ANALISIS DAN STUDI KASUS PERTAHANAN TERHADAP SERANGAN STEGANALISIS YANG MENGGUNAKAN TEKNIK ANALISIS STATISTIK

Anna Maria J S – NIM : 13503075

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13075@students.if.itb.ac.id

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan tersebut tidak bisa diketahui. Ciri utama dari steganografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbulkan kecurigaan bagi pihak ketiga.

Makalah ini berisi metode yang digunakan pada steganalisis untuk menyembunyikan *embedded message* (pesan rahasia) sehingga keberadaan *stegotext* semakin tidak menimbulkan kecurigaan. Ada 2 metode yang akan dibahas pada makalah ini, yaitu dengan cara analisis probabilistik dan *error-correcting*. Analisis statistik adalah metode yang menggunakan analisis probabilistik untuk menentukan bagian *covertext* (pesan yang digunakan untuk menyembunyikan pesan rahasia) yang akan dimodifikasi sehingga modifikasi dapat terjadi secara minimal. Metode selanjutnya adalah *error-correcting*. Metode ini adalah metode menentukan bit-bit mana yang akan dimodifikasi. Bit-bit yang dipilih adalah bit-bit yang sulit untuk terdeteksi dan mirip dengan bit yang menggantikannya.

Kedua metode yang akan dijelaskan merupakan metode yang digunakan untuk mengatasi serangan terhadap steganalisis dengan menggunakan metode statistical. Pada makalah ini juga akan dijelaskan mengenai bagaimana metode analisis statistik dapat mendeteksi adanya *embedded message* pada suatu berkas. Untuk lebih memperjelas metode yang digunakan, maka dijelaskan contoh kasus dengan menggunakan berkas dengan format .jpeg sebagai *stegotext* untuk menyimpan pesan.

Kata kunci : kriptografi, steganalisis, *statistical analysis*, enkripsi, dekripsi, *stegotext, covertext, embedded message*

1. Pendahuluan

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan tersebut tidak bisa diketahui. Ciri utama dari steganografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbulkan kecurigaan bagi pihak ketiga. Selain itu, unser penting lainnya yang diperlukan di dalam steganografi adalah adanya kunci rahasia. Bila kunci rahasia ini diketahui oleh pihak luar, maka keberadaan *embedded message* pada suatu berkas dapat dilacak.

Namun ada kalanya, tanpa mengetahui kunci rahasia, pihak luar mampu melacak adanya sebuah *embedded message* yang tersimpan pada *file* tertentu.. Salah satu metode yang digunakan untuk melacak keberadaan steganalisis ini adalah dengan menggunakan metode statistika. Secara singkat, metode ini melacak adanya anomali statistik pada sebuah *file* dan dapat membuktikan bahwa pada file tersebut adalah *stegotext* file yang menyimpan pesan rahasia).

Pada steganalisis, proses pemimpanan informasi dilakukan dengan menggunakan dua cara, yaitu:

1. Mengidentifikasi *redundant bits* pada *cover médium. Redundant bits* adalah bit-bit yang dapat diganti tanpa

mengubah integritas ataupun kualitas *covertext.*

2. Memilih bagian *redundant bits* yang dapat diganti dengan *message bits*. Bit-bit yang dipilih adalah bit-bit yang yang sulit terdeteksi atau mirip dengan bit-bit yang akan menggantikannya.

Makalah ini memaparkan dua metode yang dapat digunakan untuk memilih bagian *redundant bits*. Metode yang pertama adalah dengan menggunakan pseudo random generator (PRNG). Setiap PRNG akan menghasilkan himpunan *bit* yang dapat digantikan oleh *message bits*. Himpunan bit yang jika digunakan untuk menyembunyikan *message bit* akan mengakibatkan paling sedikit terjadi perubahan pada *cover medium* akan digunakan untuk proses *embedding*. Metode kedua adalah dengan menggunakan *error-correcting codes*. Kedua metode ini dapat digunakan secara bersamaan untuk meningkatkan kerahasiaan dalam menyimpan *message bits*.

Walaupun demikian, setiap modifikasi yang dilakukan pada *redundant bits* dapat mengakibatkan perubahan *statistical property* dari *cover medium.* Sebagai contoh *statistical property* adalah perbandingn bit-bit 0 dan 1 pada *redundant bits*-nya. Proses *embedding* dapat merusak perbandingan tersebut, sehingga diperlukan suatu metode lain untuk menjaga nilai *statistical property* tetap. Cara yang digunakan adalah dengan mengecilkan ukuran *hidden message*. Metode ini dilakukan dengan melakukan perubahan-perubahan khusus pada *redundant bits* untuk memperbaiki deviasi yang terjadi akibat perubahan *redundant-bits*. Dengan cara ini pula, kemampuan *stego medium* untuk menyimpan pesan tidak akan berubah, sehingga pihak luar menduga bahwa tidak terjadi perubahan apapun pada *stegotext*.

Makalah ini akan membahas kedua metode yang diterapkan untuk menyimpan pesan, contoh implementasi metode tersebut pada berkas dengan tipe .jpeg, metode yang dapat digunakan untuk memperbaiki deviasi yang terjadi akibat perubahan pada *redundant bits*, serta kesimpulan dan analisis segala metode yang telah diterapkan.

2. Syarat-syarat Steganalisis

Pada steganografi, terdapat beberapa syarat yang harus terpenuhi untuk mencegah *stegotext* terdeteksi oleh pihak yang tidak berkepentingan. Syarat-syarat yang harus terpenuhi itu adalah:

1. *Cover medium* yang belum termodifikasi tidak boleh terlihat oleh pihak ketiga. Karena, jika pihak luar membandingkan berkas *cover medium* dengan *stegotext*, keberadaan *message bits* akan segera terdeteksi.
2. Kunci rahasia tidak diketahui oleh pihak luar.

Pada kenyataannya, walaupun cover medium tidak diketahui oleh pihak luar, namun stegotext mungkin memiliki distorsi yang dapat dilacak. Karena itulah dilakukan transformasi setelah dilakukan proses *embedding*, sehingga, distorsi yang ada dapat dihilangkan.

3. Proses *Embedding*

Transformasi yang dilakukan untuk menerapkan *statistical correction* bergantung pada metode pendistribusian *hidden message* pada *redundant bits* yang terdapat pada *cover message*. Bab ini menjelaskan proses *embedding message bits* dengan metode yang digunakan untuk meningkatkan kerahasiaannya.

Metode untuk melakukan proses *embedding message bit* pada cover medium dibagi ke dalam 2 langkah, yaitu:

1.1 Identifikasi redundant bits.
Redundant bits adalah *bit* yang dapat dimodifikasi tanpa mengakibatkan penurunan kualitas *cover medium.* Modifikasi terhadap *redundant bit* juga tidak menyebabkan *cover medium* yang sudah termodifikasi menjadi lebih mudah dilacak.

2.1 Proses seleksi bit-bit yang akan digantikan oleh *hidden message*.

Proses identifikasi *redundant bits* adalah proses yang bergantung pada jenis format data yang akan dimodifikasi. Sedangkan proses seleksi hanya menerima hasil identifikasi *redundant bits*. Proses seleksi ini tidak bergantung kepada format atau proses apa yang dilakukan untuk identifikasi redundant bits, sehingga cost untuk melakukan seleksi tidak bergantung kepada *cost* untuk melakukan identifikasi *redundant bits*.

Pembagian *embedding process* menjadi dua langkah memiliki suatu kelemahan, yaitu kemungkinan kehilangan informasi yang diperlukan untuk proses seleksi. Contohnya, pada *redundant bit* yang dipilih mungkin terdapat area yang dapat menampung lebih banyak *hidden information* atau area tersebut adalah area yang sulit untuk terdeteksi. Pada model ini, algortma seleksi hanya melihat *redundant bit* yang ada dan tidak menghiraukan aslinya. Akibatnya, informasi yang penting tadi hilang begitu saja. Untuk mengatasi hal ini langkah identifikasi harus disertakan dengan menambahkan atribut pada setiap *redundant bits*. Atribut ini mengindikasikan apakah sebuah bit di-*lock* atau dapat berupa keterangan bagaimana kemungkinan terdeteksi bila dilakukan perubahan pada bit tersebut.

