Pembangunan Aplikasi Penyembunyian Pesan yang Terenkripsi dengan Metode *MARS* pada Citra dengan Metode Zhang *LSB* Image

Ferry Pangaribuan - 13505080
Program Studi Teknik Informatika, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung
Email: if15080@students.if.itb.ac.id

Abstract **-** *Sistem yang mengkombinasikan keunggulan kriptografi dan keunggulan steganografi sangat diperlukan saat ini. Kriptografi yang memberikan manfaat pesan dalam keaadaan tidak dapat dibaca dan steganografi yang memberikan manfaat bahwa pesan tidak dapat diketahui keberadaannya merupakan perpaduan yang saling melengkapi.Perbedaan representasi data yang diperlukan pada operasi internal metode Zhang dan metode MARS, sehingga memerlukan sejumlah perubahan representasi data yang diperlukan untuk menghasilkan hasil yang diharapkan. Perubahan representasi data tersebut harus sedapat mungkin terjaga dengan baik dan dapat dikembalikan ke representasi asalnya. Metode Zhang memiliki beberapa kelemahan untuk beberapa kasus, sehingga dilakukan pemodifikasian metode ini pun dilakukan untuk menangani permasalahan pada beberapa kasus yang ada dan membuat modifikasi metode tersebut dapat berjalan dengan baik.*

Kata kunci : steganografi, kriptografi, dan *stego-image*

1. PENDAHULUAN

Penelitian steganografi secara umum dipengaruhi oleh kekurangan dari sistem kriptografi. Banyak pemerintah di beberapa negara telah menciptakan aturan untuk membatasi kekuatan sistem kriptografi, sehingga memaksa orang untuk mempelajari metode lain untuk melakukan pengiriman informasi rahasia. Bisnis-bisnis juga telah mulai menyadari potensi steganografi dalam mengkomunikasikan rahasia-rahasia dagang atau informasi produk baru. Penghindaran komunikasi melalui jalur-jalur yang telah dikenal untuk mengurangi resiko informasi tersebut bocor. Penyembunyian informasi dalam sebuah foto piknik perusahaan memberikan efek mencurigakan yang lebih sedikit daripada komunikasi menggunakan arsip terenkripsi[KHA04].

Oleh karena itu, penggunaan kriptografi yang dapat menyebabkan seseorang langsung mengetahui bahwa pesan yang terkirim adalah sebuah pesan rahasia, sehingga penyembunyian informasi ke dalam sebuah media perlu dilakukan agar menjamin bahwa pesan rahasia tidak diketahui oleh orang lain.

Oleh karena itu, pembuatan aplikasi yang dapat menyediakan kombinasi antara steganografi dan kriptografi perlu untuk diimplementasikan. Hal ini akan meningkatkan kepercayaan pemerintah, bisnis, atau pihak lain yang ingin mengirimkan pesan rahasia secara aman.

Pada makalah ini dibahas mengenai sistem gabungan kriptografi-steganografi dengan metode *Zhang dan MARS* dengan media citra tidak terkompresi *bitmap*.Pemodifikasian terhadap algoritma Zhang dilakukan karena beberapa kasus melibatkan bit terdepan pada saat operasi internalnya.

2. CITRA DIJITAL

Citra dijital memiliki informasi berupa gambar dan terdiri dari elemen terkecil yaitu piksel. Citra dijital direpresentasikan dalam bentuk matriks 2 dimensi yang setiap elemen merepresentasikan piksel pada gambar.

2.1 Warna pada Citra Dijital

Terdapat beberapa jenis pewarnaan pada citra dijital yaitu *duotone* (dua warna), *grayscale* dan citra berwarna. Citra berwarna dapat memiliki sistem pewarnaan *RGB*, *indexed color* atau *256 color*.

