METODE *BLIND IMAGE-WATERMARKING* BERBASIS *CHAOS* DALAM RANAH *DISCRETE COSINE TRANSFORM* (*DCT*)

Rinaldi Munir[1], Bambang Riyanto[2], Sarwono Sutikno[3], Wiseto P. Agung [4]

Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10 Bandung
e-mail: rinaldi-m@stei.itb.ac.id[1], , briyanto@lskk.ee.itb.ac.id[2], sswarwono@ieee.org[3], wiseto@telkom.co.id[4]

ABSTRAK

Makalah ini memaparkan metode *image watermarking* berbasis *chaos* pada ranah frekuensi dengan menggunakan *Discrete Cosine Tranform* (*DCT*). Penyisipan *watermark* dilakukan secara lokal yaitu pada *subimage* yang dibentuk dari kumpulan blok berukuran 8 x 8 dan dipilih secara acak dari citra semula. Dalam hal ini, *chaotic map* digunakan untuk membangkitkan bilangan acak. Selanjutnya *subimage* ditransformasi ke dalam ranah frekuensi dengan *DCT*, dan *spread spectrum watermark* disisipkan ke dalam citra. Kelebihan metode ini adalah pendeteksian *watermark* tidak memerlukan citra semula (*blind*). Ada dua kunci yang dibutuhkan pada teknik ini, pertama nilai awal barisan *chaos* dan kedua *spread spectrum watermark*. Tidak seperti teknik *watermarking* lain yang umumnya *watermark* merupakan barisan bit acak tidak bermakna, maka pada teknik ini *watermark* adalah citra logo hitam-putih. Simulasi dengan MATLAB menunjukkan bahwa teknik ini kokoh terhadap serangan seperti kompresi *JPEG*, *cropping*, *resizing*, dan penambahan derau.

Kata kunci: *watermarking*, citra, *DCT*, *chaos*, *blind*, *subimage*, *spread spectrum watermark*.

Makalah diterima pada tanggal 7-1-2007. Revisi akhir: 12-1-2007.

1. PENDAHULUAN

Image watermarking adalah teknik untuk menyisipkan informasi yang disebut *watermark* ke dalam citra digital. *Image Watermarking* mempunyai banyak aplikasi, antara lain untuk bukti kepemilikan, otentikasi, perlindungan *copyright*, *fingerprinting*, dan *tamper proofing*. Persyaratan umum *watermarking* adalah *imperceptible*, *robustness*, dan *secure*.

Sejumlah metode *image watermarking* sudah banyak dipublikasikan dalam beberapa tahun terakhir. *Review* beberapa metode dapat ditemukan di dalam [3]. Kebanyakan metode *watermarking* didasarkan pada modulasi *spread spectrum* informasi dengan *watermark* yang berupa sinyal derau-semu (*pseudo-noise*) sebagai kunci penyisipan dan pendeteksian *watermark* [1, 2]. Istilah "*spread spectrum*" muncul karena penyisipan *watermark* ke dalam citra menggunakan teknik yang analog dengan komunikasi *spread spectrum*, yaitu *watermark* disebar di antara banyak komponen frekuensi [3]. Teknik *spread spectrum watermarking* umumnya melakukan penyisipan dan pendeteksian *watermark* dalam ranah *tranform* dengan menggunakan salah satu dari kakas transformasi yang sudah dikenal (*DCT*, *FFT*, *DWT*, dan lain-lain). Mula-mula citra ditransformasikan kedalam ranah *transform*, lalu bit *watermark* disisipkan pada koefisien transformasi tersebut. Secara umum, *watermarking* dalam ranah *transform* menghasilkan teknik yang lebih kokoh terhadap serangan seperti kompresi, *cropping*, dan operasi tapis lolos-rendah dibandingkan dengan *watermarking* dalam ranah spasial.

