, seperti menghancurkan pesan tesebut. Dengan adanya aplikasi ini diharapkan pesan tersembunyi dapat dideteksi dan dihancurkan dengan mudah.

2. LANDASAN TEORI

2.1. Citra Dijital

Hampir semua format arsip dijital dapat digunakan untuk steganografi, tetapi format yang paling cocok adalah yang memiliki tingkat *redundancy* yang tinggi.

Redundancy dapat didefinisikan sebagai jumlah bit berlebih dari sebuah objek yang menghasilkan akurasi jauh lebih besar dari yang dibutuhkan untuk penggunaan dan menampilkan objek. Bit berlebih dari suatu objek adalah bit-bit yang dapat diubah akan tetapi menghasilkan perubahan yang tidak dapat dideteksi dengan mudah pada objek tersebut. Oleh karena itu citra yang paling sesuai untuk steganografi ialah citra tidak terkompresi *bitmap* 24 bit.

2.2. Steganografi

Steganografi, berasal dari bahasa Yunani yaitu *stegos* yang berarti atap atau tertutup dan *graphia* yang berarti tulisan, adalah ilmu dan seni menyembunyikan keberadaan komunikasi [MOR05]. Dengan menggunakan steganografi, pesan rahasia dapat disisipkan ke dalam sebuah media yang tidak mencurigakan dan mengirimnya tanpa ada seorangpun yang mengetahui keberadaan pesan tersebut.

Menyisipkan data, yang ingin disembunyikan, ke dalam sebuah media membutuhkan dua buah arsip. Arsip pertama adalah media penampung (citra, suara, video dsb.) yang terlihat tidak mencurigakan untuk menyimpan pesan rahasia. Arsip kedua adalah pesan yang ingin disembunyikan. Media penampung berupa citra disebut juga *cover-image* dan citra yang telah disipi pesan disebut *stego-image*

2.3. Metode *LSB*

Untuk menjelaskan metode ini, digunakan citra dijital sebagai *cover-object*. Pada setiap byte terdapat bit yang paling kurang berarti (*Least Significant Bit* atau *LSB*). Misalnya pada byte 0001100**1**, maka bit *LSB*-nya adalah **1**. Untuk melakukan penyisipan pesan, bit yang paling cocok untuk diganti dengan bit pesan adalah bit *LSB*, sebab pengubahan bit tersebut hanya akan mengubah nilai byte-nya menjadi satu lebih tinggi atau satu lebih rendah .
Sebagai contoh, urutan bit berikut ini menggambarkan 3 *pixel* pada *cover-image*
24-bit.

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

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

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

Ada dua jenis teknik yang dapat digunakan pada metode *LSB*, yaitu penyisipan pesan secara sekuensial dan secara acak. Sekuensial berarti pesan rahasia disisipkan secara berurutan dari data titik pertama yang ditemukan pada file gambar, yaitu titik pada pojok kanan bawah gambar. Sedangkan acak berarti penyisipan pesan rahasia dilakukan secara acak pada gambar, dengan masukan kata kunci (*stego-key*).

2.4 Steganalisis

Pengertian steganalisis mengacu pada seni dan ilmu pengetahuan dalam mendeteksi ada-tidaknya pesan tersembunyi dalam suatu objek [KHA06].

Steganalisis untuk metode *LSB* terdiri dari metode subjektif dan metode statistik [WEN00]. Metode subjektif melibatkan indera penglihatan manusia untuk mengamati bagian gambar yang dicurigai, sehingga disebut juga *visual attack*. Salah satu teknik steganalisis secara visual adalah metode *enhanced LSB*. Metode ini menampilkan bit-bit terakhir dari sebuah citra dan mengandalkan penglihatan manusia untuk menentukan ada tidaknya pesan rahasia dalam citra.

Sedangkan metode statistik melibatkan analisis matematis terhadap sebuah gambar untuk menemukan perbedaan antara gambar asli dengan gambar yang telah disisipi pesan. Meskipun *stego-image* identik dengan *cover-image*-nya bila ditangkap dengan indera penglihatan, *stego-image* seringkali menunjukkan statistik yang tidak biasa yang membedakan *stego-image* dari *cover-image*-nya. Tujuan dari steganalisis statistik memperlihatkan ketidakbiasaan ini adalah untuk menunjukkan perbedaan yang kuat antara *stego-image* dengan *cover-image*-nya.

