Metode *Asymmetric Watermarking* pada Citra Digital Berbasiskan pada Permutasi-*RC4* dan Fungsi *Chaos*

Rinaldi Munir[1)], **Bambang Riyanto**[1)], **Sarwono Sutikno**[1)], **Wiseto P. Agung**[2)]
1) Sekolah Teknik Elektro dan Informatika ITB, Bandung 40132, email: rinaldi@informatika.org
2) PT Telekomunikasi Indonesia, Bandung , email: wiseto@telkom.co.id

Abstract *– Makalah ini memaparkan metode* asymmetric watermarking *pada citra digital yang menggunakan algoritma kriptografi RC4 dan fungsi* chaos. Asymmetric watermarking *menggunakan kunci yang berbeda untuk menyisipkan dan mendeteksi* watermark. *Kunci publik adalah watermark publik yang berupa barisan nilai riil berdistribusi normal, sedangkan kunci privat adalah watermark privat yang merupakan permutasi watermark publik dengan menggunakan kombinasi algoritma RC4 dan fungsi chaos. Watermark disisipkan pada koefisien DCT yang dipilih dari sub-band middle frequency. Pendeteksian watermark dilakukan dengan menghitung korelasi antara citra yang diterima dengan waternark publik. Hasil eksperimen menunjukkan bahwa metode ini terbukti* robust *terhadap beberapa serangan non-malicious attack (kompresi JPEG, rotasi,* cropping, resizing, noising, sharpening*) dan* malicious attack.

Kata Kunci*: asymmetric watermarking, RC4, chaos, DCT, korelasi.*

1. PENDAHULUAN

Digital watermarking adalah teknik menyisipkan informasi *copyright* (disebut *watermark*) ke dalam data multimedia (citra, adio, video) sehingga keberadaan *watermark* dapat digunakan untuk mengontrol penggandaan dan distribusi data multimedia tersebut [1]. Persyaratan utama skema *digital watermarking* adalah *imperceptibility*, *robustness*, dan *security* [2]. Ada dua proses utama di dalam skema *watermarking*, yaitu penyisipan dan pendeteksian *watermark*. Kedua proses ini menggunakan kunci agar hanya pihak yang punya otoritas yang dapat melakukannya.

Skema *watermarking* ada dua macam: simetri dan nirsimetri. Skema simetri (*symmetric watermarking*) menggunakan kunci yang sama untuk menyisipkan dan mendeteksi *watermark*. Skema simetri mempunyai kelemahan mendasar, yaitu sekali penyerang mengetahui kunci dan semua parameter penting lainnya (termasuk algoritma *watermarking* yang bersifat publik), ia dapat menggunakan informasi tersebut untuk menghapus *watermark* dari data multimedia tanpa menimbulkan kerusakan berarti. Hal ini dimungkinkan karena pada kebanyakan sistem simetri kunci adalah *watermark* itu sendiri atau nilai yang yang menspesifikasikan lokasi penyisipan *watermark* di dalam data multimedia.

Kelemahan skema simetri dapat diatasi dengan menggunakan skema nirsimetri (*asymmetric watermarking*). Pada skema ini digunakan kunci (dalam hal ini kunci = *watermark*) yang berbeda untuk penyisipan dan pendeteksian. Skema *asymmetric watermarking* dikatakan *public-key watermarking* jika kunci yang digunakan untuk deteksi dipublikasikan, sehingga kunci tersebut dinamakan juga kunci (*watermark*) publik. Kunci yang digunakan pada proses penyisipan *watermark* dirahasiakan sehingga dinamakan kunci (*watermark*) privat. Kedua kunci ini berkorelasi satu sama lain. Skema *public-key watermarking* dilakukan dengan suatu cara sedemikian sehingga: (a) secara komputasi tidak mungkin menghitung kunci privat dari kunci publik, dan (b) kunci publik tidak dapat digunakan oleh penyerang untuk menghilangkan *watermark* [3]. Review beberapa metode *asymmetric watermarking* awal dapat ditemukan di dalam [4].

Umumnya pendeteksian di dalam skema *asymmetric watermarking* dilakukan dengan menghitung korelasi antara *watermark* publik dan data multimedia yang diterima [6]. Nilai korelasi ini kemudian dibandingkan dengan sebuah nilai-ambang (*threshold*) yang telah dispesifikasikan untuk menentukan apakah data multimedia tersebut mengandung *watermark*.