3. Metode-metode Lain di Dalam Steganalisis

Pada bab ini akan dijelaskan mengenai metode-metode lain yang digunakan pada steganalisis selain yang digunakan pada contoh-contoh selanjutnya pada makalah ini.

Beberapa metode ini adalah:

1. Metode Walton
   Metode Walton menjaga autentifikasi dengan menyimpan jumlah *redundant bits* pada berkas gambar. Penjumlahan ini didistribusikan secara tidak merata ke seluruh area gambar dengan menggunakan pseudo-random number generator (PRNG), yaitu sebuah metode yang digunakan utnuk menghasilkan angka yang acak, yang selanjutnya akan digunakan untuk mendistribusikan. Probabilitas yang digunakan pada makalah ini juga dilakukan dengan menggunakan PRNG, namun terdapat perbedaan, yaitu pada metode yang digunakan di makalah ini, nilai PRNG dibangkitkan dengan memilih bit-bit *seed* (dasar). Hal ini dilakukan untuk mengurangi jumlah modifikasi yang harus dilakukan pada *stego-medium.*

2. Metode Aura
   Metode Aura menggunakan pseudo Random Permutation Generator untuk memilih bit-bit pada *cover-medium.* Dengan metode ini diperoleh bahwa untuk kunci yang sama dan *cover-medium* tidak berubah, maka bit-bit yang diseleksi akan sama. Pada proses embedding yang dilakukan di makalah ini, PRNG dibangkitakan dengan memilih bit-bit *seed* (dasar). Hal ini mengakibatkan bit-bit yang selanjutnya akan dipilih tidak bergantung kepada bit-bit yang sebelumnya telah dipilih. Sebagai tambahan bit-bit yang dipilih pada makalah ini bergantung kepada ukuran *message bits* yang akan disimpan.

3. Metode Johnson dan Jajodia
   Metode ini menganalisis gambar dengan membuat perangkat lunak steganografi. Walaupun hasil analisis yang dilakukan menunjukkan adanya distorsi koefisien DCT, namun mereka tidak membahas lebih lanjut mengenai distorsi tersebut.

4. Westfed dan Pfitzman menunjukkan cara-cara yang dilakukan pada steganografi pada umumnya untuk meng-*embed* data pada berkas dengan format JPEG. Untuk mendeteksi adanya pesan rahasia, mereka menggunakan X^2 dengan menmperkirakan distribusi warna pada gambar yang membawa informasi rahasia dan membandingkannya dengan distribusi yang diperoleh lewat penelitian.

3.1 Identifikasi Redundant Bits

Secara imum identifikasi terhadap *redundant bits* tergantung kepada format outputnya. Proses embedding dapat dilakukan jika *cover media* sama dengan format *output* yang diinginkan. Konversi ke format data akhir mungkin membutuhkan operasi khusus seperti kompresi. Untuk meminimalkan modifikasi pada cover medium membutuhkan pengetahuan tentang *redundant bits*, sebelum data tersebut dimodifikasi. Proses identifikasi *redundant bits* dapat dilakukan pada saat cover medium akan diubah ke dalam format yang sama dengan stegotext.

Message bit dapat disalin pada *redundant bits* pada saat *file* output telah selesai dibuat. Karena itu proses konversi *stego medium* ke dalam format yang diinginkan harus mengikuti kaedah tertentu yang harus bisa dimengerti pada saat melakukan identifikasi *redundant bits*. Prinsip ini dapat dipenuhi dengan memanfaatkan sebuah *pseudo-random number generator* yang diinisialisasikan pada saat identifikasi dan pada saat konversi *stego medium.*

Sebelum bit-bit yang sudah teridentifikasi diseleksi pada *selection step*, bit-bit tersebut dikenakan informasi tambahan. Informasi ini dapat berupa *locked bits*, yaitu bit yang mungkin tidak boleh dimodifikasi pada saat *embedding process* atau sebuah pendekatan *heuristic* yang menentukan bagaimana kemungkinan terdeteksi sebuah bit bila dikenakan perubahan terhadapnya.

Sebuah bit di-lock bila bit tersebut sudah membawa *message bits*. Hal ini bisa terjadi jika message yang disimpan pada cover medium tersebut lebih dari satu.

3.2 Selection of Bits

Sebelum proses seleksi dilakukan, maka sebuah RC4 *stream cipher* diinisialisasikan dengan menggunakan sebuah kunci yang dimasukkan oleh *user*. *Stream cipher* yang berkunci tersebut digunakan untuk mengenkripsikan *hidden message* dan menentukan PRNG (*pseudo random number generator*) sebagai awal proses seleksi.

Proses pergantian bit dengan hidden message dengan memanfaatkan PRNG dilakukan dengan cara sebagai berikut:

Pertama, kita perlu menyimpan 32 bit *state* (awal). Bit ini merupakan konkatenansi antara 16 bit *seed* (dasar) dan 16 bit bilangan bulat yang menyimpan panjang *hidden message*. Agar proses *embedding* semakin baik, maka proses seleksi dilakukan dengan memvariasikan *bit-bit seed*. Bit yang pertama sekali dipilih untuk digantikan adalah bit awal pada *redundant bits* yang sudah teridentifikasi sebelumnya.

Penentuan bit selanjutnya adalah dengan menghitung *random offset* pada interval tertentu dan menjumlahkan *offset* tersebut dengan posisi bit sekarang. Untuk menghitung *random offset*, digunakan *PRNG* yang sudah dijelaskan sebelumnya. Setiap data pada posisi bit yang baru digantikan oleh *message bit*. Proses ini dilakukan 32 kali. Proses penentuan bit selanjutnya yang akan diseleksi dapat dipresentasikan sebagai berikut:

$$b_o = 0$$
$$b_i = b_{i-1} + R_i(x) \text{ untuk } i = 1,...,n$$

dimana b_i adalah posisi dari bit seleksi yang ke – i dan $R_i(x)$ adalah *random offset* dalam interval [i, x].

Setelah data berhasil di-embed, maka PRNG dikembalikan lagi ke awal dengan menggunakan bit-bit *seed* tadi. Sisa panjang dari *message bits* digunakan untuk mengukur *interval*. Interval yang dimaksud adalah jarak tertentu dimana *random number* tersebut akan digunakan lagi pada *remaining data* (sisa *redundant bits* yang belum disubstitusi oleh *message bits*), sehingga *message bits* akan tersebar secara merata pada *stego medium.*

$$\text{interval} \approx \frac{2 \times \text{remaining redundant bits}}{\text{remaining length of message}}.$$

Proses seleksi dilanjutkan seterusnya hingga *message bits* selesai diproses dengan menggunakan cara diatas, perbedaannya hanyalah interval akan berubah setiap 8 bit. Dengan cara ini, *message bits* akan didistribusikan dengan merata di sepanjang bit-bit yang tersedia.

Pemilihan interval dengan cara seperti ini mengakibatkan *message bits* hanya bisa mengisi maksimum 50% dari *redundant bit* yang tersedia. Dengan tidak menggunakan semua *redundant bit* dapat memberikan kesempatan yang lebih besar bagi proses seleksi untuk menemukan *embedding* yang lebih baik, karena hal ini juga menyebabkan tersedianya cukup bit-bit yang dapat digunakan untuk mengkoreksi transformasi demi mengatasi serangan dengan menggunakan perhitungan statistik.