Metode statistik yang akan dibahas adalah metode uji *chi-square* dan metode *RS-analysis*. Uji *chi-square* terbukti handal dalam mendeteksi pesan rahasia yang disisipkan secara sekuensial. Metode lainnya adalah *RS-analysis* yang terbukti handal dan akurat dalam mendeteksi pesan rahasia yang disisipkan secara acak.

2.4.1 *Enhanced LSB*

Algoritma ini dikemukan oleh Andreas Westfeld. Proses utama dari metode *enhanced LSB* akan dijelaskan sebagai berikut: setiap *pixel* memiliki tiga buah komponen yaitu *red, green, blue*. Setiap komponen direpresentasikan oleh satu byte, setiap byte memiliki sebuah bit *LSB*. Apabila bit *LSB* tersebut adalah 1, maka semua bit pada byte tersebut diganti dengan bit 1 sehingga nilai byte tersebut adalah 11111111 (biner) atau 255 (desimal). Sedangkan, apabila bit *LSB* tersebut adalah 0, maka semua bit pada byte tersebut diganti dengan bit 0 sehingga nilai byte tersebut adalah 00000000 (biner) atau 0 (desimal). Misalnya terdapat sebuah *pixel* dengan komposisi byte sebagai berikut :

BLUE	GREEN	RED
1010010**1**	1001110**0**	1110011**1**

Maka setelah mengalami *enhanced LSB* byte-byte diatas akan menjadi :

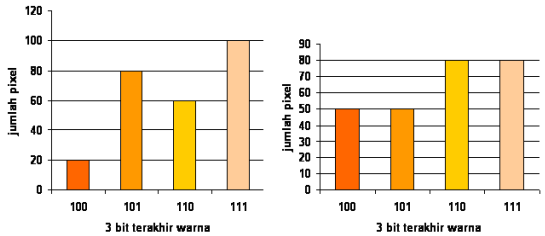
BLUE	GREEN	RED
11111111	00000000	11111111

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.

2.4.2 Uji *Chi-Square*

Gagasan ini juga dikemukakan oleh Andreas Westfeld, yaitu bahwa gambar yang telah disisipi pesan akan memiliki frekuensi yang relatif sama antara *Pair of Values* (*PoV*) yang bersangkutan [WES99]. *PoV* adalah pasangan titik yang hanya berbeda di *LSB*-nya saja, seperti 00000000 dan 00000001. Misalkan terdapat suatu gambar yang hanya memiliki kedua warna tersebut, dengan distribusi sebesar 20 dan 80 masing-masing titik. Setelah disisipi pesan, frekuensi tersebut akan berubah mendekati 50 dan 50, karena jumlah *PoV* pasti berjumlah tetap, dan jumlah dari tiap-tiap *PoV* akan cenderung menjadi sama [GUI04], hal ini dapat dilihat pada **Gambar 1**.

Metode ini bekerja dengan melakukan perbandingan uji *chi-square* antara dua buah statistik distribusi frekuensi, yang pertama adalah statistik pada gambar yang dicurigai mengandung pesan tersembunyi, dan yang kedua adalah statistik yang diprediksi akan dimiliki oleh gambar tersebut apabila disisipi pesan. Apabila kedua statistik ini sama, atau terdapat suatu bagian yang sama, maka kemungkinan besar terdapat suatu pesan dalam gambar. Gambaran dari kedua distribusi tersebut dapat dilihat pada **Gambar 1**.



Gambar 1 Histogram warna sebelum disisipi pesan (kiri) dan sesudah disisipi pesan (kanan)

Ujicoba ini beroperasi pada pemetaan observasi ke dalam kategori-kategori, berikut langkah-langkahnya :

1. Anggap terdapat k kategori dan sampel acak dari hasil observasi. Tiap hasil observasi harus dimasukan ke dalam satu dan hanya satu kategori. Jadi kita memiliki 128 *Pairs of Values* (dari 256 warna) sehingga jumlah kategori k=128.