Di dalam makalah ini disajikan metode *asymmetric watermarking* yang berbasiskan pada operasi permutasi. Permutasi *watermark* publik untuk menghasilkan *watermark* privat dilakukan dengan menggunakan sebagian algoritma kriptografi *RC4* dan fungsi *chaos*. *Chaos* diterapkan karena ia mempunyai karakteristik penting untuk meningkatkan keamanan, yaitu sensitivitas pada kondisi awal. Karakteristik ini cocok untuk enkripsi dan *watermarking* [8]. Data multmedia yang disisipi *watermark* adalah citra *greyscale*. Baik penyisipan maupun pendeteksian *watermark* keduanya dilakukan pada ranah *discrete cosine transform* (*DCT*).

2. FUNGSI *CHAOS*

Beberapa tahun terakhir teori *chaos* banyak digunakan di dalam *digital watermarking* [4,6]. *Chaos* digunakan khususnya sebagai pembangkit bilangan acak.

Salah satu fungsi *chaos* sederhana adalah persamaan logistik (*logistic map*). Persamaan logistik dinyatakan sebagai

$$x_{i+1} = r\, x_i\, (1 - x_i) \qquad (1)$$

dengan x_0 sebagai nilai awal iterasi. Konstanta r menyatakan laju pertumbuhan fungsi, yang dalam hal ini $0 \le r \le 4$. Dengan melakukan iterasi persamaan (1) dari nilai awal x_0 tertentu, kita memperoleh barisan nilai-nilai *chaos*. Nilai-nilai *chaos* tersebut terletak di antara 0 dan 1 dan tersebar secara merata serta tidak ada dua nilai yang sama.

Karakteristik umum sistem *chaos* adalah kepekaannya terhadap perubahan kecil nilai awal (*sensitive dependence on initial condition*). Kepekaan ini berarti bahwa perbedaan kecil pada nilai awal fungsi, setelah fungsi diiterasi sejumlah kali, akan menghasilkan perbedaan yang sangat besar pada nilai fungsinya [9, 10].

3. PERMUTASI-*RC4*

Ada banyak cara untuk membangkitkan *watermark* privat dari *watermark* publik yang saling berkorelasi satu sama lain. Salah satu cara adalah dengan mempermutasikan elemen-elemen *watermark* publik. Sejumlah teknik permutasi yang ada dapat digunakan untuk tujuan ini. Algoritma *watermarking* di dalam makalah ini memodifikasi sebagian algoritma kriptografi *RC4* untuk melakukan permutasi.

Algoritma *RC4* termasuk ke dalam kelompok *stream cipher*. Inti algoritma *RC4* adalah membangkitkan kunci-aliran (*keystream*) yang kemudian di-*XOR*-kan dengan plainteks. Algoritma *RC4* menggunakan larik *S*[0..255] yang diinisialisasi dengan 0, 1, 2, …, 255. Selanjutnya larik *S* dipermutasi berdasarkan kunci eksternal *U* yang panjangnya variabel. Jika panjang *U* < 256, maka lakukkan *padding* sehingga panjangnya menjadi 256 *byte*. Permutasi terhadap nilai-nilai di dalam larik *S* dilakukan dengan cara sebagai berikut:

```
j ← 0
for i ← 0 to 255 do
   j ← (j + S[i] + U[i]) mod 256
   swap(S[i], S[j])
end
```

Algoritma permutasi di atas dimodifikasi untuk mempermutasikan *watermark* publik. Misalkan panjang *watermark* adalah *N*, maka larik *S*[0..*N*] yang diinisialisasi dengan 0, 2, …, *N* – 1. *U* adalah sebuah larik *integer* sepanjang *N* yang elemen-elemennya dibangkitkan dengan *logistic map* (nilai *chaos* yang riil terlebih dahulu dikalikan dengan *N* dan dibulatkan ke *integer* terdekat). Selanjutnya permutasi terhadap nilai-nilai di dalam larik *S* dilakukan dengan cara yang sama seperti potongan algoritma di atas kecuali 255 diganti dengan *N*. Nilai-nilai di dalam larik *S* digunakan untuk mempermutasikan *watermark* publik.