Pada citra dijital dengan pewarnaan *duotone*, warna pada piksel hanya memiliki 2 kemungkinan warna, pada umumnya hitam-putih. Dengan penggunaan warna 1-bit, maka kualitas gambar pada citra dijital tidak begitu bagus. Pewarnaan *grayscale* memiliki kualitas lebih baik. Pada *grayscale*, warna yang tersedia hanyalah warna-warna yang ada diantara hitam dan putih, meliputi warna abu-abu yang beragam.

Citra *RGB* adalah yang paling populer saat ini, dimana setiap piksel direpresentasikan dengan intensitas warna merah, hijau dan biru. Citra *indexed color* hanya memiliki 256 warna yang telah didefinisikan pada tabel warna, namun memiliki ukuran *file* yang lebih kecil.

2.2 Citra Dijital Tidak Terkompresi

Struktur *bitmap*, seperti yang tertera pada Lampiran A, secara garis besar dibagi menjadi empat bagian, yaitu *File Header, Image Header, Color Palette,* dan *Pixel Data.* Perlu diperhatikan, bagian *File Header, Image Header,* dan *Color Palette* terdiri atas informasi-informasi yang penting untuk menampilkan citra, apabila terjadi kehilangan data atau kerusakan data pada bagian-bagian ini maka hal tersebut akan mengakibatkan citra rusak atau bahkan tidak bisa ditampilkan.

Agar *stego-image* dapat ditampilkan persis dengan aslinya, dalam melakukan steganografi, yang disisipi pesan hanya bagian *pixel data* saja karena jika bagian *file header, image header,* dan *color palette* ikut disisipi pesan, maka bagian citra tidak dapat ditampilkan lagi. Sebagai contoh, salah satu bagian dari *file header* adalah bfType yang mengandung karakter "BM" yang mengidentifikasi tipe arsip, apabila tipe arsip ini disisipi pesan, maka tipe arsip dapat berubah menjadi tidak dikenali sehingga citra tidak dapat ditampilkan. Hal ini menunjukkan bahwa penyisipan pesan dengan teknik *LSB* hanya dapat dilakukan pada bagian *pixel data,* agar citra yang menyembunyikan tidak rusak.

Pada representasi arsip 24-bit bitmap, setiap piksel akan terdiri dari 3 byte karena setiap 1 byte akan merepresentasikan nilai red, blue, atau green. Apabila terdapat pemilihan nilai LSB tertentu akan dibagi secara merata pada 3 representasi warna tersebut. Contoh apabila nilai LSB adalah 20; maka 6 bit LSB red, 7 bit LSB blue, dan 7 bit LSB green. Jadi apabila terjadi nilai m LSB yang jika dimodulus 3 lebih besar dari nol, maka nilai bit LSB green yang pertama kali ditambahkan baru kemudian nilai bit LSB blue.

3. METODE LSB

Pengubahan *LSB (Least Significant Bit)* pada citra yang tidak terkompresi sangat sulit untuk diketahui secara kasat mata, sehingga metode ini sangat banyak digunakan. Metode ini memanfaatkan ketidakmampuan mata manusia dalam menemukan perbedaan antara antara citra asli dengan yang sudah dimasukkan pesan.

Untuk menjelaskan metode ini, digunakan citra dijital sebagai *cover-object*. Setiap piksel dalam citra dijital berukuran 1 sampai 3 *byte*. Pada susunan bit di dalam *byte* (1 *byte* = 8 bit), terdapat bit yang memiliki arti yang paling kecil (*Least Significant bit* atau *LSB*). Misalnya pada *byte* 0001100**1,** maka bit *LSB*-nya adalah bita yang terletak di paling kanan yaitu **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 piksel 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)

4 . ALGORITMA ZHANG LSB IMAGE STEGANOGRAPHY

Algoritma ini dikembangkan oleh Hong-Juan Zhang dan Hong-Jun Tang dari Universitas Hangzhou Dianzi. Algoritma ini dapat bertahan terhadap serangan steganalisis yang berdasarkan kepada analisis statistikal seperti RS-Analysis dan Chi-Square.