2. Anggap n_i menjadi jumlah *pixel* pada citra yang diobservasi dengan nilai warna 2*i* dimana *i* = 0,1,2,...,k-1

3. Anggap n_i^* adalah jumlah *pixel* yang diharapkan pada kategori *i* setelah sebuah pesan disisipkan dan terdistribusi merata pada citra, nilai ini dapat dihitung dengan persamaan (2.1):

$$n_i^* = \frac{\text{jumlah pixel dengan nilai warna } 2i + \text{jumlah pixel dengan nilai warna } 2i+1}{2} \quad ..(2.1)$$

4. Nilai *chi-square* ($\chi 2$) dengan derajat kebebasan $k-1$ dapat diperoleh dengan persamaan (2.2):

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*} \quad \text{.....(2.2)}$$

5. *p* adalah probabilitas bahwa distribusi dari n_i dan n_i^* adalah sama. Nilai *p* diperoleh dengan persamaan (2.3):

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\frac{k-1}{2}} \int_0^{\chi^2_{k-1}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1}\, dx \quad \text{.....(2.3)}$$

Apabila distribusi frekuensi memiliki perbedaan yang signifikan maka distribusi dari *LSB* tidak bersifat acak, yang berarti kemungkinan besar tidak terdapat pesan rahasia. Sedangkan apabila kedua frekuensi ini tidak memiliki perbedaan yang signifikan maka distribusi *LSB* mendekati acak, sehingga kemungkinan besar terdapat pesan rahasia yang telah disisipkan pada *LSB* citra.

2.4.3 *RS-Analysis*

RS-analysis dikemukakan oleh Fridrich et al.[FRI02]. Teknik ini memanfaatkan korelasi spasial pada *stego-image*. *RS-analysis* dapat mendeteksi penyisipan secara acak dengan akurat.

Diberikan gambar yang kemudian dipartisi menjadi kelompok-kelompok *n pixel* yang bertetangga ($x_1,...,x_n$). Untuk mendapatkan korelasi spasial, digunakan fungsi diskriminasi *f* , dimana *f* merupakan nilai absolut rata-rata dari perbedaan antara *pixel-pixel* yang bertetangga. Secara matematis, fungsi diskriminasi *f* dinyatakan dalam persamaan (2.4).

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \quad \text{.....(2.4)}$$

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. Penyisipan dengan *LSB* dapat

dideskripsikan dengan menggunakan fungsi *flipping* F_1 dan fungsi *dual flipping* F_{-1} sebagai berikut :

$$F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$$
$$F_{-1} : 1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$$
$$F_0 : F_0(x) = x \quad \forall x \in P$$

Hubungan antara fungsi *flipping* F_1 dan fungsi *dual flipping* F_{-1}, ditunjukan oleh persamaan (2.5).

$$F_{-1}(x) = F_1(x+1) - 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.5)$$

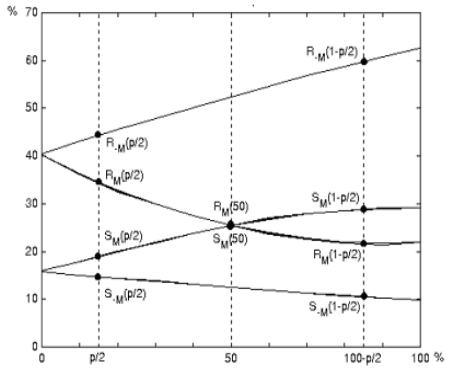
Selanjutnya kelompok *pixel K* dapat diklasifikasikan ke dalam tiga tipe berbeda, yaitu R, S, dan U. Secara formal, ketiga tipe tersebut dapat dinyatakan sebagai berikut:

- 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*).

Secara umum, operasi *flipping* yang berbeda diaplikasikan pada *pixel-pixel* yang berbeda dalam kelompok *K*. Pola *pixel-pixel* untuk di-*flip* disebut *mask*.. Kelompok $F(G)$ yang telah di-*flip* didefinisikan sebagai $(F_{M(1)}(x_1), \ldots, F_{M(n)}(x_n))$, dimana $M(i)$ dengan i=1,2,..,n adalah elemen dari *mask M* dan nilainya dapat berupa -1,0,1.

