ANALISIS *STEGANOGRAPHY* TERHADAP KEAMANAN DATA

Kundang Karsono Juman
Fasilkom – Universitas INDONUSA Esa Unggul Jakarta
Jl. Arjuna Utara Tol Tomang Kebon Jeruk, Jakarta 11510
Kundang.karsono@indonusa.ac.id

Abstract

The computer security system includes security data this time able to issue demand. Steganography is the art of concealing the existence of information within seemingly innocuous carriers. Steganography can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information. At times these two technologies seem to converge while the objectives of the two differ. Cryptographic techniques "scramble" messages so if intercepted, the messages cannot be understood. Steganography, in an essence. The Implementation steganography in side data security with: Learning to the concept computer security system, to be understand steganography, and to knows in implementation steganography in the data security

Keywords: *The Computer Security, Data Security, Steganography Method Data Security and Media*

Pendahuluan

Masalah *security* atau keamanan jaringan merupakan masalah yang tidak dapat dihindarkan dalam suatu sistem jaringan. Masalah ini dapat menjadi krusial ditengah-tengah dominannya *Computer Based Information System*. Namun demikian, masalah keamanan ini kerap terabaikan oleh administrator jaringan dengan berbagai alasan misalnya seperti mengganggu performa sistem.

Hampir disemua aspek kehidupan manusia, komputer menjadi andalan. Dalam perusahaan-perusahaan, komputer diandalkan untuk mengerjakan transaksi-transaksi rutin, pemrosesan data, dan pengolahan data. Semua ini membuat perusahaan perlu membangun jaringan komputer yang dapat digunakan seluruh aspek perusahaan untuk meningkatkan produktifitas perusahaan. Ditambah lagi dengan lepas landasnya dunia dari era industri menuju era informasi.

Informasi kini menjadi sesuatu yang berpengaruh dalam kehidupan manusia. Untuk alasan ini kemudahan didalam melakukan *sharing* informasi mendorong munculnya jaringan komputer. Masalah tidak selesai sampai disana karena muncul masalah berikutnya dimana *sharing* informasi inilah yang menimbulkan celah bagi pihak-pihak yang tidak berkepentingan untuk mencuri informasi yang di*share* tersebut. Jika misalnya informasi rahasia perusahaan, seperti *prototype* produk terbaru perusahaan yang akan dirilis, bocor dan diketahui pesaing maka perusahaan akan mengalami kerugian yang tidak sedikit. Karena itulah muncul *issue-issue* mengenai keamanan jaringan.

Membangun keamanan jaringan komputer yang kuat merupakan keharusan untuk mencegah bocornya informasi-informasi perusahaan keluar. Pengamanan terhadap komunikasi data juga harus mendapat perhatian untuk mencegah kebocoran informasi. Salah satu cara yang dapat digunakan untuk mengatasi hal ini adalah *steganography* yaitu seni atau ilmu penyembunyian data. Metode ini dilakukan untuk menyembunyikan data dari pencuri data sehingga eksistensi data tidak diketahui. Namun demikian, bukan berarti dengan menyembunyikan data semua masalah menjadi selesai, keamanan jaringan dapat menjadi tembok pertama untuk mencegah pencurian data. Kedua hal ini bersifat komplementer untuk melindungi kerahasian informasi maupun data. Oleh karena itu, penulisan ini akan menganalisis *steganography* terhadap keamanan data.

Dengan didasari latar belakang masalah tersebut dapat dirumuskan beberapa masalah sebagai berikut:

a. Sejauh mana peranan *steganography* pada *security* jaringan.
b. Kapabilitas *steganography* pada keamanan data.
c. Teknik *steganography* mana yang paling baik untuk diimplementasikan.

Tujuan dalam penulisan ini adalah mengidentifikasikan peranan dan kemampuan *steganography* dalam pengimplementasiannya pada keamanan data sehingga kerahasiaan dan integritas data terjaga.

Adapun manfaat dari penulisan ini adalah membuka wawasan baru terhadap metode pengamanan data yang dapat dimplementasikan bersama dengan metode lain untuk memperkuat aspek *confidentially* dan *integrity* data dalam keamanan sistem jaringan informasi.

Pembahasan

Teknik Steganography

Umumnya teknik *steganography* dikelompokkan berdasarkan media yang akan digunakan sebagai *cover medium*. Secara garis besar dua konsep yang paling sering dan populer untuk untuk dibahas berdasarkan media yang digunakan adalah *image steganography* dan *audio steganography*.

Image *Steganography*

Secara umum teknik *image steganography* diklasifikasikan menjadi dua kelompok (Silman, 2001), yakni:

- *Image domain*
- *Transform domain*

Image domain menyembunyikan data langsung pada intensitas *pixel*. Sementara itu *transform domain* mentransformasikan gambar terlebih dahulu sebelum data disembunyikan.