Makalah ini menyajikan metode *image watermarking* berbasis *chaos* yang diadaptasi dari metode yang diusulkan oleh Dawei dan Mabtoul [4, 5]. *Chaos* diterapkan karena ia mempunyai dua karakteristik penting untuk meningkatkan keamanan, yaitu sensitivitas pada kondisi awal dan sebarannya yang merata pada seluruh ruang yang ada [4]. Karakteristik ini cocok untuk enkripsi dan *watermarking*. Fungsi *chaos* digunakan untuk membangkikan barisan bilangan acak. Barisan bilangan acak di dalam metode ini digunakan untuk membentuk *subimage* yang akan dijadikan sebagai tempat penyisipan *watermark*.

Kebanyakan sistem *watermarking* yang ada hanya dapat memutuskan apakah *watermark* ada atau tidak ada di dalam citra uji berdasarkan pada prinsip korelasi, tetapi konten *watermark* itu sendiri tidak diketahui [5]. Umumnya *watermark* yang digunakan di dalam sistem tersebut adalah barisan bit acak yang tidak mempunyai makna. Di dalam makalah ini *watermark* adalah citra hitam-putih seperti logo. Selain itu, kelebihan metode ini adalah pendeteksian *watermark* tidak membutuhkan citra asal atau dikenal dengan istilah *blind watermarking*.

Ada dua kunci yang dibutuhkan untuk pendeteksian *watermark*, yaitu nilai awal barisan *chaos* dan *spread spectrum watermark*. Yang terakhir ini adalah citra biner {+1, -1} yang dibangkitkan dari *subimage* dan *watermark* asal melalui proses *thresholding* tertentu.

2. *CHAOS* DAN *WATERMARKING*

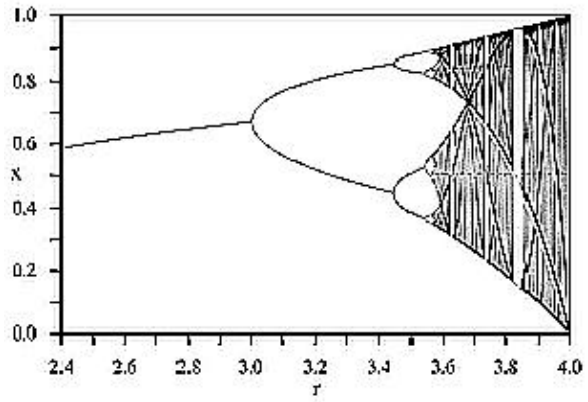
Teori *chaos* berasal dari teori sistem yang memperlihatkan kemunculan yang tidak teratur, meskipun sebenarnya teori ini digunakan untuk menjelaskan kemunculan data acak. Meskipun sistem *chaos* muncul dengan ketidakteraturan yang tinggi, tetapi ia deterministik artinya dimungkinkan membangkitkan nilai-nilai *chaos* dengan kepastian. Hal ini adalah fitur yang menjanjikan untuk komunikasi secara aman.

Karakteristik yang umum di dalam teori *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.

Salah satu fungsi *chaos* sederhana adalah persamaan logistik (*logistic map*) yang biasa dipakai di dalam ekologi untuk mensimulasikan pertumbuhan spesies di dalam ekosistem. Persamaan logistik dinyatakan sebagai

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

dengan x_0 sebagai nilai awal iterasi. Daerah asal x adalah dari 0 sampai 1. Konstanta r menyatakan laju pertumbuhan fungsi, yang dalam hal ini $0 \le r \le 4$. Konstanta r juga menyatakan bagian nirlanjar dari persamaan. Ketika r meningkat, maka kenirlanjaran sistem juga naik.