Karena PRNG dikunci dengan menggunakan sebuah kunci rahasia, maka tidak mungkin untuk mengetahui *message bits* tanpa mendapatkan kunci tersebut. Penerima

menginisialisasi PRNG dengan kunci rahasia tersebut dan menggunakan proses seleksi yang sama untuk mendapatkan *message bits* dari *stego medium.* Ukuran interval hanya akan berubah hanya setelah bit-bit berhasil di-*embed*, sehingga dapat diperoleh kembali dan dapat digunakan lagi untuk mendapatkan kembali PRNG.

3.3 Keuntungan *Reseeding* (membangkitkan nilai PRNG dari bit-bit *seed*)

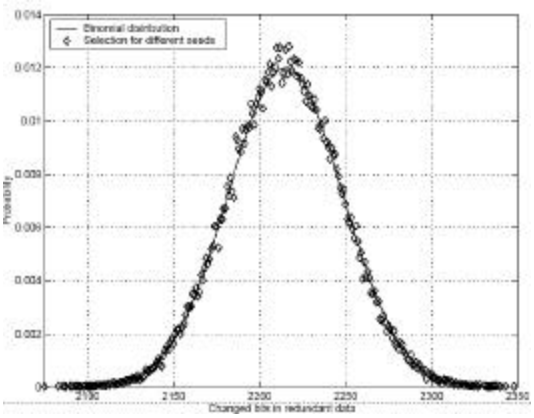
Pada subbab ini dijelaskan mengapa modifikasi terhadap cover medium dapat dikurangi dengan menggunakan algoritma seleksi seper yang dipaparkan subbab sebelumnya.

Pada proses seleksi, pseudo-random number generator dapat dihasilkan kembali dengan menggunakan 16 bit *seed* yang dipilih oleh orang yang mengenkripsi. Pada efeknya, setiap *seed* akan menghasilkan *pseudo random generator* yang unik untuk setiap bit. Setiap *pseudo random generator* akan memilih sendiri bagian himpunan yang akan diubah. Jumlah anggota himpunan yang dihasilkan juga berbeda untuk masing-masing PRNG. Distribusi perubahan yang dihasilkan bersifat binomial. Probabilitas bahwa k-bit dari panjang n bit *redundant bits* dapat diubah diberikan oleh persamaan berikut:

$$p_k^{(n)} = \binom{n}{k} p^k (1-p)^{n-k},$$

dimana p adalah probabilitas/kemungkinan sebuah bit dapat diubah.

Karena *message bits* soenkripsi dengan menggunakan RC4, maka nilai p = ½ sehingga diperoleh rata-rata perubahan yang terjadi pada *message bits* setelah dikenakan proses enkripsi adalah 50%.



Gambar 1. Distribusi probabilitas banyaknya bit yang dapat diubah dengan menggunakan *seeds* yang berbeda-beda dengan jumlah n = 4430 dan p = ½.

Dari data probabilitas di atas diperoleh kesimpulan bahwa dengan memilih seed pada bagian bawah diagram akan mengurangi jumlah bit yang harus dimodifikasi. Hal inilah yang menyebabkan, PRNG dipilih berdasarkan *seed* yang dipilih oleh orang yang mengenkripsi. Dengan cara demikian, akan semakin sulit untuk mendeteksi adanya *message bits* karena bit-bit tersebut secara alamiah telah direpresentasikan pada *cover medium.*

Detactibility (kecenderungan untuk terlacak) merupakan hal yang menjadi pertimbangan dalam menentukan bit-bit yang akan diseleksi.Pemilihan bit tidak hanya mencoba mengurangi jumlah bit yang akan diubah tetapi juga nilai *detactibility* secara keseluruhan. Setiap perubahan bit akan mengakibatkan perubahan nilai detactibility secara keseluruhan. Semakin rendah tingkat *detactibilty* menunjukkan bahwa *covertext* semakin mirip dengan *cover medium*-nya sehingga kemungkinan dilacak semakin kecil.

Standar deviasi (tingkat perubahan yang mungkin terjadi) dinyatakan dengan $\sigma = \sqrt{npq}$ dan q = 1 - p, pada kasus ini $\sigma = \frac{1}{2}\sqrt{n}$

Bits n	Standard Deviation σ Expected	Measured	Probability $\sigma/\sqrt{n}$
1216	17.436	16.193	0.464
2400	24.495	24.254	0.495
4176	32.311	32.205	0.498
6840	41.352	40.163	0.486
9800	49.497	44.257	0.447
14832	60.893	53.123	0.436

Tabel 1 Perhitungan probabilitas p yang menyatakan kemungkinan bit yang diseleksi harus diubah. *Message bits* yang digunakan adalah message bits dengan panjang n dan dicobakan pada berkas dengan format JPEG dan 77135 *redundant bits* terdapat di dalamnya.

Dari tabel, dapat disimpulkan kemungkinan terdeteksi terhadap bit-bit yang diseleksi sebenarnya hampir sama, karena berkisar antar 0.4-0.5. Untuk mengecilkan kemungkinan tersebut, cara yang dapat dilakukan adalah dengan membagi-bagi *message bits* menjadi *message bit-message bits* yang kecil-kecil.

Akhirnya, secara umum, proses seleksi dengan menggunakan PRNG dapat menghindari kemungkinan penggunaan *locked bits*, Sedangkan dengan memperhitungkan deviasi yang mungkin terjadi dapat diperoleh data deviasi yang menunjukkan probabilitas perubahan yang harus dilakukan pada bit-bit yang harus diseleksi.

3.4 *Error Correcting*

Pada dua sub-bab sebelumnya telah dipaparkan mengenai metode seleksi dengan menggunakan PRNG dan keuntungan penggunaan metode di dalam menjaga kerahasiaan *message-bits* yang akan disimpan.

Penggunaaan metode seleksi seperti di atas memungkinkan proses embedding dengan representasi probabilistik, namun fleksibilitas dari PRNG dalam memilih bit-bit tidak selalu cukup untuk menghindari semua bit yang terkunci atau memiliki kemungkinan yang besar untuk dideteksi. Perlu diterapkan suatu metode untuk mencegah modifikasi bit-bit yang sudah di-*lock* atau besar kemungkinan untuk terdeteksi dengan cara menambahkan *error* pada hidden message tanpa merusak *content*. Pada akhirnya, semua *error* harus dikoreksi.

Teori *coding* menyediakan *code* yang dapat memperbaiki *error* dengan melakukan *decoding* (proses untuk menghasilkan *covertext* lagi) yang kemiripannya paling besar.

Pada saat data sudah diubah ke dalam bentuk *stegotext*, kita dapat memilih t bit pada setiap block yang merupakan bit-bit yang tidak bermasalah, sehingga tidak perlu diperbaiki. Pemilihan bit-bit ini tergantung apakah bit tersebut di-lock.

Dapat ditulis [n, k, d] untuk menunjukkan k-dimensi linear dari *code* dengan panjang n dan *Hamming distance* senilai d. Code seperti ini dapat memperbaiki t *errors*, dimana d = 2t +1.

Secara umum, aplikasi dari *error-correcting* dapat meningkatkan jumlah data yang harus di-*embed*. Data dengan panjang k akan menghasilkan n-bit *code block*. Dari penelitian diperoleh bahwa sekitar setengah dari data sudah direpresentasikan dan tidak perlu dimodifikasi dan misalnya diasumsikan adanya t errors pada code block. Maka, dapat dituliskan

$$\frac{n}{2} - t = \frac{k}{2}$$

dari persamaan di atas diperoleh:

$$d = n - k + 1$$

Dari persamaan di atas juga dapat disimpulkan bahwa jumlah bit yang harus diperbaiki sama dengan jumlah bit yang tidak harus diperbaiki.

3.5 Penyerangan yang Masih Mungkin Dilakukan untuk Mendeteksi Keberadaan *Stegotext*.

Untuk menyimpan *message bits* pada *cover medium*, dapat dilakukan modifikasi *redundant data* pada cover medium. *Redundant data* mungkin memiliki properti statistik yang tidak kita sadari atau property yang lebih dis adari oleh musuh. Jika proses *embedding* telah merusak karakteristik dari *cover medium*, maka peneliti yang memiliki pengetahuan lebih banyak akan mudah menyimpulkan kehadiran

message bits tanpa perlu menunjukkan bit-bit mana yang telah diubah.