Selain itu terdapat teknik-teknik yang mengombinasikan kedua teknik diatas.

Teknik Image Domain

Seperti yang telah disebutkan pada sub bab sebelumnya, teknik *image domain* menyembunyikan data langsung pada intensitas *pixel cover mediumnya*. Teknik ini melingkupi metode *bitwise* yang mengimplementasikan *bit insertion*. Teknik ini cocok diimplementasikan pada media yang menggunakan *lossless* dan teknik ini bergantung kepada format file yang digunakan sebagai *cover medium-nya*. Teknik *image domain* yang paling umum adalah *least significant bit modification* dan *masking* dan *filtering.*

Least Significant Bit Modification

Least significant bit modification (LSB *modification*) adalah metode di dalam menyembunyikan data dimana metode ini pada prinsipnya bekerja dengan mengganti LSB dari setiap *byte* pada masing-masing *pixel image* yang akan digunakan sebagai *cover image* dengan *binary code* dari data-data yang akan disembunyikan (Bret Dunbar, 2002). Metode ini sangat sederhana namun cukup efisien untuk digunakan dalam menyembunyikan data. Selain digunakan untuk menyembunyikan data pada *image file* metode ini juga dapat digunakan untuk *audio steganography*. Metode ini dapat menggunakan *image 8-bit* maupun *image 24-bit*. Metode LSB *modification* ini memerlukan media yang besar untuk dapat menampung data yang besar pula. Dengan menggunakan *image 8-bit*, yang terdiri dari 256 warna, maka dari tiap satu *pixel* gambar yang digunakan sebagai *cover* dapat menyembunyikan satu *bit* data. Jika *image 24-bit* yang digunakan maka tiap satu *pixel* pada *cover image* dapat disembunyikan tiga *bit* data yang akan disembunyikan.

Cara kerja metode ini dapat dipaparkan melalui contoh seperti berikut ini:

misalnya terdapat *file image* A yang menggunakan *image 24-bit color* (3 *bytes* pada masing-masing *pixel*). Pada *file* A tersebut diambil *sample* berupa 4 *pixel* yang akan digunakan untuk menyembunyikan data. Keempat *pixel* tersebut bernilai desimal sebagai berikut:

Pixel I : 181 77 73
Pixel II : 202 133 54
Pixel III : 190 23 194
Pixel IV : 153 20 139

Adapun bentuk biner dari kedua belas *bytes* adalah sebagai berikut:

10110101 01001101 01001001
11001010 10000101 00110110
10111110 00010111 11000010
10011001 00010100 10001011

Kemudian sebuah data 12 *bits*, kita sebut saja data B, akan disembunyikan pada keempat *pixel* dari *file* A diatas. Nilai biner dari data B tersebut adalah sebagai berikut:

1 0 0 0 1 1 0 1 1 0 1 1

Dengan metode LSB *modification*, masing-masing LSB dari kedua belas *bytes file* A (dicetak berwarna biru) akan digantikan dengan dua belas *bits* data B yang akan disembunyikan secara berurutan dari tiga *bytes* pada *pixel* pertama sampai tiga *bytes* pada *pixel* terakhir. Hasil dari metode LSB *modification* ini akan menghasilkan dua belas *bytes* baru sebagai berikut:

10110101 01001100 01001000
11001010 10000101 00110111
10111110 00010111 11000011
10011000 00010101 10001011

Dari susunan *bits* diatas terlihat bahwa kedua belas *bits file* B telah disembunyikan pada empat *pixel* yang terdapat pada *file* A. Proses tersebut menyembunyikan data B dengan mengganti 50% LSB pada *file image* A, yakni

- *byte* kedua dan ketiga pada *pixel* pertama
- *byte* ketiga pada *pixel* kedua
- *byte* ketiga pada *pixel* ketiga
- *byte* pertama dan kedua pada *pixel* keempat

Bit yang dicetak berwarna merah adalah *bit file* A yang diganti dengan *bit* pada data B.
Nilai desimal masing-masing *byte* pada keempat *pixel image* A akan menjadi:

Pixel I : 181 76 72
Pixel II : 202 133 55
Pixel III : 190 23 195
Pixel IV : 152 21 139

Nilai yang dicetak merah merupakan nilai pada *pixel* yang mengalami perubahan karena metode LSB *modification*.