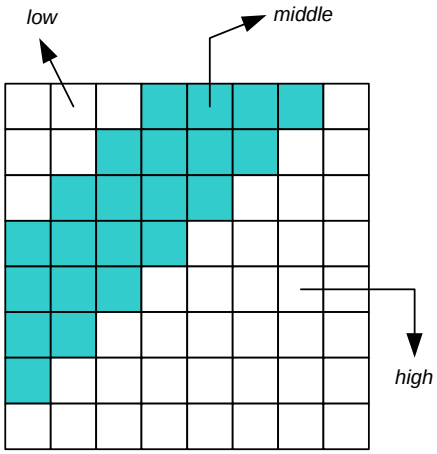
4. *WATERMARKING* DALAM RANAH *DCT*

Menyisipkan dan mendeteksi *watermark* dalam ranah *transform* menghasilkan *robustness* yang lebih tinggi bila dibandingkan dengan ranah spasial. Dalam hal ini, citra ditransformasikan ke dalam ranah *transform*, kemudian *watermark* disisipkan pada koefisien-koefisien *transform* yang dipilih. Selanjutnya lakukan transformasi balikan untuk mendapatkan citra ber-*watermark*. Metode *watermarking* di dalam makalah ini menggunakan transformasi *DCT*. Transformasi *DCT* dapat dilakukan terhadap keseluruhan citra atau pada blok-blok citra berukuran 8 × 8. Dengan mengacu pada kompresi JPEG, *watermarking* berbasis blok berukuran 8 × 8 umumnya lebih *robust* [12]. Metode *watermarking* dalam makalah ini berbasis blok 8 × 8.

Ranah *DCT* membagi citra ke dalam tiga *sub-band* frekuensi (*low*, *middle*, dan *high*), lihat Gambar 1. Penyisipan pada bagian *low frequency* dapat merusak citra karena mata manusia lebih peka pada frekuensi yang lebih rendah daripada frekuensi lebih tinggi. Sebaliknya bila *watermark* disisipkan pada bagian *high frequency*, maka *watermark* tersebut dapat terhapus oleh operasi kuantisasi seperti pada kompresi *lossy* (misalnya *JPEG*). Oleh karena itu, untuk menyeimbangkan antara *robustness* dan *imperceptibility*, maka *watermark* disisipkan pada bagian *middle frequency* (bagian yang diarsir pada Gambar 1).

5. METODE YANG DIUSULKAN

Metode *asymmetric watermarking* di dalam makalah ini didasarkan pada skema umum dari Guo [5]. Ada tiga tahapan proses yang dilakukan di dalam metode *asymmetric watermarking* ini. Masing-masing tahap dijelaskan di dalam sub-bab berikut.



Gambar 1. Pembagian tiga kanal frekuensi pada blok *DCT* berukuran 8 × 8

5.1 Pembangkitan *Watermark* Publik dan Privat

Watermark yang akan disisipkan memiliki ukuran kira-kira seperempat dari ukuran citra. Jika citra berukuran $N_1 \times N_2$, maka *watermark* berukuran $N_1N_2/4$. *Watermark* adalah barisan bilangan riil semi-acak yang mempunyai distribusi normal dengan rerata = 0 dan variansi = 1 (notasi: $N(0, 1)$).

Mula-mula bangkitkan *watermark* publik $\mathbf{w_p}$ berdasarkan $N(0, 1)$:

$$\mathbf{w_p} = (w_p(1), w_p(2), ..., w_p(N))$$

Selanjutnya, hasilkan *watermark* privat $\mathbf{w_s}$ dengan mempermutasikan $\mathbf{w_p}$ berdasarkan tabel permutasi S yang diperoleh dengan algoritma RC4 (lihat bagian 3), yaitu:

$$\mathbf{w_s} = (w_s(1), w_s(2), ..., w_s(N))$$
$$= (w_p(S(1)), w_p(S(2)), ..., w_p(S(N)))$$

Tabel permutasi S ini dijaga tetap rahasia, begitupun nilai awal *logistic map* untuk pembangkitan kunci eksternal U.

5.2 Penyisipan *Watermark*

Watermark yang disisipkan ke dalam citra merupakan kombinasi dari *watermark* publik dan *watermark* dengan pembobotan tertentu. *Watermark* ini diperoleh dengan rumus berikut [5]:

$$\mathbf{w_e} = (1 - \alpha)\mathbf{w_s} + \alpha\mathbf{w_p} \qquad (2)$$

yang dalam hal ini $0 < \alpha < 1$ adalah faktor pembobotan untuk mengontrol nilai ambang deteksi secara publik.