4.1 Penanaman Pesan

Proses penanaman pesan ke dalam citra menggunakan *generator* angka *pseudo-random*. Misalkan $G = < x_0, x_1, x_2, \ldots \ldots, x_n >$ kumpulan piksel-piksel yang dipilih oleh angka *pseudo-random*. Sebuah *stego-key* digunakan sebagai benih dari *generator* angka *pseudo-random*, *x* adalah nilai dari piksel itu, *n* ditentukan dari ukuran pesan

yang ditanamkan dan berapa banyak bit-bit *LSB* dalam setiap piksel yang dapat digunakan untuk menanamkan pesan. Ini dapat dihitung dengan menggunakan fungsi ini:

$$n = \frac{l}{m} \qquad (2.1)$$

dimana *l* adalah panjang *bit stream* dari pesan yang ditanamkan, *m* adalah jumlah bit yang digunakan untuk menanamkan pesan-pesan dalam setiap piksel, dan n adalah sejumlah kumpulan *m* bit pesan. *Bit stream* dari pesan yang ditanamkan dibagi menjadi bit segmen dengan panjang *m* bit dan dinotasikan dengan $E = < e_1, e_2, \ldots \ldots, e_n >$, $e \in \{0,1, \ldots, 2^m - 1\}$. Didefinisikan $LSB_m(x)$ menjadi fungsi untuk mendapatkan nilai m bit *LSB* dari *x* tersebut dan didefinisikan *MaxVal* sebagai nilai $2^p - 1$ (dimana p adalah banyaknya bit yang merepresentasikan setiap piksel).

Untuk melakukan penanaman pesan dilakukan seperti *pseudo* algoritma pada Gambar 1.

```
for  i = 1, 2, …., n do
x_i = x_i + e_i − (LSB_m(x_{i-1}) + LSB_m(x_i)) Mod 2^m;
        if  x_i > MaxVal then
                x_i = x_i - 2^m;
        end
        if x_i <  0 then
                x_i = x_i + 2^m;
        end
end
```

Gambar 1 *Pseudocode* penanaman pesan

4.2 Pengekstrakan Pesan

Dengan menggunakan *stego-key* yang sama untuk membangkitkan angka *pseudo-random* tersebut, piksel-piksel yang bersesuain dipilih dengan menggunakan angka *pseudo-random* untuk membangun $G = < x_0, x_1, x_2, \ldots \ldots, x_n >$.

Pesan dapat diekstrak seperti *pseudo* algoritma pada Gambar 2.

```
for i= 1, 2, …., n do
        e_i = (LSB_m(x_{i-1}) + LSB_m(x_i)) Mod 2^m
end
```

Gambar 2 *Pseudocode* proses ekstraksi

sehingga didapatkan $E = < e_1, e_2, \ldots \ldots, e_n >$. dan dapat membangun kembali pesan tersebut.

5. ALGORITMA MARS

Input dan output metode ini berupa 4 *word* data 32-bit. Metode ini merupakan metode yang berorientasi *word*, karena semua operasi internalnya dilakukan dalam *word* 32-bit. Kode yang sama untuk mesin dengan struktur internal *little-endian* dapat digunakan untuk mesin dengan struktur internal *big-endian.* Ketika input atau output berupa sebuah *byte stream,* digunakan susunan *byte little-endian* untuk menginterpretasikan setiap 4 *byte* sebagai sebuah *word* 32-bit. Gambar II-11 menunjukkan struktur algoritma *MARS*.

Tahap enkripsi dan dekripsi dilakukan dalam 3 fase:

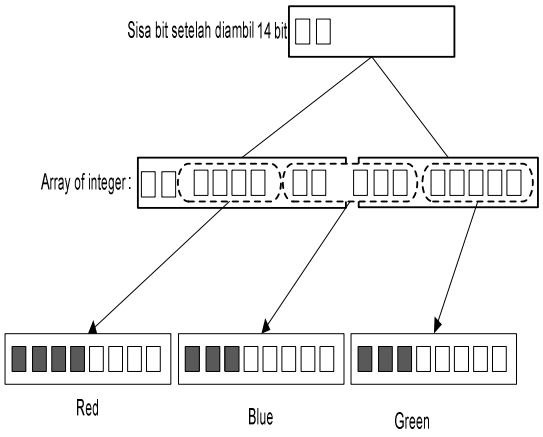
1. Fase pertama ini menyediakan *rapid mixing* dan *key avalanche,* ini berfungsi untuk mengatasi serangan *chosen-plaintext* dan membuat lebih sulit untuk melakukan penyerangan dengan metode linier dan diferensial. Fase ini terdiri dari penambahan kunci ke data, diikuti dengan delapan putaran *S-box.*
2. Fase kedua disebut "*cryptographic core*" metode ini, terdiri dari 16 putaran dari transformasi Feistel tipe-3. Ini untuk memastikan bahwa enkripsi dan dekripsi memiliki kekuatan yang sama, dilakukan 8 putaran pertama dalam "*forward mode*" dan 8 putaran terakhir dalam "*backwards mode*".
3. Fase terakhir menyediakan kembali *rapid mixing* dan *key avalanche,* pada saat ini untuk melindungi dari serangan *chosen-chipertext.* Fase ini merupakan inverse dari fase pertama, terdiri dari 8 putaran Feistel tipe-3 seperti yang terdapat pada fase pertama (namun dalam "*backwards mode*" sedangkan fase pertama dalam "*forward mode*"), diikuti dengan substraksi kunci dari data.

6. ANALISIS

6.1 Analisis Struktur Data yang Terlibat

Sejumlah perubahan representasi data diperlukan untuk membangun sistem kombinasi ini. Algoritma Zhang yang operasi internalnya yang berorientasi *word,* sedangkan algoritma Zhang yang operasi internalnya berorientasi bit dan *byte*; sehingga perubahan representasi perlu dikendalikan agar memberikan hasil yang diharapkan.

Gambar 3 mengilustrasikan perubahan representasi yang dilakukan. Pada contoh tersebut diambil nilai m *LSB* bernilai 14, sehingga akan diisi 4 bit terakhir untuk nilai komponen *red,* 5 bit terakhir untuk nilai komponen *blue,* dan 5 bit terakhir untuk nilai komponen *green* dari bit representasi *array of integer* tersebut. Sisa bit terdepan akan diisi dengan nilai 0. Sisa 2 bit dari *array of integer* tersebut akan digunakan lagi untuk membentuk *array of integer[3]* berikutnya.



Gambar 3 Perubahan representasi dari *array of integer* ke *array of integer[3].*

6.2 Pemodifikasian Algoritma Zhang

Pada algoritma *Zhang LSB Image Steganography* terdapat satu kejanggalan yang akan menyebabkan kualitas suatu piksel akan berkurang karena persamaan

$$e_i - \left(LSB_m(x_{i-1}) + LSB_m(x_i)\right) Mod\ 2^m$$

memiliki kemungkinan untuk bernilai negatif. Apabila nilai yang dihasilkan negatif dan nilai m *LSB* dari suatu piksel yang akan melakukan operasi penambahan dengan nilai ini bernilai 0(nol) semua maka bit-bit yang lebih tinggi akan berpengaruh dan berkemungkinan merusak kualitas citra secara keseluruhan.

Gambar 3 menunjukkan hasil pemodifikasian tahap akhir dari algoritma Zhang.

for i = 1, 2, …., n do

$LSB_m(x_i) = LSB_m(x_i) + e_i - \left(LSB_m(x_{i-1}) + LSB_m(x_i)\right) Mod\ 2^m$;

if $LSB_m(x_i) >= 2^m$ then

$LSB_m(x_i) = LSB_m(x_i)$ mod 2^m;

end

if $LSB_m(x_i) < 0$ then

$LSB_m(x_i) = LSB_m(x_i) + 2^m$;

end

end

Gambar 4 Pemodifikasian akhir algoritma Zhang

6.3 Penentuan Ukuran Pesan yang dapat Ditanamkan

Berdasarkan dasar teori yang terdapat pada bab II, diketahui bahwa penentuan jumlah *least significant bit* yang dipilih untuk setiap piksel dan ukuran citra akan mempengaruhi ukuran pesan yang dapat ditanamkan. Misalkan ukuran citranya adalah 1028 x 700 dan jumlah *LSB* yang dipilih adalah 2, maka ukuran maksimal dari pesan yang dapat ditanamkan adalah 1028 * 700 * 2 bit (1.439.200 bit atau 179,9 kilo *byte*).