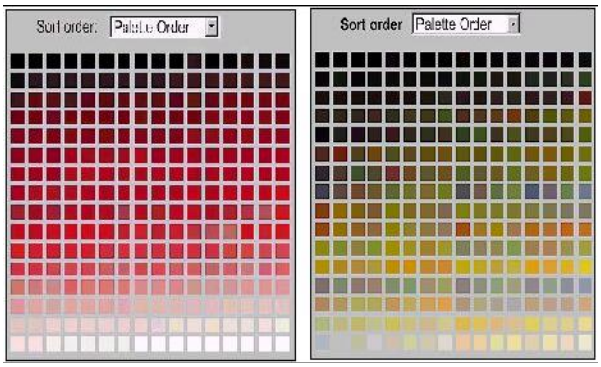
Dari contoh di atas dapat dilihat bahwa dari keempat *pixel* tersebut hanya ada hanya terdapat 6 *byte* yang mengalami perubahan. Hal tersebut berarti rata-rata hanya setengah dari jumlah keseluruhan *byte* pada *cover medium* yang akan diubah. Karena hanya ada 256 kemungkinan intensitas warna pada masing-masing *byte* pada *image 24-bit* maka modifikasi pada LSB hanya akan memberikan dampak yang sangat kecil pada intensitas warna dimana mata manusia tidak akan mampu membedakannya.

Dalam implementasinya LSB *modify-cation* tidak hanya begitu saja memodifikasi LSB tiap *byte* dengan *bit-bit* data secara linier. Hal ini tentu saja dilakukan agar obyek *steganography* tersebut tidak mudah dideteksi. Untuk mendukung ide tersebut maka dibutuhkan suatu *key* yang akan menjadi acuan *byte* mana yang akan dimodifikasi. Sehingga jika suatu obyek *steganography* diketahui akan sulit untuk menentukan *byte* mana yang LSB-nya dimodifikasi.

Biasanya metode ini menggunakan *file image* dengan 256 warna (*image 8-bit*). Hal tersebut dikarenakan jika menggunakan *image 24-bit*, ukuran *file* tersebut akan sangat besar sehingga dapat menimbulkan kecurigaan selain akan sulit melakukan transmisi data pada jaringan global seperti internet (Robert Krenn, 2004). Format *file* yang sering digunakan adalah format *file* BMP (*Bitmap*) dan GIF. Namun jika ingin menyembunyikan data menggunakan *file* dengan format BMP sebagai *cover medium* maka dibutuhkan *file* dengan ukuran yang sangat besar (Robert Krenn, 2004). Selain itu format tersebut sudah jarang digunakan dalam internet sehingga dapat menimbulkan kecurigaan mengenai keberadaan data tersembunyi didalamnya. Format *file* yang paling cocok digunakan pada teknik *least significant bit modification* ini adalah format *file* yang berbasis *pallete*. Format *file* berbasis *pallete* yang paling populer adalah GIF.

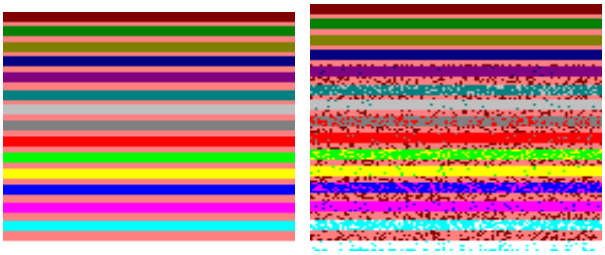
LSB Modification Format File Berbasis Pallete

Format *file* berbasis *pallete* yaitu format *file* yang gambarnya diindeks dimana setiap warna yang membentuk gambar tersebut disimpan pada *pallete*. Setiap *pixel* pada gambar merepresentasikan sebuah *byte* dan berisi data indeks pada *pallete*-nya. Warna-warna pada *pallete* tersebut biasanya diurutkan berdasarkan warna yang paling sering digunakan. Hal ini dilakukan untuk mengurangi waktu pada saat mengakses warna pada *pallete* (Neil F. Johnson, Sushil Jajodia, 1998). Format *file* ini, contohnya GIF (*Graphic Interchange Format*), hanya mempunyai batas warna sebanyak 256 warna. Format *file* GIF sering digunakan sebagai media pada teknik *least significant bit modification*. Namun dalam menggunakan format *file* ini harus berhati-hati ketika akan memilih gambar yang akan digunakan sebagai *cover medium* (Neil F. Johnson, Sushil Jajodia, 1998). Sebaiknya gambar yang akan digunakan sebagai *cover medium* tidak mempunyai perbedaan intensitas warna yang sangat kontras pada tiap *adjacent pallete*-nya (letak *pallete* yang bersisian langsung) karena jika perbedaan tersebut sangat kontras maka ketika dilakukan teknik LSB *modification* perbedaan tersebut makin tampak dan membuat gambar asli dan gambar hasil *steganography* terlihat jelas perbedaannya oleh mata manusia (Kevin Curran, Karen Bailey, 2003).