Citra I yang berukuran $N_1 \times N_2$, dibagi menjadi blok-blok kecil berukuran 8×8. Setiap blok ditransformasi dengan *DCT*, lalu koefisien *DCT* dipindai secara *zig-zag* dan semua koefisien *DCT* pada bagian *middle frequency* diambil. Misalkan koefisien-koefisien *DCT* yang terpilih ini disimpan di dalam larik f. Penyisipan *watermark* ke dalam f dilakukan dengan persamaan berikut [12]:

$$f_w(i) = f(i) + \gamma w_e(i) \qquad (3)$$

yang dalam hal ini γ adalah faktor kekuatan *watermark* ($0 < \gamma < 1$) yang dipilih sedemikian rupa sehingga *watermark* tidak dapat dipersepsi secara visual namun masih dapat dideteksi.

Terakhir, terapkan transformasi *DCT* balikan (*IDCT*) pada setiap blok untuk mendapatkan citra ber-*watermark*.

5.3 Pendeteksian *Watermark*

Pendeteksian *watermark* tidak membutuhkan citra asal maupun *watermark* privat, tetapi hanya membutuhkan *watermark* publik yang berkorelasi dengan *watermark* privatnya. Hasil pendeteksian ada dua kemungkinan: citra mengandung *watermark* atau tidak mengandung *watermark*.

Citra yang diterima dibagi menjadi blok-blok berukuran 8×8, lalu koefisien *DCT* pada bagian *middle frequency* (yang mungkin mengalami kerusakan karena *non-malicious attack*) diekstraksi. Misalkan koefisien-koefisien *DCT* yang diekstraksi ini disimpan di dalam larik f^*. Pendeteksian dilakukan dengan menghitung korelasi antara f^* dan *watermak* publik $\mathbf{w_p}$:

$$c = \frac{1}{N}\sum_{i=i}^{N} f^*(i) \cdot w_p(i) \qquad (4)$$

Keputusan ada tidaknya *watermark* di dalam citra uji ditentukan dengan membandingkan nilai c dengan sebuah nilai ambang T. Citra mengandung *watermark* bila $|c| > T$, sebaliknya dikatakan citra tidak mengandung *watermark*. Nilai ekspekstasi korelasi adalah $E(c) = \alpha\gamma$ var($\mathbf{w_p}$) [5] yang dalam hal ini var($\cdot$) adalah operator variansi. Kita dapat memilih nilai ambang $T = \alpha\gamma$ var($\mathbf{w_p}$)/2.

6. EKSPERIMEN DAN HASIL

Metode ini diuji dengan menggunakan kakas MATLAB 7. Citra yang digunakan adalah citra *peppers* (256×256). *Watermark* yang disisipkan berukuran 128×128 ($N = 16384$) dan mempunyai distribusi normal dengan rerata = 0 dan variansi = 1. Nilai awal *logistic map* yang digunakan adalah $x_0 = 0.5$. Nilai $\alpha = 0.2$ dan nilai $\gamma = 0.8$. Nilai ambang T yang digunakan adalah $T = \alpha\gamma$ var($\mathbf{w_p}$)/2 = 0.08 (catatan: var($\mathbf{w_p}$) $\approx$ 1). Gambar 2(a) memperlihatkan citra asal dan Gambar 2(b) adalah citra yang telah mengandung *watermark* (*PSNR* = 57,0632).

Pada kasus tidak ada serangan, nilai korelasi yang dihasilkan adalah $c = 0.2501$, lebih besar dari nilai ambang T, yang berarti citra tersebut mengandung *watermark*. Jika citra yang diuji tidak mengandung

(a) (b)

Gambar 2. (a) Citra *peppers* asli, (b) citra *peppers* yang sudah mengandung *watermark* (*PSNR* = 57,1).

watermark (dalam eksperimen ini digunakan citra *peppers* yang asli), maka c = 0.1299, lebih kecil dari T, yang berarti citra tersebut memang tidak mengandung *watermark*.

Eksperimen selanjutnya dilakukan untuk melihat kekokohan *watermark* terhadap berbagai serangan *non-malicious attack*, yaitu operasi tipikal yang umum dilakukan pada pengolahan citra (*cropping*, kompresi, dll). Program pengolahan citra yang digunakan adalah *Jasc Paintshop Pro*.

Eksperimen 1: Pemotongan (*cropping*)

Watermark masih dapat dideteksi dari citra ber-*watermark* meskipun citra tersebut dipotong kira-kira 25% dan 50% (Gambar 3). Nilai korelasinya masih lebih tinggi daripada nilai ambang T.