Gambar 1. Diagram *bifurcation* untuk $x_{i+1} = r\, x_i\, (1 - x_i)$

Gambar 1 memperlihatkan kelakuan fungsi yang dalam hal ini sumbu-x menyatakan nilai r sedangkan sumbu y menyatakan status sistem, yaitu nilai-nilai x. Bila $0 < r < 1$, nilai awal berapapun akan menghasilkan kepunahan. Bila $1 < r < 3$, fungsi konvergen ke sebuah nilai (*fixed-point*), yaitu nilai r yang menghasilkan sistem mempunyai periode satu siklus. Ketika $r = 3$, kurva fungsi terpecah menjadi dua (*bifurcation*) dan menghasilkan dua nilai populasi berbeda, yang berarti nilai x secara periodik berosilasi dari status tinggi ke status rendah. Periode sistem pada nilai r ini adalah dua. Ketika r meningkat lagi, kurva fungsi terpecah lagi menjadi empat, yang berarti nilai-nilai x yang dihasilkan berosilasi di antara 4 nilai. Periode sistem pada nilai r ini adalah empat.

Demikianlah seterusnya *bifurcation* menjadi lebih cepat lagi dengan meningkatnya nilai r sampai tiba pada nilai r tertentu dimana sifat *chaos* pun mucul. Pada titik ini tidak mungkin lagi memprediksi kelakuan sistem. Kita dapat melihat bahwa ketika $r > 3.75$ sistem mulai melaju dengan cepat menuju area *chaos* (di dalam Gambar 1 area tersebut diarsir hitam) [7]. Akhirnya, ketika $r = 4$, iterasi bergantung sepenuhnya pada nilai awal x_0 dan nilai-nilai yang dihasilkan muncul acak meskipun sistem ini deterministik [8]. Nilai-nilai *chaos* yang dihasilkan akan berada di dalam rentang yang lengkap antara 0 dan 1.

Beberapa tahun terakhir teori *chaos* banyak digunakan di dalam *digital watermarking*. *Chaos* digunakan khususnya sebagai pembangkit bilangan acak. Barisan nilai *chaos* digunakan langsung sebagai *watermark* [4] atau menyatakan lokasi penyisipan *watermark* di dalam citra [6]. Di dalam metode ini barisan nilai *chaos* digunakan untuk memilih secara acak blok-blok citra berukuran 8 x 8 untuk membentuk sebuah *subimage*.

2. METODE YANG DITELITI

Metode *image watermarking* berbasis *chaos* yang dipaparkan di dalam makalah ini diadaptasi dari metode yang diusulkan di dalam [4] dan [5]. Perbedaannya, di dalam metode ini penyisipan *watermark* dilakukan dalam ranah *DCT*, bukan dalam ranah *wavelet* (*DWT*). Selain itu, *watermark* adalah berupa citra biner berupa logo atau gambar bermakna lainnya (di dalam [4] *watermark* adalah barisan nilai *chaos*).

Ada empat tahapan di dalam metode yang dibahas: (1) pembentukan *subimage*, (2) pembentukan *spread spectrum watermark*, (3) penyisipan *watermark*, dan (4) pendeteksian *watermark*. Masing-masing tahapan dijelaskan di dalam upa-bab berikut.