Sumber: Johnson, 1998

Gambar B.1.
Perbedaan Pallete yang Berwarna Tidak Kontras dengan Pallete Berwarna Kontras



Sumber: Johnson, 1998

Gambar B.2.
Hasil Teknik LSB Modification pada Cover Berwarna Kontras

Untuk mengeleminasi perbedaan yang begitu ekstrim, maka warna-warna pada *pallete* tersebut diurut berdasarkan intensitas warnanya. Hal ini dilakukan agar perbedaan warna pada *pallete* tidak begitu jelas. Namun hal tersebut belum cukup untuk mengeleminasi perbedaan yang muncul, sehingga muncul ide untuk menambah *pallete* sehingga *cover image* diubah dari *8-bit* menjadi *24-bit* (Kevin Curran, Karen Bailey, 2003). Dengan cara ini perbedaan yang timbul tidaklah terlalu ekstrim tetapi tetap tidak efektif karena gambar ini akan berukuran sangat besar sehingga mengundang kecurigaan keberadaan pesan didalamnya. Salah satu cara yang mungkin dilakukan adalah mengkonversi kembali *image 24-bit* tersebut menjadi *image 8-bit* ketika data telah berhasil disembunyikan. Namun ada solusi lain dari masalah tersebut yakni menggunakan *greyscale image*. Jika menggunakan *greyscale image* berarti terdapat 256 warna perubahan dari hitam sampai putih (Kevin Curran, Karen Bailey, 2003). Dengan *greyscale image*, perubahan warna yang dihasilkan karena teknik LSB *modification* sangat *gradual* sehingga semakin sulit untuk dibedakan antara gambar asli dengan gambar *steganography*. Selain hal tersebut di atas, dalam memilih gambar yang akan digunakan sebagai *cover medium* juga harus memperhatikan jumlah warna pada *pallete* yang telah digunakan karena dengan teknik LSB *modification* kemungkinan terciptanya warna baru pada *pallete* sangat besar Neil F. (Johnson and Sushil Jajodia, 1998). Contoh saja jika pada suatu *cover medium* yang digunakan telah mempunyai 200 warna maka pada saat dilakukan teknik LSB *modification* yang mengubah 50% LSB pada *byte cover medium*

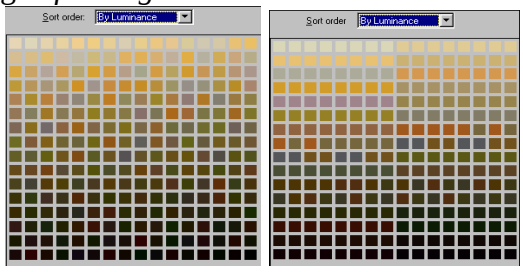
timbul kemungkinan warna baru yang dihasilkan dapat melebihi batas 256 warna. Untuk itu pada beberapa *tools steganography* seperti S-Tools yang mengimplementasikan teknik LSB *modification* digunakan teknik tambahan lain yakni *color reduction*.

Color Reduction

Teknik ini dilakukan dengan mereduksi jumlah warna menjadi tidak kurang dari 32 warna. Ke-32 warna ini diperluas sampai 8 masukan pada *pallete* dengan menambahkan warna *adjacent* pada *pallete* yang sangat serupa dengan warna aslinya. Teknik ini menghasilkan obyek *steganography* yang serupa dengan gambar aslinya yang secara visual sangat sulit dibedakan oleh mata manusia. Gambaran mereduksi *image* menjadi 32 *color* adalah dengan memvisualisasikan masing-masing LSB pada RGB yang dirangkai menjadi satu rangkaian 3 *bit*. Dalam proses menyembunyikan data, masing-masing rangkaian tiga *bit* tersebut dapat diganti dengan delapan kemungkinan nilai ($2^n = 8$; n=3) yakni:

- 000
- 001
- 010
- 011
- 100
- 101
- 110
- 111

Delapan kemungkinan nilai itu termasuk di dalamnya *bit-pattern file* asli. Dengan begitu satu warna dapat diperluas menjadi delapan warna sehingga didapat batas warna sejumlah 32 *color* (254 dibagi 8). Sejumlah *bit* yang akan disembunyikan mungkin saja sama sekali tidak mengubah delapan kemungkinan nilai tersebut sehingga detil warna *file* asli dapat dipertahankan dan membuat perbedaan antara *file* asli dengan *file stego* semakin kecil.



Sumber : Kevin Curran dan Karen Bail, 2003

Gambar B.3.
Pallete Cover Medium (kiri) dan Obyek Steganography (kanan) Setelah Implementasi Color Reduction

Masking dan Filtering

Teknik ini umum digunakan pada *image 24-bit* dan *greyscale image*. Data disembunyikan seperti teknik *digital watermarks* dan sering dipakai untuk *digital watermarks*. Latar belakang *digital watermarks* sangat berbeda dengan *steganography* dimana *digital watermarks* ditujukan untuk memberikan tanda (*sign*) untuk menunjukan kepemilikan (*ownership*) sehingga ketika terdapat dua buah gambar yang sama dimana salah satu gambar mengandung tanda yang Anda buat maka posisi Anda kuat untuk menuntut kepemilikan Anda.