Oleh karena itu, diperlukan suatu prosedur untuk melakukan verifikasi terhadap pesan yang ingin ditanamkan setelah melakukan pemilihan jumlah *LSB* dan citra tertentu agar tidak terdapat beberapa bagian pesan yang tidak dapat ditanamkan. Prosedur ini akan memperkirakan terlebih dahulu nilai dari ukuran maksimal pesan yang dapat ditanamkan setelah pengguna meng-input jumlah *LSB* dan citra, kemudian nilai ini akan digunakan saat pengguna melakukan konfirmasi terhadap pesan yang akan ditanamkan. Apabila ukurannya lebih besar dari nilai maksimal tersebut, maka pengguna akan diminta untuk memperkecil ukuran pesan tersebut.

6.4 Penentuan LSB dari Setiap Piksel

Pada representasi arsip 24-bit bitmap, setiap piksel akan terdiri dari 3 *byte* karena setiap 1 *byte* akan merepresentasikan nilai *red, blue,* atau *green.* Apabila terdapat pemilihan nilai *LSB* tertentu akan dibagi secara merata pada 3 representasi warna tersebut. Contoh apabila nilai *LSB* adalah 20; maka 6 bit *LSB red*, 7 bit *LSB blue*, dan 7 bit *LSB green*. Jadi apabila terjadi nilai m *LSB* yang jika dimodulus 3 lebih besar dari nol, maka nilai bit *LSB green* yang pertama kali ditambahkan baru kemudian nilai bit *LSB blue*

7. HASIL DAN PENGUJIAN

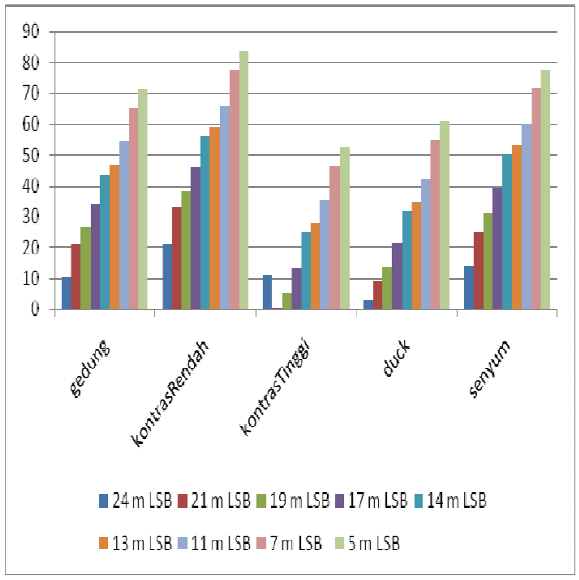
Dari hasil pengujian, perangkat lunak CombinoZM dapat menangani beberapa ekstensi arsip pesan seperti jpg, java (arsip teks), zip, docx, dan pdf. Teknik gabungan steganografi dan kriptografi berjalan sesuai dengan fungsinya masing-masing.

Pemilihan *cover-image* yang memiliki kontras tinggi dan *cover-image* yang kontras rendah tidak akan mempengaruhi hasil dari *stego-image.* Dari hasil pengujian, parameter kontras suatu citra tidak mempengaruhi kualitas dari *stego-image* yang dihasilkan. Parameter yang menentukan suatu *cover-image* akan menghasilkan *stego-image* yang baik adalah nilai m *LSB* yang dipilih.