Eksperimen 2: Kompresi *JPEG*

Citra ber-*watermark* dikompresi ke format *JPEG* dengan kualitas kompresi 100%, 60%, 20%. *Watermark* masih dapat dideteksi dari citra hasil kompresi (Gambar 4) sebab nilai korelasinya masih lebih tinggi dari T.

Eksperimen 3: Penajaman dan derau

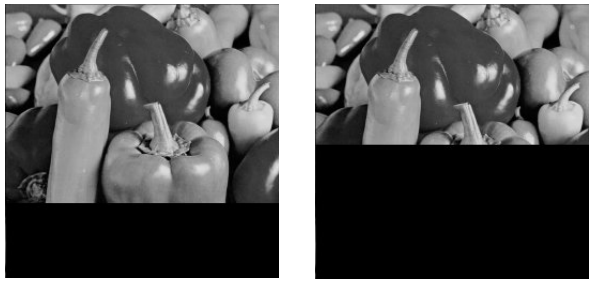
Citra ber-*watermark* dipertajam sehingga tepi-tepi di dalam citra terlihat lebih menonjol. *Watermark* masih dapat dideteksi dari citra hasil kompresi (Gambar 5). Sedangkan untuk menguji ketahanan terhadap derau, citra ditambahkan dengan derau berupa *salt and peppers* 20%. Hasilnya, *watermark* masih dapat dideteksi. Lihat Gambar 5.

Eksperimen 4: Pengubahan ukuran gambar

Citra ber-*watermark* diperkecil ukurannya hingga 75%, lalu dikembalikan lagi ke ukuran semula untuk pendeketsian (Gambar 6). *Watermark* masih dapat dideteksi (c = 0.1400). Tetapi jika pengecilan hingga 50% *watermark* tidak berhasil dideteksi (c = 0.0246). Untuk perbesaran hingga 2 kali ukuran semula, *watermark* juga masih dapat dideteksi (c = 0.2010).

Eksperimen 5: Rotasi

Citra ber-*watermark* diputar 10 derajat searah jarum jam. Untuk pendeteksian, citra yang telah diputar dikembalikan lagi ke posisi semula. Nyatanya, *watermark* masih dapat dideteksi (c = 0.1831). Lihat Gambar 7.

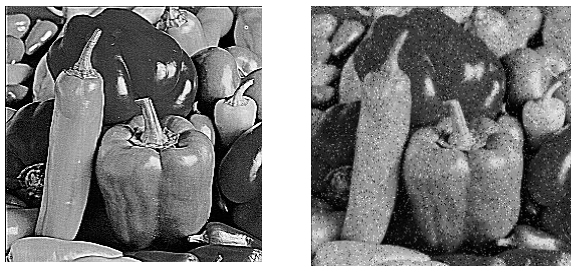


(a) c = 0.1958 (b) c = 0.1603

Gambar 3. Pemotongan sebesar (a) 25% dan (b) 50%.*Watermark* masih dapat dideteksi.

(a) Kualitas 100% c = 0.2511 (b Kualitas 60% c = 0.1489 (c) Kualitas 20% c = 0.1329,

Gambar 4. Kompresi ke format JPEG dengan berbagai kualitas kompresi. *Watermark* masih dapat dideteksi.



(b) c = 1.2613 (b) c 0.1805

Gambar 5. (a) Penajaman citra, (b) Distorsi karena derau *salt and peppers* sebesar 20%.

c = 0.1400

Gambar 6. Pengecilan ukuran citra hingga 75% dari ukuran semula. *Watermark* masih dapat dideteksi.

c = 0.1831

Gambar 7. Citra diputar 10 derajat searah jarum jam. *Watermark* masih dapat dideteksi.

7. ANALISIS *MALICIOUS ATTACK*

Serangan *malicious attack* bertujuan untuk menghapus *watermark* dari citra dengan melakukan manipulasi persamaan (3) untuk memperoleh $f(i)$:

$$f(\mathrm{i}) = f_w(\mathrm{i}) - \gamma w_e(i) \qquad (4)$$

Informasi seperti $f_w(i)$, $\mathbf{w_p}$, dan γ dimiliki oleh penyerang (tidak rahasia). Tetapi, penyerang harus mengetahui $\mathbf{w_s}$ agar dapat bisa menghitung $\mathbf{w_e}$ dengan menggunakan persamaan (2). Untuk mendapatkan $\mathbf{w_e}$, penyerang perlu mengetahui $\mathbf{w_s}$. Karena $\mathbf{w_s}$ diperoleh dari $\mathbf{w_p}$ berdasarkan permutasi rahasia *S* dan permutasi *S* juga bergantung pada nilai awal *logistic map* x_0 (yang juga rahasia), maka penyerang tidak dapat melakukan hal ini. Jika penyerang mencoba membangkitkan barisan *chaos*, maka ada tidak berhingga kemungkinan barsian *chaos* dihasilkan oleh *logistic map* dengan nilai awal antara 0 dan 1. Dengan mengingat fungsi *chaos* sensitif terhadap perubahan kecil nilai awal, maka penyerang dapat frustasi untuk menemukan nilai awal *chaos* yang tepat. Jadi, *exhaustive search* untuk menemukan barisan *chaos* menjadi tidak mungkin dilakukan. Dengan kata lain, *watermark* privat tidak mungkin diturunkan dari *watermark* publik.

8. KESIMPULAN

Di dalam makalah ini telah dipresentasikan metode *asymmetric watermarking* pada citra digital berbasis permutasi-RC4 dan fungsi *chaos*. Penyisipan dilakukan pada area *middle frequency* dari ranah *DCT* untuk memperoleh keseimbangan antara *imperceptibility* dan *robustness*. *Watermark* privat diperoleh dengan mempermutasikan *watermark* publik berdasarkan (sebagian) algoritma kriptografi *RC4*. Hasil eksperimen dengan nilai ambang T = 0.08 menunjukkan bahwa metode ini terbukti *robust* terhadap serangan *non-malicious attack* (kompresi, *cropping*, *resizing*, *sharpening*, distorsi karena derau, dan rotasi). Pengujian dengan kompresi *JPEG* dengan kualitas hingga 20%, *cropping* hingga 50%, penambahan derau *salt and peppers* 20%, penajaman berkali-kali, *resizing* hingga 75%, dan rotasi hingga 10 derajat menunjukkan bahwa nilai korelasi yang dihasilkan selalu lebih tinggi dari nilai ambang T. Metode ini juga tahan terhadap serangan malicious attack sebab penurunan *watermark* privat membutuhkan pengetahuan permutasi yang bergantung pada barisan *chaos*. Karena *chaos* sensitif terhadap perubahan kecil nilai awal, maka *exhaustive search* untuk menemukan barisan *chaos* pasti gagal.

DAFTAR REFERENSI

[1] Ingemar J. Cox, dkk, "*Secure Spread Spectrum Watermarking for Multimedi*a", IEEE Trans. On Image Processing, Vol. 6, No. 12, Dec 1997, pp.1673-1687.

[2] I. Wiseto P. Agung, *Watermarking and Content Protection for Digital Images and Video*, thesis of PhD in University of Surrey, 2002.

[3] Mauro Barni, Franco Bartolini, *Watermarking Systems Engineering*, Marcel Dekker Publishing, 2004.

[4] Joachim J. Eggers, Jonathan K. Su, and Bernd Girod, *Asymmetric Watermarking Schemes*, GMD Jahrestagung, Proceddings, Springer-Verlag, 2000.

[5] G.F Gui, L.G Jiang, C He, *General Construction of Asymmetric Watermarking Based on Permutation*, Proc. IEEE Int. Workshop VLSI Design & Video Tech., May 28, 2005.

[6] T.T. Kim, T. Kim, dan H. Choi, *Correlation-Based Asymmetric Watermarking Detector*, Int. ITCC, 2003.

[7] H. Choi, K. Lee, dan T. Kim, *Transformed-Key Asymmetric Watermarking System*, IEEE Signal Processing Letters, Vol. 11. No. 2, February 2004.

[8] Zhao Dawei, dkk, "*A Chaos-Based Robust Wavelet-Dmain Watermarking Algorithm*", Jurnal Chaos Solitons and Fractals 22 (2004) 47-54.

[9] www.yahoo.com, *Chaos Theory: A Brief Introduction*, diakses pada bulan November 2005

[10] James Lampton, *Chaos Cryptography: Protecting Data Using Chaos*, Mississippi School for Mathematics and Science.

[11] Hongxia Wang, dkk, "*Public Watermarking Based on Chaotic Map*", IEICE Trans. Fundamentals, Vol. E87-A, No. August 2004.

[12] Sangoh Jeong dan Kihyun Hong, *Dual Detection of A Watermark Embedded in the DCT Domain*, EE368A Project Report, 2001.