Berdasarkan klasifikasi yang dilakukan, terbentuk sebuah diagram yang disebut diagram RS, dapat dilihat pada **Gambar 2** (Penurunan diagram ini tidak dijelaskan, untuk lebih jelasnya dapat dilihat pada makalah FRI02]). Diagram tersebut memperlihatkan R_{-M},S_M , R_{-M} dan S_{-M} sebagai fungsi dari jumlah *pixel* dengan *LSB* yang telah di-*flip,* dimana sumbu-x (absis) merepresentasikan persentasi *pixel* dengan *LSB* yang di-*flip,* sedangkan sumbu-y (ordinat) merepresentasikan jumlah relatif dari kelompok regular dan kelompok singular dengan *mask* M dan –M, dimana M = [0,1,1,0].



Gambar 2 Diagram RS [FRI02]

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:

$2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0$..... (2.6)

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 (2.7).

$$p = \frac{x}{(x-1/2)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.7)$$

3. ANALISIS MASALAH

3.1. Analisis Citra Dijital

Agar *stego-image* dapat ditampilkan persis dengan aslinya, dalam melakukan steganografi, yang disisipi pesan hanya bagian *pixel data* saja karena jika bagian dari *file header, image header, dan color palette* ikut disisipi pesan, maka mungkin citra tidak dapat ditampilkan lagi. Hal tersebut menunjukan bahwa penyisipan pesan dengan teknik *LSB* hanya dapat dilakukan pada bagian *pixel data*. Oleh karena itu pendeteksian pesan rahasia juga hanya akan dilakukan pada bagian *pixel data*.

3.2. Analisis Algoritma Steganalisis

Dari berbagai kakas steganografi yang ada, kakas yang digunakan untuk pengujian program steganalisis yang dibangun adalah InPlainView, S-Tools, dan The Third Eye. Dasar pemilihan aplikasi-aplikasi di atas adalah aplikasi-aplikasi di atas menyediakan semua variasi dari algoritma *LSB*, yaitu menyisipkan pesan secara sekuensial dan acak.

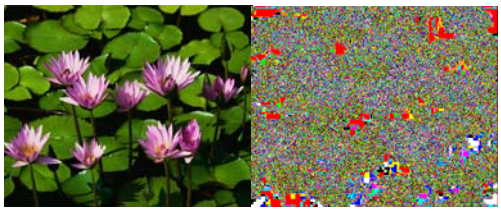
InPlainView(http://www.softpile.com/Utilities/Encryption/Download_05300_1.html) menyisipkan pesan dengan metode *LSB* secara sekuensial. S-tools (http://digitalforensics.champlain.edu/download) dapat menyisipkan pesan secara acak dengan enkripsi dan kompresi. Sedangkan kakas ketiga, The Third Eye (http://cs.uic.edu/%7Espopuri/tte/tte.zip) menyisipkan pesan secara acak.

3.2.1 Analisis *Enhanced LSB*

Keberhasilan metode ini didukung atas tingginya kontras gambar yang dipakai. Apabila steganalisis dilakukan dengan menggunakan gambar yang memiliki warna latar yang jelas atau memiliki kontras yang tinggi antara latar dengan gambar utama seperti **Gambar 3**, maka steganalis dapat memprediksi seperti apa gambar *enhanced LSB* yang seharusnya. Sedangkan untuk gambar yang nilai kontrasnya rendah, seperti gambar fotografis pada **Gambar 4**, teknik steganalisis dengan cara ini akan menyulitkan

steganalis. Karena steganalis akan kesulitan membedakan antara gambar yang seharusnya muncul dengan pesan rahasia.