Melakukan *masking* pada gambar pasti akan merubah keterangan (*luminosity*) dari area yang di-*masking*. Semakin kecil keterangan yang diganti, semakin kecil perubahan yang dapat dideteksi. Selain itu semakin tinggi *luminosity* maka semakin tinggi juga *visibility* gambar tersebut. Gambar dibawah menunjukan 15% pada area yang di-m*asking* jika dikurangi keterangannya akan menjadi *invinsible*. Hal ini berarti juga bahwa *luminosity* berbanding lurus dengan *visibility*.

Sumber: Johnson, 1998

Gambar B.4.
Gambar yang Di-*masking*

Obyek *steganography* yang di-*masking* lebih *robust* jika dibandingkan dengan obyek *steganography* yang menggunakan teknik LSB *modification* baik ketika dikompres, ketika gambar tersebut dimodifikasi, maupun ketika dilakukan *image processing* terhadap gambar tersebut. Alasan lain dipilihnya metode ini adalah karena dengan metode *masking* lebih sedikit mengalami penurunan kualitas ketika dilakukan kompresi JPEG dimana data disembunyikan pada area gambar yang signifikan.

Tools steganography seperti JPEG-JSteg memanfaatkan kompresi JPEG saat mencoba untuk mempertahankan *robustness* setinggi mungkin. Program tersebut meng-*input* data tersembunyi dan *lossless cover medium* dan mengeluarkan *output* berupa *stego image* yang mempunyai format JPEG (*Joint Photographic Experts Group*).

Teknik Transform Domain

Teknik ini dilakukan dengan mentransformasikan gambar terlebih dahulu sebelum data disembunyikan pada gambar tersebut. Teknik ini memanipulasi algoritma dan transformasi gambar. Teknik ini umum diterapkan pada formati *file* JPEG. Teknik ini menyembunyikan data pada area yang lebih luas pada *cover medium*. Dengan menggunakan teknik ini obyek *steganography* yang dihasilkan lebih *robust* dan memiliki daya tahan lebih dari metode kompresi, pemotongan gambar, dan *image processing* lainnya. Teknik ini menggunakan modifikasi dari *Discrete Cosine Transformation* (DCT) yang digunakan oleh algoritma kompresi *file* JPEG. Langkah pertama adalah mengkonversi warna RGB menjadi representasi YUV dimana komponen Y berkorespondensi dengan keterangan (*luminosity*) dan komponen U dan V berkorespondensi dengan warna. Lalu kompresi *file* JPEG ini memodifikasi data warna pada *file* untuk mereduksi ukuran *file* tersebut. Hal ini dilakukan karena mata manusia lebih sensitif terhadap perubahan keterangan suatu warna disbandingkan dengan perubahan warna itu sendiri. Langkah kedua adalah mentransformasikan gambar tersebut. Cara yang dilakukan adalah dengan menggunakan perhitungan transformasi matematika untuk mengkonversi *pixel* pada gambar sedemikian rupa sehingga memberikan efek menyebarkan lokasi nilai *pixel* pada seluruh bagian gambar. DCT mentransformasikan sinyal dari representasi gambar menjadi representasi frekuensi dengan mengelompokkan *pixel* kedalam 8x8 blok *pixel* dan mentransformasikan blok *pixel* tersebut kedalam masing-masing 64 koefisien DCT. Modifikasi yang dilakukan pada satu koefisien DCT akan berdampak pada kesemua 64 *pixel* dalam blok tersebut. Langkah berikutnya adalah fase penghitungan dari kompresi yang dilakukan. Pada tahap ini sistem *visual* manusia dimana mata manusia cukup sensitif terhadap perubahan kecil keterangan warna pada suatu area yang luas daripada perbedaan kekuatan frekuensi keterangan tinggi dieksploitasi. Frekuensi tinggi ini yang direduksi. JPEG menggunakan ini dengan membagi semua nilai pada blok dengan jumlah koefisien. Hasil tersebut dibulatkan menjadi nilai *integer* dan koefisien di-*encode* menggunakan *coding Huffman* untuk mereduksi ukuran *file*. Algoritma pemrogramannya akan terlihat sebagai berikut:

***Input** : message, cover image*
***Output**: steganographic image containing message*
***while** data left to embed do*
  get next DCT coefficient from cover image
  if** DCT 6= 0 and DCT 6= 1 **then
    get next LSB from message
    replace DCT LSB with message bit
  end if
  insert DCT into steganographic image
end while

Sebelumnya terlihat tidak mungkin untuk menggunakan format JPEG untuk menjadi obyek *steganography* karena JPEG menggunakan metode *lossy compression* dimana data-data yang bersifat *redundant* dibuang sehingga dikhawatirkan data-data yang akan disembunyikan pada data *redundant* tersebut hilang karena terbuang. Padahal algoritma kompresi JPEG terbagi menjadi dua tahap yaitu *lossy* dan *lossless*. DCT dan fase penghitungan yang telah dijabarkan diatas termasuk tahap *lossy* dan di lain pihak *encoding Huffman* termasuk dalam tahap *lossless*. *Steganography* dilakukan diantara kedua tahap ini dengan menggunakan prinsip LSB *modification* data disembunyikan pada LSB dari koefisien sebelum dilakukan *encoding Huffman*.