Message bits mungkin saja segera diketahui. Namun karena hal itu sangat sulit untuk dilakukan, kita mengasumsikan bahwa yang paling mungkin dilakukan oleh musuh adalah menyimpulkan fakta bahwa *cover medium* telah dimodifikasi.

Hal ini dapat diatasi dengan menyertakan message bits lain pada cover mediium untuk menutupi keberadaan *message bits* yang sebenarnya. Yang harus diperhatikan hanyalah penambahan message bit yang beru tidak dilakukan pada bit-bit yang telah diubah. Dengan cara demikian, pihak musuh tidak akan berusaha untuk melacak pesan yang sesungguhnya terdapat pada *stegotext*.

Selain itu untuk meningkatkan keamanan, pengirim pesan dapat terlebih dahulu memilih jenia *cover medium* dengan format yang memungkinkan minimalisasi modifikasi yang harus dilakukan pada *cover medium.*

4. Format Gambar JPEG

Bab-bab berikutnya akan sangat berkaitan dengan penerapan metode yang dijelaskan di atas dan contoh statistical analysis yang mungkin dilakukan pada gambar dengan format JPEG. Karena itu, pada bab ini akan dijelaskan mengenai gambar dengan format JPEG.

Setiap format data yang dapat dijadikan sebagai *cover medium* memiliki *statistical property* yang berbeda-beda, sehingga proses *embedding* yang dilakukan juga bergantung kepada format data yang akan diolah . Namun, pada makalah ini proses embedding dilakukan pada format data JPEG saja, sehingga tidak ada pembahasan antara jenis format data dan proses embedding yang dilakukan terhadap format data tersebut.

Pada dasarnya untuk setiap format data proses *correcting statistical deviations* adalah sama, perbedannya hanya terletak pada jenis transformasi yang dilakukan.

Format data JPEG menggunakan *Descrete Cosine Transform* (DCT) untuk mengubah setiap block 8x8 pixel menjadi 64 DCT *coefficient*. Koefisien DCT, F(u,v) dari 8x8 blok *image pixel* f(x,y) diberikan sebagai berikut :

$$F(u,v) = \frac{1}{4}C(u)C(v)\Big[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)* \cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\Big],$$

dimana C(u), C(v) = $1/\sqrt{2}$. Bila u dan v = 0 , maka C(u) dan C(v) = 1.

Hasil diatas merupakan DCT koefisien yang selanjutnya akan digunakan untuk menghitung 64 DCT *coefficient* untuk masing-masing píxel.

Proses perhitungan koefisien adalah sebagai berikut:

$$F^{Q}(u,v) = \text{IntegerRound}\Big(\frac{F(u,v)}{Q(u,v)}\Big),$$

dimana Q(u,v) adalah tabel terdiri dari 64 elemen yang akan diubah.

Least Significant Bits pada berkas berformat TPEG adalah bit-bit yang nilai $F^{Q}(u,v)$ tidak sama dengan 0 maupun 1. LSB ini digunakan sebagai *redundant bits* untuk menyembunyikan data yang akan disimpan.

5. Test Statistik

Dengan menggunakan tes Statical dapat dibuktikan apakah suatu berkas, dapat berupa berkas plain maupun gambar telah dimodifikasi oleh proses steganografi atau belum. Prinsip yang digunakan adalah dengan menentukan apakah *statistical properties* telah mengalami deviasi (berubah dari normal) atau tidak. Beberapa tes yang dilakukan Sangay bergantung kepada jenis format data yang akan dites, ada pula yang hanya mengukur *entropy redundant bits* yang ada.

Metode yang paling sederhana adalah sebuah metode yang dikembangkan oleh Ueli Maurer di dalam jurnalnya yang berjudul ”Universal Statistical Test for Random Bit Generetor”. Selanjutnya metode ini akan dinamakan tes Maurer. Prinsip ini akan berlaku jika berkas yang berisikan *message bits* memiliki entropy yang lebih tinggi daripada berkas aslinya.

Tes Maurer dilakukan dengan menghitung *statistical property* dari berkas yang akan dicek. *Statistical property* yang diukur hanya merupakan statistical property yang sederhana, misalnya jumlah bit 0 dan 1. Bila jumlah bit-bit tersebut menyimpang dari batas normal, maka kehadiran *message bits* akan segera terdeteksi. Kelemahan dari metode ini adalah metode ini tidask bisa diterapkan pada berkas *statistical property*-nya adalah property yang makroskopik, atau yang sifatnya lebih rumit daripada sekedar jumlah bit 0 dan 1. Contoh statistical property seperti ini adalah DCT pada berkas dengan format JPEG.

Metode lain diterapkan oleh Westfeld dan Pfitzman. Mereka menemukan bahwa proses *embeding of encrypted data* dapat mengakibatkan histogram frekuensi warna yang dimiliki oleh berkas gambar tersebut berubah.

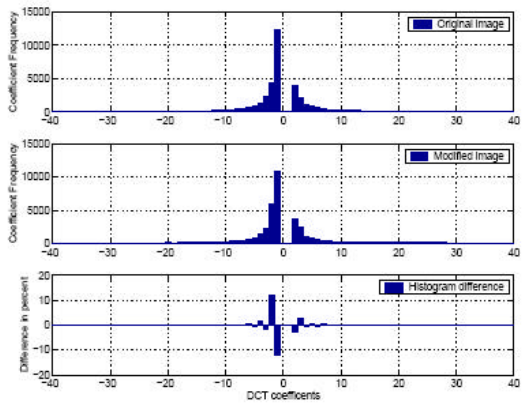
Untuk selanjutnya, makalah ini akan membahas pendekatan yang mereka lakukan dan bagaimana pendekatan itu berlaku untuk berkas dengan format JPEG. Proses *embedding* yang dikenakan pada stego medium dapat mengubah *least significant bits* warna-warna yang ada pada suatu berkas gambar. Setiap warna mengacu kepada warna yang telah berindeks yang terdapat pada tabel warna.

Jika n_i dan n_i^* adalah frekuensi warna pada tabel indeks sebelum dan sesudah dilakukan embedding, maka relasi yang terbentuk adalah:

$$|n_{2i} - n_{2i+1}| \geq |n^*_{2i} - n^*_{2i+1}|.$$

Hubungan ini menunjukkan bahwa besar perbedaan frekuensi antara dua warna yang berurutan akan berkurang dengan adanya proses *embedding*. Pada *message bits* yang sudah dienkripsikan, bit-bit 0 dan 1 didistribusikan secara merata. Bila ditulis $n_{2i} > n_{2i+1}$, maka hal ini berarti bit-bit pada hidden message mengubah n_{2i} menjadi n_{2i+1} lebih sering dari yang lain. Dengan kata lain terjadi deviasi.

Kasus yang sama dapat tejadi pada berkas dengan format JPEG, hanya saja *property* yang diukur bukanlah frekuensi warna, namun frekuensi dari koefisien DCT. Gambar berikut menunjukkan histogram sebelum dan sesudah dilakukan proses *embedding* pada berkas dengan format JPEG. Perbedaan pada histogram ditunjukkan pada *subgraph* pada bawah gambar.



Gambar 2 Perbaikan statistikal yang dilakukan secara sederhana menyebebkan frekuensi dari DCT yang bertetanggaan menjadi sama. Hal ini membuktikan berkas gambar sudah dimodifikasi

Dari hasil penelitian diperoleh adanya reduksi pada perbedaan frekuensi antara koefisien -1 dan -2. Bertetanggaan maksudnya kedua koefisien tersebut memiliki perbedaan paling kecil dalam hal bit-bit yang digunakan sebagai acuan. Dari histogram diatas juga diperoleh adanya pengurangan frekuensi yang dapat diamati antara koefisien dua dan tiga.