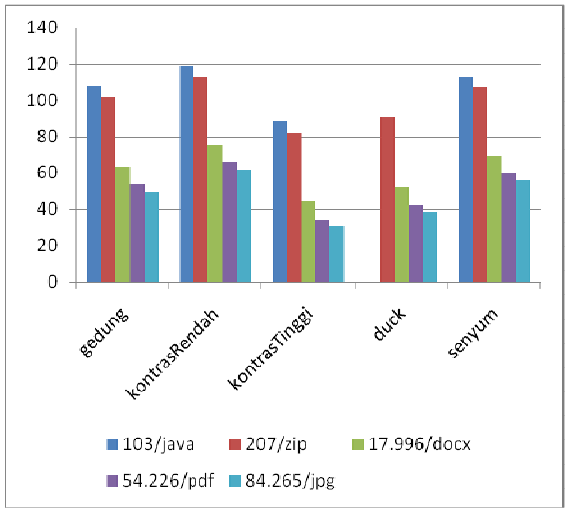
Seperti yang dijelaskan di atas dan diperkuat dengan Gambar V-6, kualitas *stego-image* yang dihasilkan bergantung pada nilai m *LSB* yang dipilih, semakin besar nilai m *LSB* maka kualitas gambar semakin berkurang/rusak. Pada hasil pengujian, bila m *LSB* lebih besar dari 8 maka akan terdapat bintik-bintik pada *stego-image* yang dihasilkan. *Stego-image* yang baik dapat dihasilkan dengan memilih nilai m *LSB* lebih kecil dari 8.

Pada saat memilih nilai m *LSB* yang lebih kecil dengan nilai ukuran arsip pesan yang sama, akan terjadi peningkatan kualitas *stego-image* yang dihasilkan. Pesan yang disebar pada piksel yang lebih banyak dengan pemilihan m *LSB* ini menyebabkan perubahan yang terjadi pada *cover-image* semakin sedikit dan semakin meningkat kualitas dari *stego-image.*

Parameter yang juga ikut menentukan kualitas dari *stego-image* yang dihasilkan adalah ukuran dari arsip pesan yang ingin ditanamkan. Pada pengujian dengan menggunakan *PSNR* yang ditunjukkan pada Gambar 5, dihasilkan kesimpulan bahwa semakin besar ukuran arsip pesan yang ditanamkan maka kualitas dari *stego-image* semakin menurun. Penentuan ukuran arsip yang sangat berbeda jauh dengan ukuran *cover-image* akan menentukan kualitas dari *stego-image* yang dihasilkan, seperti yang ditunjukkan oleh hasil pengujian pada *cover-image* kontrasRendah di Gambar 4 dan Gambar 5, hasil *cover image*-nya menghasilkan kualitas yang paling baik dari semua hasil *cover-image* yang ada karena ukuran *cover-image*-nya paling besar.



Gambar 5 Nilai *PSNR* dengan nilai m *LSB* yang bervariasi dan ukuran arsip pesan yang sama



Gambar 6 Nilai *PSNR* dengan nilai m *LSB* tetap dan nilai ukuran pesan yang berubah-ubah

Pemilihan ukuran arsip pesan dan ukuran *cover-image* akan mempengaruhi lamanya proses untuk melakukan penanaman arsip pesan maupun proses sebaliknya. Dari proses besar enkripsi penanaman terdiri dari bagian-bagian proses yang kecil yaitu proses pembacaan *cover-image,* proses pembacaan arsip pesan, proses enkripsi arsip pesan, proses penentuan piksel mana yang akan dimodifikasi, proses transformasi represntasi dari arsip pesan, proses modifikasi piksel yang digunakan untuk

menanam arsip pesan, dan proses pembentukan *stego-image;* proses pembentukan *cover-image* dan proses modifikasi piksel yang digunakan untuk menanam arsip pesan merupakan proses yang menyumbang waktu terlama dalam proses besar enkripsi penanaman. Sehingga performansi dari CombinoZM akan bergantung dari ukuran *cover-image* dan ukuran arsip pesan.