Teknik Image dan Transform Domain

Selain dua teknik yang telah dibahas diatas terdapat pula teknik yang mengombinasikan kedua teknik tersebut. Kombinasi

kedua teknik tersebut ada karena dengan menggunakan perpaduan kedua teknik itu kekurangan-kekurangan masing-masing teknik sebelumnya dapat diminimalisasi sehingga diharapkan dapat menghasilkan kinerja yang lebih maksimal. Beberapa teknik yang biasa digunakan ini adalah *patchwork* dan *spread spectrum*.

Patchwork

Teknik ini merupakan teknik statistikal yang menggunakan *redundant pattern encoding* untuk menyembunyikan data pada gambar. Algoritma tersebut menambahkan *redundancy* kedalam informasi yang akan disembunyikan dan kemudian menyebarkannya keseluruh gambar. Dengan menggunakan *pseudorandom generator*, dipilih dua area dari gambar (disebut *patch*) yakni *patch* A dan *patch* B (Bender W, 1996). Semua *pixel* pada *patch* A lebih terang sementara tiap *pixel* pada *patch* B lebih gelap. Dengan kata lain intensitas dari tiap *pixel* dari satu *patch* bertambah oleh sebuah nilai yang konstan (Marvel, 1999), dimana tiap *pixel* dari *patch* yang lain berkurang dengan nilai yang sama. Perubahan yang kontras pada subset *patch* tersebut meng-*encode* satu *bit* dan perubahan tersebut sangat kecil dan tidak terlihat dimana tidak merubah rata-rata *luminosity*-nya.

Kelemahan dari teknik *patchwork* ini adalah bahwa hanya satu *bit* yang ditambahkan pada *cover medium*. Namun terdapat solusi dari permasalah tersebut yakni dengan membagi gambar tersebut kedalam sub-gambar dan melakukan penambahan data pada tiap-tiap sub-gambar tersebut. Keuntungan memakai teknik tersebut adalah teknik ini mendistribusikan pesan yang akan disembunyikan pada keseluruhan gambar sehingga ketika satu *patch* dihancurkan karena kompresi misalnya, maka *patch* yang lain tetap bertahan. Akan tetapi teknik tersebut tergantung pada ukuran data yang akan disembunyikan. Pesan yang akan disembunyikan dapat ditambahkan pada *cover medium* secara berulang-ulang pada keseluruhan gambar hanya jika ukuran data yang akan disembunyikan tersebut cukup kecil. Tetapi jika data yang akan disembunyikan besar maka data tersebut hanya dapat ditambahkan pada *cover medium* sekali saja.

Teknik ini tidak bergantung pada format *file cover medium* yang akan digunakan dan data yang disembunyikan lebih *robust* terhadap baik *lossless* maupun *lossy compression*.

Spread Spectrum

Teknik *spread spectrum* menyembunyikan data dengan menyebarkan data keseluruh *cover medium*-nya. Teknik ini mengombinasikan komunikasi *spread spectrum, error control coding*, dan *image processing* untuk menyembunyikan data pada gambar. Komunikasi *spread spectrum* dapat didefinisikan sebagai proses menyebarkan *bandwidth* dari sinyal frekuensi *narrowband* menjadi *wideband*. Hal tersebut dapat dilakukan dengan menyetel bentuk gelombang *narrowband* menjadi bentuk gelombang *wideband*. Setelah disebarkan, energi sinyal *narrowband* pada tiap frekuensi *band* menjadi rendah dan sulit untuk dideteksi..

Pada *image steganography*, data disembunyikan pada *noise* yang ada dan dikombinasikan dengan *cover medium* untuk menghasilkan obyek *steganography*. Jika kekuatan sinyal yang disembunyikan pada *cover medium* lebih rendah daripada kekuatan *cover medium* itu sendiri maka data yang disembunyikan tidak dapat dideteksi oleh mata manusia tanpa memiliki gambar aslinya.

Evaluasi *Image Steganography*

Masing-masing teknik yang telah dijabarkan diatas mempunyai kelebihan dan kekurangan. Baik tidaknya suatu teknik dapat dilihat berdasarkan parameter-parameter yang diidentifikasikan untuk mencapai tujuan *steganography*. Adapun parameter tersebut adalah sebagai berikut:

1. *invisibility*,
2. kapasitas data yang dapat disembunyikan,
3. kemampuan bertahan (*robustness*),
4. kebebasan akan format *file cover medium*, dan
5. kelaziman properti *file*.

Invisibility adalah parameter yang utama dalam menilai suatu obyek *steganography*, dimana

keberhasilan *steganography* diukur dengan tidak terlihatnya obyek *steganography* oleh mata manusia.