Westfeld dan Pfitzmann menggunaan tes X^2 untuk menentukan apakah distribusi frekuensi warna pada gambar sesuai dengan distribusi yang oleh distorsi dari message bits. Pada makalah ini, tes X^2 akan digunakan untuk berkas dengan format TPEG.

Karena tes ini hanya dilakukan pada *stego medium*, maka distribusi yang diharapkan y_i^* untuk tes X^2 ini harus dihitung secara langsung dari berkas gambar. Sedangkan untuk *stegotext*-nya, diasumsikan bahwa frekuensi DCT yang bertetanggaan akan bernilai sama. Bila frekuensi DCT pada histogram disebut n_i, maka nilai *arithmatic mean* (rata-rata aritmatika) akan bernilai:

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

Persamaan di atas menentukan jumlah distribusi yang diharapkan pada stegotext. Hasil yang diperoleh ini dibandingkan dengan distribusi yang diperoleh sebelumnya, yaitu:

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

Nilai X^2 yang diperoleh dengan menghitung perbedaan distribusi antara yang diperoleh sebelumnya dengan hasil yang diperoleh lewat perhitungan adalah sama dengan:

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

dimana v adalah derajat kebebasan. Derajat kebebasan adalah nilai kategori yang berbeda dikurangi 1. Pada kasus ini, mungkin sekali dilakukan penjumlahan terhadap nilai-nilai distribusi yang diharapkan dengan nilai yang diperoleh sebelumnya untuk memastikan bahwa ada perhitungan yang cukup untuk setiap kategori. Metode ini mensyaratkan bahwa hasil penjumlahan tidak lebih dari empat. Bila dua kategori yang bertetanggaan dijumlahkan bersama-sama, maka derajat kebebasan harus dikurangi satu.

Nilai probabilitas p yang menunjukkan dua distribusi adalah sama diberikan dengan persamaan seperti berikut:

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

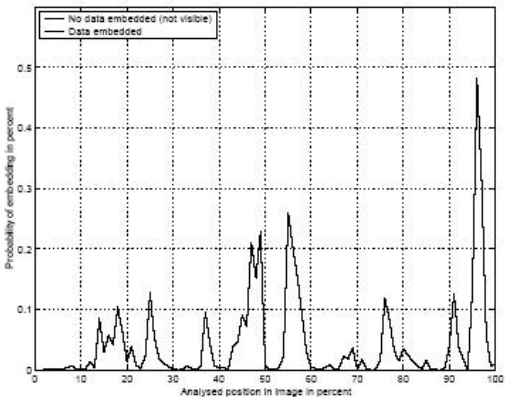
dimana ? disebut fungsi EulerGamma

Probabilitas adanya proses embedding pada suatu berkas, ditentukan dengan cara menghitung p dari koefisien DCT pada berkas gambar yang menjadi acuan.

Karena tes yang dilakukan ini membutuhkan ukuran berkas gambar yang lebih besar dari aslinya dan selalu dimulai pada awal gambar, maka tes ini hanya mampu mendeteksi perubahan jika distorsi yang terjadi, terdapat di awal berkas gambar dan terus berlanjut sampat akhir berkas gambar. Bila terdapat daerah di tengah-tengah pada berkas gambar yang tidak mencerminkan adanya distorsi, maka hal ini justru dapat mengakibatkan hasil yang negatif. Hasil yang negatif ini diperoleh walaupun pada area lain di dalam berkas gambar terdapat banyak distorsi. Bila hal ini terjadi, maka tes tidak akan dapat diterapkan untuk contoh kasus yang dipaparkan pada makalah ini.

5.1 Deteksi

Dari pengembangan yang dilakukan terhadap tes X^2, diperoleh kesimpulan bahwa masih mungkin untuk mengecek adanya distorsi secara parsial pada berkas gambar. Pada contoh kasus ini, diasumsikan adanya dua buah distribusiyang sama menghasilkan nilai X^2 yang sama pada setiap area diatribusi. Tes X^2 yang terdahulu mensyaratkan adanya penambahan besar berkas gambar dan penerapan metode pada posisi yang konstan, yaitu diawal, namun pada tes X^2 yang sudah dikembangkan ini dapat digunakan ukuran berkas yang sama dan tetapi posisi dari contoh yang digunakan diubah-ubah antara area sisa yang belum dianalisis. Dengan menggungkan tes X^2 yang telah dikembangkan ini, maka akan dapat dilakukan deteksi terhadap proses embedding. Hal ini ditunjukkan oleh gambar sebagai berikut:



Gambar 3 Metode tes X^2 yang telah dikembangkan dimana contoh yang digunakan menggunakan berkas gambar dengan nilai DCT 3,2% berhasilmendeteksi adanya informasi rahasia pada suatu berkas gambar, namun tes ini tidak menunjukkan reaksi terhadap gambar yang tidak termodifikasi.

Pada kasus ini digunakan berkas gambar dengan DCT bernilai 3,2%. Tes dilakukan pada awal berkas gambar dan posisi selanjutnya ditingkatkan 1% untuk setiap tes X^2 yang dilakukan. Tes ini tidak akan bereaksi terhadap berkas gambar yang tidak termodofikasi, tetapi dapat mendeteksi adanya modifikasi pada sebagian gambar yang terdapat pada berkas.

Untuk menentukan ukuran berkas yang tepat, maka perlu dipilih distribusi yang diharapkan dengan menggunakan tes X^2 yang telah dikembangkan dan dapat menghasilkan hasil yang negatif. Bila pada contoh sebelumnya dilakukan perhitungan rata-rata aritmatika antara sebuah koefisien dan koefisien yang bertetanggaan dengannya, maka kali ini dilakukan perhitungan rata-rata aritmatika antara dua koefisien yang tidak saling berhubungan.

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

Algoritma Binary Search diterapkan untuk mendapatkan nilai dimana tes X^2 tidak menunjukkan korelasi antaa distribusi yang diharapkan dan koefisien yang tidak berhubungan.

6. Perbaikan Terhadap Devíasi Statistik

Seperti yang disebutkan pada bab-bab sebelumnya, perubahan yang terjadi pada setiap *redundant bits* akan mengakibatkan adanya deviasi *statistical property* yang di-embed oleh *message bits*. Deviasi ini dapat dijadikan oleh musuh sebagai acuan untuk mendeteksi steganografi. Untuk mengatasi kecurigaan yang mungkin terjadi sebagai akibat adanya deviasi, maka sebelum *stegotext* dikirimkan, perlu dilakukan perbaikan terhadap deviasi yang ada. Perbaikan ini dilakukan dengan memanfaatkan bit-bit *redundant bits* yang tidak di-embed oleh *message bits*.

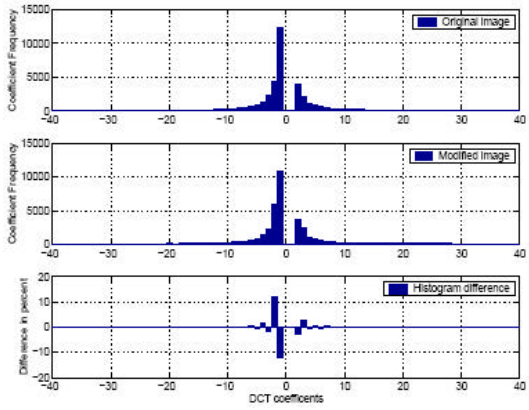
Pada kenyataannya proses seleksi hanya mungkin mengubah kurang dari setengah *redundant bit*. Bit-bit selebihnya tetap sama.

Bila pengirim pesan mengetahui *statistical test* apakah yang mungkin digunakan musuh untuk melacak perubahan terhadap gambar yang sudah dimodifikasi, maka kita dapat memperbaiki *stegotext* dengan menggunakan *redundant bits* yang tersisa untuk memb enarkan setiap deviasi statistik yang terjadi akibat proses *embedding*.