Pengukuran performansi untuk proses besar ekstraksi dan dekripsi sama halnya dengan proses besar enkripsi penanaman. Proses besar ekstraksi dekripsi terdiri dari bagian-bagian proses yang kecil yaitu proses penentuan piksel mana yang akan dimodifikasi, proses pengekstrakan arsip pesan, proses perubahan representasi arsip pesan, proses dekripsi dari arsip pesan, dan proses pembentukan arsip pesan. Dari bagian kecil tersebut, proses pembentukan arsip pesan adalah proses yang menentukan berapa lama proses besar ekstraksi dekripsi berlangsung.

8. KESIMPULAN

Kesimpulan yang didapat selama pengerjaan makalah ini adalah sebagai berikut:

1. Sebuah perangkat lunak yang mengimplementasikan kriptografi *chiper block* dan steganografi metode *LSB* telah berhasil dibangun.
2. Perubahan representasi data yang terjadi dari metode Zhang ke metode MARS atau sebaliknya sulit untuk dikelola karena representasi data yang diperlukan pada kedua metode ini berbeda.
3. Algoritma *Zhang LSB Image Steganography* tidak menghasilkan kualitas *stego-image* yang baik apabila ukuran m *LSB* nya mendekati ukuran yang diperlukan untuk merepresentasikan sebuah piksel dalam sebuah citra.
4. Algoritma *MARS* memiliki performansi yang baik karena dalam sistem gabungan ini kontribusi waktu yang diberikan terhadap proses keseluruhan tidak signifikan/relatif lebih kecil dibanding proses-proses yang lain.
5. Penentuan parameter m *LSB*, ukuran arsip pesan , dan ukuran *cover-image* akan menentukan kualitas dari *stego-image* yang dihasilkan. Apabila terdapat bintik-bintik pada *stego-image* maka penentuan ketiga parameter tersebut dapat diulang kembali untuk mendapat hasil yang baik.
6. Penentuan parameter ukuran arsip pesan dan ukuran citra akan berpengaruh pada performansi CombinoZM

DAFTAR REFERENSI

[BUR99] Burwick, Carolynn dkk. *MARS-a candidate chiper for AES*. 1999. IBM Corporation.

[JOH98] Johnson, Neil F dan Jajodia Sushil. *Exploring Steganography: Seeing the Unseen*. 1998. George Mason University.

[KHA04] Kharrazi, Mehdi dkk. *Image Steganography: Concepts and Practice*. 2004. Brooklyn : Departement of Electrical and Computer Engineering and Departement of Computer and Information Science Polytechnic University Brooklyn.

[KRU02] Kruus, Peter, Caroline Scace, Michael Heyman, dan Mathew Mundy. *A Survey of Steganographic Techniques for Image Files*. 2002. Advanced Security Research Journal – Network Associates Laboratories, Network Associates, Inc.

[LYN02] Lynch and Horton, *Graphic: Color Displays,* 2002. URL : http://webstyleguide.com/graphics/displays .html. Tanggal akses 25April2009.

[MIA99] Miano, John. *Compressed Image File Formats*. 1999. Massachusetts : Addison Wesley Longman, Inc.

[MOR] Morkel, T., JHP. Eloff, dan MS. Olivier. *An Overview of Image Steganography*. Pretoria: Information and Computer Security Architecture (ICSA) Research Group, Department of Computer Science, University of Pretoria.

[PRO03] Provos Neils dan Honeyman Peter. *Hide and Seek: An Introduction to Steganography*. 2003. University of Michigan.

[ZHA07] Zhang Hong-Juan dan Tang Hong-Jun. *A Novel Image Steganography Algorithm Against Statistical Analysis*. 2007. Hangzhou: Institute of Intelligence and Software Technology, Hangzhou Dianzi University.

[KAT00] Katzenbeisser S dan Petitcolas F. *Information Hiding Techniques for Steganography and Digital Watermarking*. 2000. Norwood : Artech House