Kapasitas data yang dapat disembunyikan menjadi parameter penting karena *steganography* ditujukan untuk menyembunyikan data dan membutuhkan kapasitas data yang cukup untuk data yang akan disembunyikan.

Kemampuan bertahan atau *robustness* adalah kemampuan obyek *steganography* di dalam menghadapi metode pendeteksian obyek *steganography* baik dari manipulasi gambar maupun *image processing*. Dengan kata lain *integrity* data yang disembunyikan harus dijaga dengan benar.

Kebebasan akan format *file cover medium* menjadi parameter yang dipilih karena dari begitu banyaknya format *file* yang ada jika komunikasi antara dua pihak hanya menggunakan satu format *file* yang itu-itu saja maka akan terlihat mencurigakan. Teknik *steganography* yang baik adalah teknik yang mampu menggunakan media apa saja sebagai *cover medium*-nya.

Kelaziman properti *file* dipilih sebagai parameter untuk mencegah penggunaan *file* yang mempunyai ukuran tidak lazim karena dengan menggunakan *file* tersebut akan menimbulkan kecurigaan adanya data tersembunyi didalamnya dan mengundang investigasi lebih lanjut terhadap obyek *steganography* tersebut.
Berdasarkan parameter yang disebutkan diatas, akan dilakukan evaluasi terhadap teknik-teknik *image steganography*.

Evaluasi LSB *Modification* pada format BMP

Terdapat dua parameter utama yang menjadi perhatian dalam LSB *modification* menggunakan format BMP. Dengan format *file* ini, data yang disembunyikan dapat berukuran besar. Oleh karena hal tersebut pula akan ada makin banyak *bit* yang dimodifikasi sehingga perbedaan yang tampak akibat perubahan *bit* tersebut dapat jelas terlihat oleh mata manusia (*visible*). Kekurangan teknik ini yang paling mendasar adalah dengan menggunakan format *file* BMP yang berukuran besar, maka hal tersebut dapat menimbulkan kecurigaan tentang kerberadaan data tersembunyi didalamnya, ditambah lagi kenyataan bahwa format *file* ini sudah jarang digunakan untuk komunikasi antar dua pihak pada jaringan global seperti internet. Teknik ini juga bergantung kepada format *file* yang digunakan.

Rekomendasi: sebaiknya teknik ini digunakan untuk implementasi yang fokus pada besar data yang akan disampaikan dan tidak berfokus pada kerahasiaan data itu sendiri.

Evaluasi LSB Modification pada format GIF

Kekurangan dan kelebihan menggunakan teknik ini tidak jauh berbeda dibandingkan dengan teknik LSB *modification* pada format BMP. Perbedaan utamanya adalah jika dengan menggunakan format ini data yang dapat disembunyikan lebih sedikit dibandingkan jika menggunakan format BMP. Teknik ini rentan terhadap pendeteksian pada *pallete* karena format GIF memang merupakan *file image* yang berbasis *pallete*. Dengan menganalisis *pallete* lebih mudah bagi pihak lain untuk mendeteksi keberadaan data tersembunyi. Teknik ini bergantung pada format *file* yang digunakan. *Robustness* dari teknik ini sangat lemah dimana pemrosesan gambar atau modifikasi gambar seperti *cropping* akan merusak data yang disembunyikan didalamnya.

Rekomendasi: teknik ini cukup efisien jika menggunakan *greyscale image* sebagai *cover image*.

Evaluasi Color Reduction

Salah satu kelemahan mendasar dari teknik ini adalah obyek *steganography* yang dihasilkan terlihat jelas perbedaannya walaupun data yang akan disembunyikan berukuran kecil. Kapsitas data yang disimpan berbanding lurus dengan *invisibility* obyek *steganography* itu sendiri. Teknik ini bergantung kepada format media yang digunakan.
Rekomendasi: teknik ini cocok digunakan untuk menyembunyikan data yang tidak terlalu besar dengan tingkat kerahasiaan data yang rendah.

Evaluasi Masking and Filtering

Teknik ini lebih *robust* dibandingkan dengan teknik yang telah disebutkan di atas. Hal

ini dikarenakan teknik ini menyimpan data yang akan disembunyikan pada area yang signifikan pada *cover medium*. Walaupun tidak bergantung sepenuhnya, teknik ini cukup bergantung pada media yang digunakan sebagai *cover medium* khususnya *image 24-bit* dan *greyscale image*. Data yang disembunyikan tidak berukuran besar.

Rekomendasi: teknik ini berjalan sangat efisien ketika digunakan *greyscale image* sebagai *cover medium*-nya dan data yang disembunyikan tidak besar. Teknik ini lebih cocok digunakan untuk mengimplementasikan *digital watermarks*.