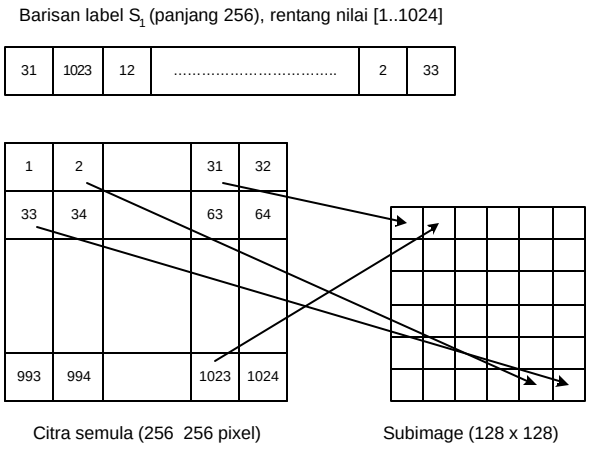
2.1 Pembentukan *Subimage*

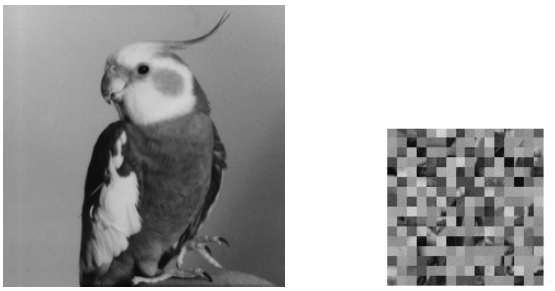
Watermark tidak disisipkan di seluruh bagian citra, tetapi hanya pada area lokal saja. Area lokal adalah berupa *subimage* yang dibentuk dari citra semula, yang langkah-langkahnya adalah sebagai berikut [4]:

1. Citra semula, I_{ori}, dibagi mejadi sejumlah blok-blok kecil berukuran 8×8 yang tidak saling beririsan, kemudian dipilih 1/4 dari total blok tersebut untuk membentuk sebuah *subimage* baru. Blok-blok yang dipilih ditentukan dengan fungsi *chaos*. Fungsi *chaos* yang digunakan adalah *logistic map*. Misalkan I_{ori} semula berukuran 256 x 256 *pixel*, maka pembagian I_{ori} menghasilkan 1024 buah blok 8×8.

2. Beri label blok-blok tersebut dari 1 sampai 1024. Dengan menggunakan *logistic map* dan nilai awal x_0, bangkitkan barisan *chaos* S_i yang berisi 1024 elemen, lalu kalikan setiap elemen dengan 1024 sehingga diperoleh barisan baru, S_n, yang rentangnya [1, 1024].
3. Selanjutnya, pilih 256 buah (yaitu 1/4 dari 1024) nilai berbeda pertama dari S_n sebagai barisan baru S_1 dan bentuklah *subimage* I_{sub} dari 256 blok berdasarkan label di dalam S_1 (kita harus menjamin bahwa di dalam 1024 elemen barisan *chaos* terdapat 256 elemen yang berbeda).

Gambar 2 memperlihatkan proses pemilihan blok citra untuk membentuk *subimage* dari citra asal berukuran 256 $\times$256.





Gambar 2. Pembentukan *subimage*

2.2 Pembentukan *Spread Spectrum Watermark*

Watermark W yang berupa citra biner bersama-sama dengan *subimage* di atas digunakan untuk membentuk *spread spectrum watermark*. *Pixel-pixel W* yang hanya bernilai 0 dan 1 terlebih dahulu diubah menjadi bernilai –1 dan 1 dengan cara menjadikan 0 sebagai –1 sedangkan 1 tetap adanya. Langkah-langkah pembentukan *spread spectrum watermark* adalah sebagai berikut [5]:

1. Transformasikan *subimage* I_{sub} ke dalam ranah frekuensi dengan menggunakan *DCT*. Koefisien *DCT* dari I_{sub} dinyatakan sebagai $\tilde{I}$. Tranformasi DCT terhadap citra I_{sub} (i, j) yang berukuran $N \times N$ dihitung dengan menggunakan rumus:

$$\tilde{I}(u,v) = C(u)c(v)\frac{2}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I_{sub}(i,j)\cos\left(\frac{p}{N}u(i+\frac{1}{2})\right)\cos\left(\frac{p}{N}v(j+\frac{1}{2})\right) \quad (2)$$

dengan

$$\begin{cases} c(w) = 2^{-\frac{1}{2}} & w = 0 \\ c(w) = 1 & w > 0 \end{cases}$$

2. Bangkitkan *spread spectrum watermark* W_k sebagai berikut: untuk tiap elemen (i, j) dari $\tilde{I}$, nilainya dibandingkan dengan nilai dari delapan tetangganya. Misalkan t menyatakan jumlah elemen tetangga yang nilainya lebih kecil dari nilai elemen (i, j). *Spread spectrum watermark* W_k dihitung dengan rumus:

$$W_k(i,j) = \begin{cases} 1 & ,(t \geq 2 dan W(i,j) = 1) atau \\ & (t < 2 dan W(i,j) = -1) \\ -1 & ,lainnya \end{cases} \quad (3)$$