Gambar 3 Citra kontras tinggi dan hasil *enhanced LSB*-nya [PAU07]



Gambar 4 Gambar dengan kontras rendah dan hasil *enhanced LSB*-nya

3.2.2 Analisis Uji *Chi-Square*

Proses analisis bit dengan metode uji *chi-square* adalah sebagai berikut:

1. Seluruh *byte* pada *pixel data* citra dibagi menjadi *n* kelompok, dimana *n* adalah (jumlah *byte* / *chunk*). *chunk* merupakan jumlah data yang akan dianalisis dalam satu iterasi, dalam hal ini nilai *chunk* adalah 128 *byte* karena sudah cukup untuk merepresentasikan kelompok *PoV* yang ingin diperiksa.
2. Masukkan hasil observasi sebuah *chunk* kedalam 128 kategori *PoV*. Sehingga diperoleh frekuensi dari setiap kategori warna (n_t).
3. Hitung frekuensi yang diprediksi apabila terjadi penyisipan pesan dari setiap kategori ($n_t^{\wedge}$) dengan menggunakan persamaan (2.1). Persamaan ini menghitung rata-rata dari dua frekuensi *PoV*.
4. Hitung nilai *chi-square* dari semua kategori dengan persamaan (2.2).
5. Pada tahap penghitungan probabilitas ada tidaknya pesan pada citra diadopsi dari ide Guillermito[GUI04]. Hitung nilai *normal chi-square* untuk derajat kebebasan *k - 1* dengan persamaan (3.1).

$$normal = \sqrt{2\chi^2} - \sqrt{(k-1)/2} \quad \text{.....(3.1)}$$

6. Probabilitas adanya pesan pada citra (*p*) dapat diperoleh dengan menggunakan tabel normal.
7. Tampilkan visualisasi grafik perbandingan.

3.2.3 Analisis *RS-Analysis*

Setiap langkah berikut akan dilakukan di masing-masing komponen warna (*red, green, blue*):

1. *Pixel* data dipartisi menjadi kelompok *n pixel* yang bertetangga, kelompok *pixel* K = (x_1,, x_2..., x_n).
2. Dilakukan penghitungan menggunakan fungsi diskriminasi *f* pada persamaan (2.4) untuk mengukur regularitas dari kelompok *pixel* K.
3. Dilakukan operasi *flipping* yang mendeskripsikan penyisipan dengan metode *LSB*, *fliping* ini menggunakan *mask* [0,1;1,0].
4. Dilakukan pengelompokan hasil operasi *flipping* kedalam tiga buah tipe, yaitu: *Regular, Singular,* dan *Unchanged.*
5. Dilakukan penghitungan estimasi panjang pesan dengan persamaan (2.6) dan (2.7) pada setiap komponen warna.
6. Tampilkan hasil estimasi panjang pesan pada setiap komponen warna.

3.2.4 Analisis Penghancuran Pesan

Penghancuran pesan dilakukan dengan mengganti seluruh bit LSB citra dengan bit 0. Penggantian seluruh bit *LSB* menjadi 0, tidak akan merusak tampilan citra, karena manusia tidak dapat membedakan perubahan yang terjadi pada bit *LSB*. Contoh citra yang memiliki pesan dan citra setelah pesan dihancurkan dapat dilihat pada **Gambar 6**.

Gambar 6 Citra dengan pesan rahasia (kanan) dan citra setelah pesan dihancurkan (kiri)

4. IMPLEMENTASI

Perangkat lunak yang dibangun diberi nama **`Steganeulis`**. Implementasi yang dilakukan menggunakan sebuah perangkat komputer untuk membangun **`Steganeulis`**. Perangkat komputer yang digunakan untuk melakukan implementasi memiliki spesifikasi sebagai berikut:

1. Prosessor Intel Core Duo 1.8 GHz
2. RAM 512 MB
3. Hard Disk 80 GB
4. Perangkat masukkan keyboard dan tetikus
5. Perangkat keluaran monitor

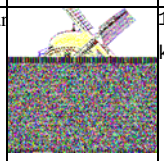
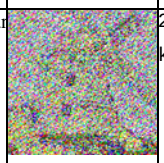
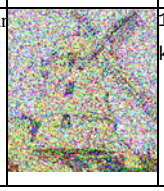
Adapun perangkat lunak yang digunakan dalam melakukan implementasi adalah sebagai berikut:

1. Sistem operasi Windows XP *Service Pack* 2
2. Microsoft Visual C# 2005 Express Edition
3. Bahasa C#

`Steganeulis` dibangun dengan menggunakan bahasa pemrograman C# dengan menggunakan *framework* .NET. Hasil implementasi diujikan pada komputer dengan lingkungan implementasi yang sama dengan yang digunakan dalam pengembangan.