Cara yang paling sederhana yang dapat digunakan adalah dengan menjaga *entropy* dari *statistical property* tetap bernilai 1. Entopy adalah perbandingan perubahan yang terjadi antara *stegotext* terhadap *covertext*. Cara ini dilakukan dengan menggunakan Maurer Test.

Pada intinya, saat sebuah bit berubah dari 0 ke 1, maka pembuat pesan harus me lakukan perubahan bit yang berdekatan dari 0 menjadi 1. Walaupun pendekatan ini membantu mencegah peningkatan *entropy* dari *redundant bits*, cara ini tidak dapat digunakan untuk berkas yang *statistical property*-nya tergantung kepada *macroskopic properties*. *Macroscopic property* adalah *statistical property* yang sifatnya tidak lebih rumit daripada hanya perhitungan jumlah-jumlah bit 0 atau 1. Contoh *macroscopic property* adalah DCT pada berkas JPEG.

Karena *statistic property* pada berkas dengan format JPEG bersifat *macroscopic*, maka cara diatas justru menimbulkan ketidakcocokan. Penerapan metode perbaikan pada berkas JPEG dengan menggunakan cara sederhana di atas memberikan hasil yang dapat digambarkan sebagai berikut:



Gambar 4 Hasil steganografi pada berkas JPEG setelah dilakukan perbaikan terhadap deviasi terhadap jumlah bit 0 dan 1.

Grafik kedua pada gambar diatas adalah koefisien frekuensi yang muncul pada *stegotext*. Dapat diperhatikan bahwa terjadi deviasi antara

cover medium dengan *stegotext*-nya. Deviasi ini tidak dapat diimbangi dengan deviasi pada DCT yang berdekatan dengannya. Padahal, sudah dilakukan perbaikan terhadap deviasi yang muncul akibat perbedaan jumlah bit 0 dan 1. Hal ini membuktikan bahwa diatas tidak dapat diterapkan pada berkas dengan tipa JPEG.

Jika hendak dilakukan pengurangan distorsi pada histogram DCT, beberapa perbaikan tambahan perlu untuk menjada distribusi koefisien DCT tetap sama. Sebagai contoh, misalnya proses embedding sebuah *message bits* mengubah koefisien yang ke-j, dinamakan DCT(j). Sebagai contoh, modifikasi yang dilakukan terhadap DCT(j) = 2i, adalah dengan cara mengubahnya menjadi di+1.

Untuk mengkoreksi perubahan ini, pertama sekali pengirim pesan harus mencari bit-bit yang memiliki koefisien yang berdekatan dengannya, yaitu DCT(k) = 2i+1. Selanjutnya bit-bit tersebut diubah senilai dengan koefisien 2i. Jika kita mengkoreksi setiap perubahan pada koefisien DCT, maka histogram yang terbentuk akan identik dengan image awalnya.

Pada dasarnya, proses perbaikan dengan mengganti nilai bit-bit ini dapat menjaga kekonstanan dari jumlah frekuensi. Dengan demikian, tidak ada metode statistik yang menggunakan perhitungan frekuensi dapat mendeteksi adanya perbedaan antara *covertext* dan *stegotext*.

Penelitian terhadap konklusi di atas daapat dijelaskan sebagai berikut. Misalnya f adalah frekuensi yang dihitung pada histogram (gambar 3) dan f' adalah frekuensi bit-bit yang *statistical property*-nya bertetanggaan dengannya. Misalkan f = f'. Dan a adalah perbandingan antara bit-bit pada *redundant bits* yang digunakan untuk menyimpan *message bits* dengan jumlah *redundat bits* secara keseluruhan. Setelah dilakukan proses *embedding*, maka perubahan yang terjadi dapat dituliskan sebagai berikut:

$$f^* = f - \frac{\alpha}{2}(f - \bar{f}),$$
$$\bar{f}^* = \bar{f} + \frac{\alpha}{2}(f - \bar{f}).$$

Persamaan 6.1

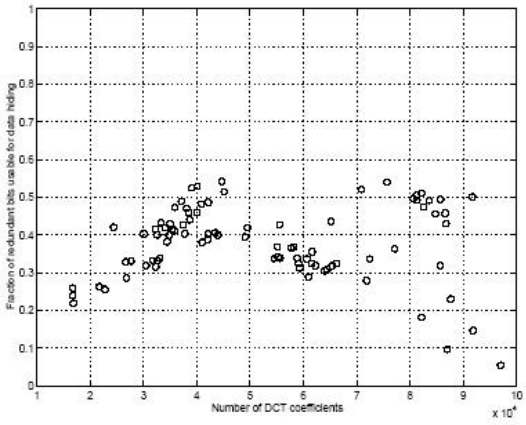
Persamaa ini dapat diturunkan untuk memperoleh perkiraan rata – rata a *redundant bits* yang dapat digunakan utnuk menyimpan *message bits* sebanyak f dan masih menyimpan *message bits* yang tidak diubah sebanyak f'. Pernyataan ini dapat ditulis dalam persamaan berikut:

$$\alpha \leq \frac{2\bar{f}}{f + \bar{f}}.$$

Persamaan 6.2

Bila pengirim pesan mendapatkan sebuah *message bits* yang hendak disimpan ke dalam sebuah file, maka pengirim pesan tersebut dapat memilih *covertext* yang digunakan untuk menyimpan *message bits* adalah *covertext* yang memenuhi persamaan diatas.

Pada berkas dengan format bertipe JPEG, jumlah perkiraan redundant bits yang dapat digunakan untuk menyimpan message bits tidak meningkat secara linear walaupun koefisien DCT telah meningkat secara linear. Hal ini digambarkan pada gambar berikut:



Gambar 5 Jumlah bit pada *redundant bits* yang digunakan untuk menyimpan data tidak meningkat secara linear, walaupun koefisien DCT meningkat.

Keseluruhan proses perbaikan terhadap deviasi yang terjadi harus memenuhi standar sebagai berikut:

1. Untuk setiap bagian dari gambar, distribusi koefisien DCT harus sama dengan koefisien DCT pada berkas yang belum dimodifikasi.

2. Jumlah perbaikan yang diperlukan untuk menjada kesamaan *statistical property* harus kecil.

Beberapa *statistical property* koefisien DCT bisa saja tidak diketahui oleh pihak pengirim pesan, jadi pengirim harus mencegah terjadinya distorsi tambahan yang mungkin terjadi setelah perbaikan. Distorsi seperti ini dapat terjadi sebagai hasil dari perbaikan terhadap property yang tidak diketahui. Jika pengirim dapat selalu menjaga proses modifikasi tambahan tetap kecil, maka pengirim dapat mencegah atau mengurangi kemungkinan distorsi *statistical property* tambahan.

Seperti yang dijelaskan pada bab sebelumnya, bahwa pada steganografi dikenal X^2 test. Agar sebuah berkas terbebas dari pendeteksian dengan menggunakan metode test ini, maka semua area dari gambar harus bebas dari adanya distorsi. Test ini tidak akan menemukan adanya proses embedding jika setiap area dari gambar yang sudah dimodifikasi memiliki distribusi yang sama dengan aslinya.

Pada bagian di atas telah dijelaskan bahwa syarat yang diperlukan untuk melakukan perbaikan terhadap deviasi yang muncul adalah untuk semua gambar dengan format JPEG, distribusi koefisien DCT harus sama dengan *covertext*-nya. Syarat selanjutnya adalah harus dilakukan perubahan sekecil mungkin terhadap *covertext* agar tidak terjadi deviasi tambahan yang muncul kembali.