2.3 Penyisipan *Watermark*

1. *Spread spectrum watermark* W_k disisipkan ke dalam $\tilde{I}$ dengan persamaan [5]:

$$\hat{I}(i,j) = \tilde{I}(i,j) + \alpha \cdot W_k(i,j) \cdot \left|\tilde{I}(i,j)\right| \quad (4)$$

yang dalam hal ini $\hat{I}$ menyatakan koefisien *DCT* ter-*watermark*, $\tilde{I}$ menyatakan koefisien *DCT* semula, W_k menyatakan *spread spectrum watermark*, dan α adalah parameter yang menyatakan kekuatan *watermark* dan dipilih sedemikian rupa untuk menyeimbangkan antara *invisibility* dan *robustness*.
2. Lakukan *IDCT* (*Inverse Discrete Cosine Transform*) terhadap $\hat{I}$ untuk menghasilkan *subimage* yang sudah ber-*watermark*. Transformasi *IDCT* adalah:

$$I_{sub}(i,j) = \frac{2}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} c(u)c(v)\hat{I}(u,v)\cos\left(\frac{p}{N}u(i+\frac{1}{2})\right)\cos\left(\frac{p}{N}v(j+\frac{1}{2})\right) \quad (5)$$

3. Akhirnya, berdasarkan label S_1, setiap blok kecil dari *submage* ber-*watermark* ditempatkan kembali pada citra asal pada posisi semula untuk memperoleh citra ber-*watermark*.

2.4 Pendeteksian Watermark

Pendeteksian *watermark* di sini bertujuan mengekstraksi *watermark* dari citra uji. Pendeteksian *watermark* tidak membutuhkan citra asal. Kunci yang dibutuhkan adalah nilai awal barisan *chaos* (x_0) dan *spread spectrum watermark* (W_k). Berikut langkah-langkah pendeteksian *watermark*.

1. Bentuklah *subimage* I_{sub} dari citra yang diuji. Pembentukan *subimage* ini memerluakan kunci x_0.
2. Lakukan transformasi *DCT* terhadap I_{sub} tersebut. Misalkan koefisien hasil transormasi disimpan di dalam matriks $\hat{I}$.
3. Untuk tiap elemen (*i, j*) dari $\hat{I}$, nilainya dibandingkan dengan nilai dari delapan tetangganya. Misalkan *t* menyatakan jumlah elemen tetangga yang nilainya lebih kecil dari nilai elemen (*i, j*). *Watermark W* dikonsruksi dengan rumus:

$$W(i,j) = \begin{cases} 1 & ,(t \geq 2 dan W_k(i,j) = 1) atau \\ & (t < 2 dan W_k(i,j) = -1) \\ -1 & ,lainnya \end{cases} \quad (4)$$

3. HASIL EKSPERIMEN DAN ANALISIS

Metode *image watermarking* berbasis *chaos* yang sudah dijelaskan di atas diprogram dengan MATLAB 7, selanjutnya citra hasil *watermarking* diuji dengan beberapa serangan. Serangan yang umum dilakukan terhadap citra ber-watermark sebenarnya adalah operasi pengolahan citra yang umum dilakukan seperti kompresi JPEG, penambahan derau, *resize*, dan *cropping*. Citra uji yang digunakan adalah citra *greyscale* 'bird' yang dengan format *bitmap* dan berukuran 256 × 256, sedangkan *watermak* yang disisipkan adalah citra 'ganesha' yang bertipe biner dan berukuran 128 × 128 (Gambar 3). Fungsi *chaos* yang digunakan adalah *logistic map* dengan r = 4.0 dan nilai awal *chaos* (kunci pertama) adalah 0.647. Parameter α yang dipilih untuk penyisipan *watermark* adalah 0.01.