3.PENGUJIAN

Tujuan dilakukannya pengujian adalah untuk memastikan bahwa semua kebutuhan perangkat lunak telah berhasil diimplementasikan dengan baik dan untuk mengetahui performansi dan akurasi dari perangkat lunak yang telah dibangun. Penggujian dilakukan pada citra `windmill.bmp` yang dapat dilihat pada **Gambar 6,** dengan pesan berukuran kecil (0.79 kb) dan ukuran besar (13.8 kb). Berikut adalah hasil pengujiannya:

No	Kategori	Pesan	Enhanced LSB	Chi-Square	RS-Analysis
1	Tidak ada pesan			Diduga tidak ada pesan	**Red : 11%** **Green: 11%** **Blue : 11%** **Total:2.5 kb**
2	Pesan disisipkan secara sekuensial	Kecil		0.79 kb	**Red : 11%** **Green: 11%** **Blue : 11%** **Total:2.5 kb**
		Besar		13.9 kb	**Red : 11%** **Green: 11%** **Blue : 11%** **Total:2.5 kb**
3	Pesan disisipkan secara acak	Kecil		Diduga tidak ada pesan	**Red : 17%** **Green: 17%** **Blue : 17%** **Total:3.74 kb**
		Besar		20.9 kb	**Red : 78%** **Green: 78%** **Blue : 79%** **Total: 16.62 kb**
4	Pesan disisipkan secara acak dengan kompresi dan enkripsi	Kecil		Diduga tidak ada pesan	**Red : 13%** **Green: 13%** **Blue : 13%** **Total : 2.91 kb**
		Besar		15.1 kb	**Red : 57%** **Green: 57%** **Blue : 56%** **Total: 12.03 kb**

Hasil pengujian metode *enhanced LSB* di atas menunjukan bahwa **`Steganeulis`** dapat menjalankan steganalisis dengan *enhanced LSB* dengan baik. Akan tetapi steganalisis dengan cara ini masih membutuhkan bantuan manusia untuk memutuskan ada tidaknya pesan rahasia pada citra yang diamati. ada citra kontras tinggi, pengguna dapat dengan mudah membedakan pesan rahasia dengan citra yang sewajarnya muncul. Oleh karena itu metode *enhanced LSB* sangat sesuai dengan citra berkontras tinggi.

Berdasarkan hasil pengujian metode uji *chi-square* di atas, ditunjukan bahwa **`Steganeulis`** dapat menjalankan steganalisis dengan uji *chi-square* dengan baik. Metode ini dapat menunjukan estimasi panjang pesan secara akurat apabila disisipkan secara sekuensial. Contohnya pada nomor 2, pesan berukuran kecil dideteksi sebesar 0.79 kb dan pesan berukuran besar dideteksi sebesar 13.9 kb, yang menunjukan angka yang sangat akurat.

Steganalisis dengan cara ini hanya akurat untuk penyisipan secara sekuensial. Hal ini disebabkan oleh bit-bit pesan yang terkonsentrasi pada suatu bagian, akan mengakibatkan distribusi bit pesan merata pada satu *chunk* yang membuat distribusi frekuensi *PoV* mendekati rata-rata. Apabila distribusi *PoV* mendekati rata-rata, maka akan menyerupai distribusi frekuensi yang diprediksi akan dimiliki oleh sebuah citra yang telah disisipi pesan.

Pengujian metode *RS-analysis* menunjukan bahwa **`Steganeulis`** dapat menjalankan steganalisis dengan *RS-analysis* dengan baik. Pada citra yang tidak mengandung pesan, hasil steganalisis menunjukan adanya pesan rahasia sebesar 11%, sebenarnya angka yang ditunjukan ini merupakan bias awal citra yang disebabkan oleh *noise* pada citra. Oleh karena itu *noise* dapat mengurangi akurasi metode ini.

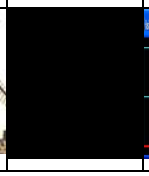
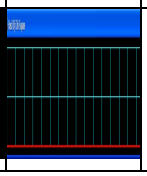
Kemudian pada penyisipan pesan secara sekuensial, hasil steganalisis menunjukan angka yang jauh dari panjang pesan sebenarnya. Oleh karena itu steganalisis dengan metode ini tidak cocok dengan penyisipan secara sekuensial.