Algoritma pertama yang akan diperkenalkan berikut memenuhi kedua persyaratan yang dimaksud. Algoritma ini dijalankan setelah proses embedding selesai dilakukan..
Langkah-langkah yang dilakukan pada algoritma ini adalah :

1. Perhitungan frekuensi DCT histogram dari gambar yang asli dan menyimpannya di dalam variabel N.
2. Penentukan frekuensi kesalahan. Frekuensi kesalahan mengindikasikan berapa banyak kesalahan yang terdapat pada histogram yang dapat kita toleransi untuk koefisien DCT yang spesifik. Perhitungan ini dilakukan dengan menggunakan persamaan 6.2. Nilai ini diperoleh dengan cara menghitung frekuensi yang diteliti dengan faktor skala a. Saat nilai kesalahan melebihi frekuensi kesalahan yang diijinkan, maka disimpulkan perlu dilakukan modifikasi berkas gambar untuk mempertahankan statistik koefisien tersebut.
3. Menentukan DCT yang bertetanggaan, yaitu indeks dari koefisien yang bertetanggaan dari yang dimodifikasi.
4. Menentukan apakah ada *pending error* koefisien yang bertetanggaan yang harus diperbaiki. Pada kasus ini perbaikan yang dilakukan pada DCT dapat digantikan dengan *pending correction* dari koefisien yang bertetanggan dengannya.
5. Bila tidak ada *pending error* yang berhasil ditemukan, maka dapat dilakukan pengecekan nilai *error* dari koefisien, N_{error} [DCT(i)], apakah nilai ini bisa ditingkatkan tanpa melebihi frekuensi kesalahan yang diijinkan. Jika *increment* selanjutnya juga mungkin untuk dilakukan, kita melanjutkan modifikasi selanjutnya. Dengan kata lain, kita harus mengkoreksi modifikasi yang sekarang dilakukan pada gambar.

Algoritma di atas dapat dituliskan sebagai berikut:

```
N ← DCTFreqTable(original);
k ← number of coefficients in image;
α ← 0.03 * 5000/k;
for i ← DCT_min to DCT_max do
    N_error[i] ← 0;
    N*[i] ← αN[i];
endfor
for i ← 1 to k do
    if DCT(i) unmodified then
        continue in loop;
    endif
    AdjDCT ← DCT(i) ⊕ 1;
    if N_error[AdjDCT] then
        decrement N_error[AdjDCT];
        continue in loop;
    endif
    if N_error[DCT(i)] < N*[DCT(i)] then
        increment N_error[DCT(i)];
        continue in loop;
    endif
    if exchDCT(i, DCT(i)) fails then
        increment N_error[DCT(i)];
        continue in loop;
    endif
endfor
for i ← DCT_min to DCT_max do
    while N_error[i] ≠ 0 do
        decrement N_error[i];
        exchDCT(k, i);
    endw
endfor
```

ExchDCT algoritma adalah algoritma yang berfungsi untuk melakukan pengujian apakah perubahan terhadap koefisien error, N_{error} [DCT(i)], telah melebihi frekuensi kesalahan yang diijinkan.

Jika cara ini juga gagal untuk dilakukan, atau dengan kata lain, perbaikan yang dilakukan pada gambar melebihi batas frekuensi kesalahan yang diijinkan maka proses terus dilanjutkan walaupun hal itu berarti meleihi koefisien error yang diijinkan untuk berkas tersebut.

Setelah semua modifikasi selesai diperiksa, kita perlu memperbaiki *error* yang ada akibat modifikasi kesalahan melebihi batas frekuensi yang diijinkan. Perlu diperhatikan, bahwa tidak semua perbaikan mungkin untuk dilakukan. Namun, jika kita sanggup untuk memperbaiki hampir semua kesalahan (*error*), maka perubahan yang ada pada histogram tidak akan terdeteksi.

Algoritma ExchDCT merupakan algoritma yang sangat sederhana untuk dilakukan. Bila diberikan koefisien DCT dan posisi *i* di dalam gambar, maka dengan menggunakan algoritma ini, untuk menemukan posisi mana yang memiliki koefisien yang sama dengan koefisien pada posisi i. Selanjutnya bit-bit dengan jumlah koefisien yang sama ini diubah menjadi koefisien yang bertetanggaan dengannya.

Langkah ini diawali dengan bit yang koefisiennya berdekatan dengan dengan bit yang koefisiennya menyebabkan algoritma ini dilakukan. Proses akan berhenti hingga tiba di awal gambar.

Koefisien yang telah digunakan untuk menyimpan *message bits* atau yang digunakan untuk memperbaiki koefisien bit-bit yang terdahulu tidak akan diperiksa oleh algoritma exchDCT(). Algoritma ini sangat menentukan kesuksesan atau kegagalan perbaikan yang dilakukan.

```
Function: exchDCT()
Data    : i, DCT
AdjDCT ← DCT ⊕ 1;
for j ← i − 1 to 1 do
    if DCT(j) = DCT and
    DCT(j) does not hold data and
    DCT(j) has not been used for corrections
    then
        DCT(j) ← AdjDCT;
        return success
    endif
endfor
return failure
```

7. Analisis

Untuk menghitung seberapa efektif algoritma perbaikan dapat berfungsi pada sebuah berkas dengan format JPEG, maka perlu dilakukan proses embedding message bits data pada berkas JPEG.

Percobaan ini dilakukan dengan melakukan penyimpanan message bits pada 54 buah berkas gambar JPEG yang berukuran 640 x 480 pixel. Gambar ini merupakan gambar digital. Sebelum

melakukan proses embedding, maka perlu dilakukan langkah konversi. Pada kasus ini langkah konversi dilakukan dengan mengkonversi berkas gambar menggunakan faktor kualitas sebesar 75.

Pada kasus ini, nilai rata-rata koefisien DCT yang digunakan adalah 46.000, berkisar antara 30.000 hingga 97.000. Setiap DCT yang ada memungkinkan adanya *redundant bits*. Tanpa dilakukan proses *embedding* pada *redundant bits*, ditemui korelasi pada *redundant bits*-nya. Secara rata-rata, deviasi yang ada berkisar antara 63.8% dengan rentang ±3.4%.

Message bits yang disimpan adalah bab pertama dari sebuah buku Lewis Carrol's dengan judul "The Hunting of the Snark". Setelah dikompresi, *message bits* yang akan disimpan mempunyai ukuran 14.700 bit.

Percobaan dilakukan sebanyak dua kali. Pertama sekali proses embedding dilakukan tanpa diikuti proses perbaikan kesalahan. Percobaan kedua dilakukan dengan melakukan embedding yang menggunakan kedua metode yang sudah diterapkan sebelumnya, kemudian dilanjutkan dengan melakukan perbaikan terhadap bit-bit yang menimbulkan kecurigaan bagia pihak musuh.

Covertext yang dihasilkan oleh kedua percobaan ini kemudian diuji dengan menggunakan Maurir Test. Hasil pengujian disajikan pada tabel berikut:

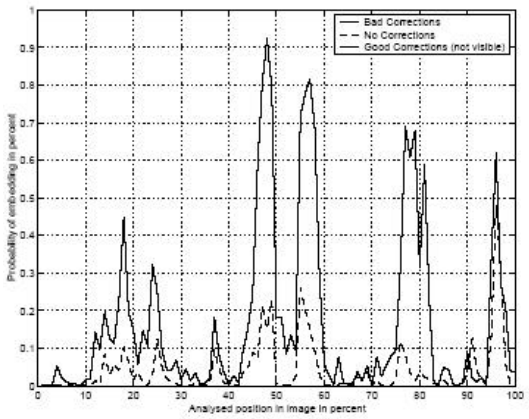
Berkas	Deviasi	Maurer Test
Asli	63-41% ± 3.50%	6.732±0.233
Tanpa perbaikan	59.10% ± 3.19%	6.976±0.168
Dengan perbaikan	62.91± 3.36%	6.775±0.231

Tabel 2 Perbandingan antara berkas asli dengan berkas yang telah dimodifikasi tanpa perbaikan statistik dan pada berkas yang dimodifikasi dengan menggunakan perbaikan statistik

Dengan Maurer Test yang menggunakan block berukuran bit, hasil yang diperoleh pada berkas secara acak adalah 7.148. Pada tabel di atas, diperoleh entropy yang cukup besar pada *stegotext* yang tidak melewati proses perbaikan perbaikan. Sedangkan pada stegotext yang telah dikenakan proses perbaikan kesalahan, ditemukan hasil Maurer Test yang dihasilkan memiliki nilai yang hampir sama dengan berkas yang asli. Hal ini membuktikan bahwa metode *embedding* dan juga perbaikan kesalahan yang diterapkan untuk meningkatkan keamanan pada stegnalisis dapat melewati serangan terhadap steganalisis dengan menggunakan metode *statistical* Maurer Test.