Evaluasi *JPEG Compression*

Teknik ini menghasilkan *stego image* dengan tingkat *invisibility* yang tinggi dimana data disembunyikan pada saat dilakukan pengompresan data. Format ini sangat populer dan mempunyai ukuran *file* yang kecil. Kekurangan yang muncul pada teknik ini adalah proses kompresinya menggunakan proses matematikal yang membuatnya sulit diimplementasikan.

Rekomendasi: teknik ini merupakan teknik yang cukup efisien untuk mengimplementasikan *steganography* dibandingkan teknik-teknik di atas.

Evaluasi Patchwork

Teknik ini mempunyai kelemahan didalam ukuran data yang mampu disembunyikan. Selain itu teknik ini sangat *robust* terhadap manipulasi gambar yang dilakukan pada *cover image*. Beberapa data mungkin dapat hilang tetapi jika dilakukan penambahan data yang berulang-ulang maka sebagian besar data tetap mampu bertahan.

Rekomendasi: teknik ini sangat cocok digunakan untuk menyembunyikan data dalam ukuran kecil dan sensitif.

Evaluasi Spread Spectrum

Teknik ini merupakan teknik yang hampir memenuhi semua persyaratan parameter-parameter yang telah didefinisikan sebelumnya. Kekuatan teknik ini terutama *robustness*-nya terhadap pendeteksian secara statistikal.

Rekomendasi: teknik ini cocok diimplementasikan untuk *steganography* walaupun lebih rumit dibanding teknik lainnya dalam komunikasi internet.

Evaluasi Umum

Dari semua teknik yang telah dijabarkan diatas, didapat suatu hasil yang menunjukan bahwa teknik *image steganography* baik yang menggunakan kombinasi *image domain* dan *transform domain* jauh lebih *powerful* diimplementasikan dibandingkan menggunakan teknik *image domain* ataupun *transform domain*. Mengenai teknik mana pada teknik *transform domain* yang paling baik masih harus dilihat pada kebutuhan implementasinya, parameter mana yang menjadi prioritas. Sebagai contoh jika parameter kapasitas data yang akan disembunyikan yang menjadi prioritas gunakan *spread spectrum* daripada menggunakan *patchwork*. Sebaliknya jika parameter yang menjadi prioritas adalah *robustness* terhadap manipulasi gambar maka teknik *patchwork* sedikit lebih baik dibandingkan *spread spectrum*.

Pada subbab berikutnya akan dipaparkan matriks perbandingan antar masing-masing teknik *image steganography* baik *image domain* maupun *transform domain* dengan parameter-parameter yang didefinisikan sebelumnya.

Kesimpulan

Adapun kesimpulan dari hasil analisis ini adalah sebagai berikut:

- Masing-masing teknik baik *image domain* maupun *transform domain* mempunyai kelebihan dan kekurangan pada masing-masing parameter yang menjadi tujuan *steganography*.
- Teknik *image steganography* yang terbaik adalah perpaduan antara teknik *image domain* dan *transform domain*.
- Dari dua teknik yang mengombinasikan kedua teknik diatas, secara umum teknik *steganography spread spectrum* sedikit lebih baik daripada teknik *steganography patchwork*.

Bagaimana pun suatu sistem keamanan, termasuk keamanan data didalamnya, pasti dapat ditembus. Akan selalu ada cara bagi *cracker*

untuk mengancam dan menyerang keamanan data. Masalah sebenarnya adalah bagaimana meminimalisasi sistem keamanan tersebut dari ancaman serta serangan *cracker* tersebut dan *steganography* khususnya *image steganography* mempunyai kemampuan untuk itu.

Daftar Pustaka

Curran, Kevin, "*An Evaluation of Image Based Steganography Methods*", International Journal of Digital Evidence, Vol 2, Issue 2. 2003.

Dunbar, Bret, "*A Detailed Look at Steganographic Techniques and Their Use In an Open-Systems Environment",* http:// www.sans.org, Januari 2002, 15 Desember 2006.

Johnson, Neil F., dan Sushil Jajodia, "*Steganalysis of Images Created Using Current Steganography Software",* Artikel Lecture Notes in Computer Science, Vol 1525. April 1998.

Krenn, Robert, "St*eganography and Steganalysis*", Penerbit tidak diketahui, 2004.

Marvel, Lisa M., dkk, "*Spread Spectrum Image Steganography",* IEEE Transactions on Image Processing, 8:08, 1999.

Queirolo, Francesco, "*Steganography in Images",* Final Communication Report, 2001.

Silman, Joshua, "*Steganography and Steganalysis: An Overview",* http:// www.sans.org, Agustus 2001, 5 Desember 2006.

W, Bender, dkk, "*Techniques for Data Hiding",* IBM Systems Journal, Vol 35, NOS 3&4, 1996.

L. M. Marvel, C. G. Boncelet Jr., & C. Retter, "*Spread Spectrum Steganography*", IEEE Transactions on Image Processing, 8:08, 1999.