Metode ini dapat menunjukan estimasi panjang pesan dengan akurat apabila disisipkan secara acak. Contohnya pada penyisipan secara acak di atas, lihat nomor ke-tiga dan ke-empat. Pada baris ketiga ditunjukan bahwa perkiraan panjang pesan adalah sekitar 16.62 kb dimana pesan yang disisipkan pada citra adalah 13.8 kb, selisih sekitar 2.5 kb ini merupakan akibat dari bias awal yang ditunjukan pada baris pertama. Jadi hasil steganalisis ini cukup akurat.

Sedangkan pada baris keempat, sebelum pesan disisipkan secara acak dilakukan kompresi dan enkripsi terlebih dahulu sehingga ukuran pesan menjadi lebih kecil, oleh karena itu hasil steganalisis menunjukan ukuran pesan adalah 12.03 kb.

Apabila citra memiliki kontras rendah, dengan ukuran pesan yang sangat kecil, dan disisipkan secara acak, maka steganalis akan kesulitan dalam menentukan ada tidaknya pesan pada citra. Karena steganalis akan kesulitan dalam menentukan apakah panjang pesan yang diestimasi merupakan panjang pesan yang sebenarnya atau merupakan bias awal.

Pengujian penghancuran pesan ditujukan untuk memeriksa apakah pesan rahasia benar-benar hilang setelah dihancurkan dan apa yang terjadi apabila citra yang pesannya telah dihancurkan disteganalisis. Pengujian ini dilakukan dengan mengekstraksi pesan menggunakan kakas steganografi dan melakukan steganalisis pada citra tersebut.

Citra Setelah Pesan Dihancurkan	***Enhanced LSB***	***Uji Chi-Square***	***RS-Analysis***
			**Red : 0%** **Green : 0%** **Blue : 0%** **Total :0 kb**

Setelah pesan pada citra dihancurkan, tampilan citra tidak mengalami kerusakan seperti yang ditunjukan pada kolom pertama. Setelah citra tersebut disteganalisis dengan metode *enhanced LSB* seluruh citra menjadi berwarna hitam, karena semua bit *LSB* bernilai 0. Ketika citra tersebut disteganalisis dengan uji *chi-square* tidak terdapat titik-titik berwarna hijau yang menunjukkan rata-rata *LSB* dan peluang adanya pesan rahasia bernilai 0, hal ini disebabkan seluruh bit *LSB* bernilai 0. Dan hasil *RS-analysis* pun menunjukan tidak ada pesan rahasia pada citra. Ketiga hasil steganalisis ini dapat menjadi penanda bahwa citra pernah memiliki pesan rahasia dan telah dihancurkan.

Waktu yang dibutuhkan untuk melakukan *enhanced LSB* adalah sekitar 0.01 detik per kilobyte. Waktu yang dibutuhkan untuk uji *chi-square* adalah sekitar 0.008 detik per kilobyte. Dan waktu yang dibutuhkan untuk mengestimasi panjang pesan dengan *RS-analysis* adalah sekitar 0.04 detik per kilobyte. Hal tersebut menunjukan bahwa uji *chi-square* adalah metode steganalisis yang tercepat dan *RS-analysis* adalah yang terlama pada perangkat lunak ini. Sedangkan waktu yang dibutuhkan untuk menghancurkan pesan adalah sekitar 0.01 detik per kilobyte.