Namun, seperti yang pada bab-bab sebelumnya, Maurer Test hanya berupa metode statistical yang sangat sederhana, karena hanya memperhitungkan statistical property yang sederhana. Selanjutnya, berkas *stegotext* akan diujikan dengan menggunakan metode yang lebih rumit, yaitu X^2 test.

Test X^2 diujikan pada dua jenis *stegotext*, yaitu *stegotext* yang dienkripsi dengan menggunakan perbaikan kesalahan yang paling sederhana, (seperti yang dijelaskan pada bab 6) dan perbaikan kesalahan dengan memperbaiki kesalahan koefisien DCT. Hasil yang diperoleh direpresentasikan oleh gambar berikut:



Gambar 6 X^2 dapat mendeteksi *hidden message* yang melewati proses perbaikan yang tidak menggunakan *statistical property* yang tepat.

Statistik yang ditunjukkan diatas menunjukkan hasil yang sangat menarik. X^2 test hanya bisa menemukan kesalahan pada *stegotext* yang tidak melewati perbaikan kesalahan pada koefisien DCT. Sebaliknya tes ini tidak dapat menemukan perbedaan distribusi yang timbul pada *stegotext* yang telah melewati perbaikan pada koefisien DCT-nya dengan menggunakan algoritmat yang digunakan pada sub-bab 6.1. Dari penjelasan ini diperoleh kesimpulan bahwa

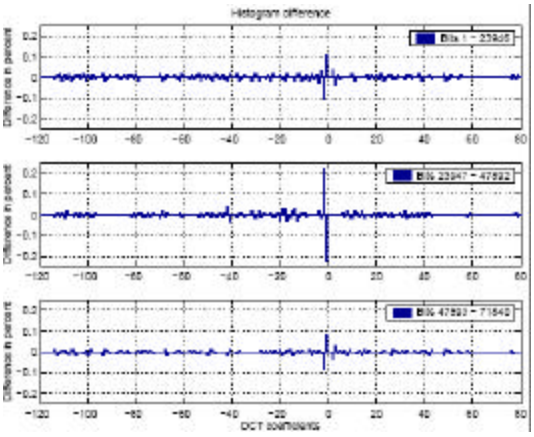
stegotext yang diperbaiki dengan menggunakan metode perbaikan, akan dapat terdeteksi dengan menggunakan X^2 test bila *statistical property* yang digunakan tidak merupakan *statistical property* yang tepat. Pada berkas JPEG contohnya, perbaikan yang dilakukan haruslah pada koefisien DCT-nya bukan pada property lain seperti jumlah bit 0 dan 1.

Pada spesifikasi *cover medium* dan *message bits* yang diberikan sebagai contoh, ditemukan sekitar 2967 ± perubahan yang harus dilakukan pada *redundant bits*. Perubahan yang dilakukan ini berkisar antara 20% dari ukurasn *message bits*. Jumlah bit-bit yang tidak dapat dikoreksi adalah sekitar 186 ± 400.

Walaupun hasil yang diperoleh sudah cukup memuaskan, namun masih mungkin dilakukan langkah-langkah untuk menurunkan jumlah perbaikan yang harus dilakukan pada. Cara yang dilakukan adalah dengan mengabaikan seleksi terhadap koefisien yang mempunyai frekuensi yang lebih besar daripada frekuensi yang bertetanggan dengannya. Karena dari hasil penelitian, koefiseien yang frekuensinya lebih besar dibandingkan frekuensi koefisien tetangganya akan lebih mudah terdeteksi. Untuk menerapkan proses seleksi seperti ini, maka pada proses identifikasi bit dilakukan penandaan terhadap bit-bit dimana frekuensi pada koefisien tersebut lebih besar daripada frekuensi koefisien yang bersesuaian dengannya.

Bila dengan metode seleksi yang paling buruk menyebabkan 3365 ± 442 perbaikan, maka dengan menggunakan seleksi yang terbaik, diperoleh 3054 ± 430 perubahan. Bila dibandingkan pada setiap gambar menggunakan teknik ini, maka akan dihasilkan pengurangan sebanyak 311 ± 68 rata-rata perubahan.

Perbaikan yang dilakukan juga harus memenuhi syarat bahwa tidak ada area di dalam *redundant bits* yang dapat dilacak apakah ada distorsi pada bagian tersebut. Bila ada perbedaan frekuensi ditemui, maka perubahan tersebut harus ditolak, sehingga X^2 test tidak dapat menemukan adanya *message bits*. Pada berkas gambar dengan format JPEG, perbedaan yang muncul pada stegotext digambarkan oleh histogram berikut:



Gambar 7 Pemeriksaan terhadap perbedaan pada histogram DCT pada bagian gambr menunjukkan tidak adanya deviasi dari gambar yang belum diubah.

8. Kesimpulan

Walaupun steganografi sering dimanfaatkan untuk menyembunyikan pesan rahasia karena sifatnya yang tidak mencurigakan pihak ketiga, namun metode *statistical analysis* mampu membuktikan keberadaan pesan rahasia pada sebuah *stegotext*.

Untuk melakukan proses *embedding* message bits maka perlu dilakukan konversi stego medium ke dalam bentuk keluaran yang diinginkan. Pada contoh kasus di makalah ini, konversi dilakukan dengan mengkompressi berkas-berkas gambar dengan format JPEG. Selanjutnya dilakukan identifikasi *redundant bits* dan dilanjutkan dengan proses seleksi bit-bit pada *redundant bit* yang dapat digunakan untuk menyimpan *message bits*.

Pada makalah ini terdapat dua metode untuk meningkatkan kerahasiaan pada proses seleksi. Metode yang pertama adalah dengan menggunakn pseudo random number generator (PRNG) untuk mengurangi modifikasi yang harus dilakukan pada *cover medium.* Metode yang kedua adalah dengan menggunakan perbaikan kesalahan sehingga meningkatkan fleksibilitas di dalam pemilihan bit-bit yang akan dimodifikasi. Perbaikan kesalahan ini dilakukan untuk menghindari pemilihan bit-bit yang telah di-*lock* atau bit-bit yang mempunyai kecenderungan mudah terdeteksi oleh pihak luar.

Walaupun metode tes X^2 tidak dapat digunakan untuk mendeteksi adanya modifikasi setelah penggunaan proses *embedding* yang telah diperbaharui yang dijelaskan pada makalah ini, namun tes ini dapat dikembangkan sehingga mampu mendeteksi perubahan yang terjadi pada *cover medium.*

Untuk menghitung tes statistik berdasarkan frekuensi yang dihitung seperti yang dilakukan pada metode tes X^2 yang telah dikembangkan, dapat dilakukan sebuah metode baru untuk memperbaikai deviasi statistik yang muncul lewat proses *embedding* dan memperbaiki transformsi pada format JPEG. Sebagai hasilnya, tidak ada suatu *statistical analysis* yang sudah dijelaskan pada makalah ini mampu mendeteksi adanya *message bits* yang tersimpan pada *stegotext* tertentu.

Daftar Pustaka

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Schneier, Bruce. *(1996). Applied Cryptography 2nd. John Wiley & Sons.*

[3] Defending Against Statistical Analysys (2004), *http://citeseer.ist.psu.edu*, Tanggal Akses 17 September 2006 pukul 20.00 WIB