Hasil analisis menunjukan tujuan pengujian telah tercapai. Telah dipastikan bahwa semua kebutuhan perangkat lunak telah berhasil diimplementasikan dengan baik dan telah diketahui bahwa waktu yang dibutuhkan perangkat lunak dalam melakukan steganalisis cukup singkat dan diketahui pula bahwa akurasi pendeteksian pesan rahasia cukup tinggi untuk kasus-kasus yang sesuai dengan metode steganalisisnya

4. KESIMPULAN

Dari keseluruhan isi makalah ini, dapat diambil kesimpulan sebagai berikut:

1. Sebuah perangkat lunak untuk steganalisis metode *LSB* pada citra telah berhasil dibangun. Perangkat lunak yang dibangun tersebut dapat melakukan pendeteksian pesan rahasia dan penghancuran pesan dengan baik. Perangkat lunak tersebut menggunakan algoritma *enhanced LSB*, uji *chi-square*, dan *RS-analysis* untuk pendeteksian.
2. Algoritma *enhanced LSB* dapat melakukan steganalisis pada citra yang telah disisipkan pesan baik secara sekuensial ataupun acak, akan tetapi keberhasilannya tergantung pada tingginya kontras citra. Semakin tinggi kontras citra maka semakin mudah untuk mendeteksi pesan.
3. Algoritma uji *chi-square* dapat melakukan steganalisis dengan baik pada citra yang disisipi pesan secara sekuensial dan mengestimasi panjang pesan yang dikandung.
4. Algoritma *RS-analysis* dapat melakukan steganalisis dengan baik pada citra yang disisipi pesan secara acak, bahkan dapat mengestimasi panjang pesan dengan akurat.
5. Penghancuran pesan tidak akan merusak citra dan dapat digunakan sebagai penanda bahwa citra pernah memiliki pesan rahasia dan pesan tersebut telah dihancurkan.

5. SARAN

Saran yang diajukan untuk pengembangan lebih lanjut ialah:

1. Agar pesan rahasia dapat diketahui, diterapkan sebuah algoritma yang dapat mengekstraksi pesan dari *stego-object*. Ekstraksi pesan mungkin dilakukan apabila pesan disisipkan secara sekuensial dengan cara mengekstrak bit-bit *LSB* dari citra dan merangkai bit-bit tersebut.
2. Dalam tugas akhir ini diterapkan tiga algoritma steganalisis, yaitu: *enhanced LSB,* uji *chi-square,* dan *RS-analysis.* Untuk meningkatkan kemampuan perangkat lunak dalam mendeteksi pesan rahasia, dapat diterapkan berbagai algoritma steganalisis lainnya, seperti : *pairs analysis, fusion technique, universal steganalysis,* steganalisis dengan pendekatan *machine learning,* dan sebagainya.
3. Steganalisis dapat diterapkan pada berbagai jenis arsip lainnya, seperti : *JPEG, GIF, PNG, TIFF* dan sebagainya.
4. Algoritma steganalisis *enhanced LSB* dibuat menjadi otomatis sehingga tidak perlu menggunakan bantuan manusia untuk memutuskan ada tidaknya pesan

DAFTAR REFERENSI

[FRI02] Fridrich, J., Goljan, M. (2002) Practical Steganalysis of Digital Images – State of the Art.

[GUI04] Guillermito. (2004). A Few Tools to Discover Hidden Data : *http://guilermito.com/* diakses pada bulan Mei 2008.

[GUP05] Gupta, Sonali. (2005). Steganalysis. *http://www.palisade.plynt.com* diakses pada bulan Maret 2008.

[KHA06] Kharrazi, Mehdi., Sencar, Husrev T., Memon, Nasir. (2006). Improving Steganalysis by Fusion Techniques: A Case Study with Citra Steganography.

[MIA99] Miano, John. (1999). Compressed Image File Formats.

[MIT03] Mitra, S., Roy, T., Mazumdar, D., Saha, A.B. (2003). Steganalysis of *LSB* Encoding in Uncompressed Citras by Close Colour Pair Analysis.

[MOR05] Morkel, T., Eloff, J.H.P., Olivier, M.S. (2005). An Overview of Citra Steganography.

[MUN06] Munir, Rinaldi. (2006). Diktat Kuliah IF5054 Kriptografi.

[PAU07] Paul Gunawan. (2007). Studi dan Analisis Mengenai Teknik Steganalisis Terhadap Pengubahan *LSB* Pada Gambar: *Enhanced LSB* dan *Chi-square.*

[RAG00] Raggo, Michael T.(2000). Steganography, Steganalysis, Cryptanalysis.

[WEN00] Wen Chen. (2000) Study Of Steganalysis Methods.

[WES99] Westfeld, A., Pfitzmann, A. (1999). Attacks On Steganographic System.