Citra 'bird' (256 × 256)

Watermark (128 x 128)

Gambar 3. Citra asal dan *watermark* yang digunakan dalam pengujian

Gambar 4 memperlihatkan kasus tidak ada serangan yang dilakukan terhadap citra ber-*watermark*. Citra ber-*watermark* hampir tidak dapat dibedakan dengan citra asalnya. *Watermark* yang diekstraksi dari citra tersebut memang tidak tepat sama dengan *watermark* asal. Hal ini disebabkan adanya operasi pemotongan bilangan riil menjadi bilangan bulat sebagai akibat transformasi *IDCT* terhadap *subimage*.

Citra ber-*watermark*

Watermark ekstraksi

Gambar 4. Citra ber-*watermark* dan *watermark* ekstraksi

3.1 Pengaruh Perubahan Nilai awal *Chaos*

Karena *chaos* peka terhadap nilai awal, maka pengubahan sedikit saja pada nilai awal x_0 menghasilkan kesalahan pada saat pendeteksian *watermark*. Gambar 5 memperlihatkan *watermark* asli dan *watermark* hasil ektstraksi bila nilai awal x_0 yang digunakan pada pendeteksian diubah sedikit dari 0.647 menjadi 0.647001.

Watermark asli

Watermark salah yang diekstraksi

Gambar 5. *Watermark* asli dan *watermark* salah yang diekstraksi dengan x_0 yang diubah sedikit

3.2 Kekokohan Terhadap Kompresi *JPEG*

Untuk melihat kekokohan (*robustness*) *watermark* terhadap pengaruh kompresi (*noise*), maka di dalam eksperimen ini digunakan program *Jasc PaintShopPro* untuk melakukan konversi format citra ber-*watermark* dari *bitmap* ke *jpeg*. Selanjutnya, citra dalam format *jpeg* dikembalikanlagi ke format *bitmap* untuk digunakan pada waktu pendeteksian *watermark*. Hasil pendeteksian memperlihatkan bahhwa kompresi *JPEG* hanya sedikit merusak *watermark* (Gambar 6). *Watermark* masih dapat dikenali dengan baik.

Citra ber-*watermark* dalam format JPEG

Watermark ekstraksi

Gambar 6. Pengujian kompresi *JPEG* terhadap citra ber-*watermark*

3.3 Kekokohan Terhadap Penambahan Derau

Program *Jasc PaintShopPro* kembali digunakan untuk menambahkan derau sebesar 5% pada citra ber-*watermark*. Hasil pendeteksian memperlihatkan bahwa *watermark* yang diekstraksi memang mengalami kerusakan tetapi masih dapat dikenali (Gambar 7).

Citra ber-*watermark* yang telah ditambah derau sebesar 5%

Watermark ekstraksi

Gambar 7. Pengujian penambahan derau 5% terhadap citra ber-*watermark*

3.4 Kekokohan Terhadap *Resize*

Citra ber-*watermark* (256 × 256) diperkecil menjadi setengah kali ukuran semula (128 × 128 mengunakan *Jasc PaintShop Pro*. Untuk mendeteksi *watermark*, citra yang sudah diperkecil tadi dikembalikan lagi ke ukuran semula. Hasil pendeteksian memperlihatkan bahwa *watermark* yang diekstrraksi masih dapat dikenali (Gambar 8).

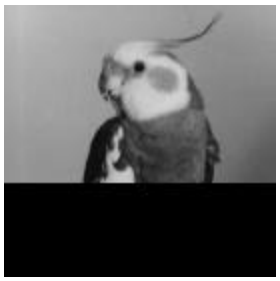
Citra ber-*watermark* yang telah diperkecil menjadi 50%

Watermark ekstraksi

Gambar 8. Pengujian *resize* sebesar 50% terhadap citra ber-*watermark*

3.5 Kekokohan Terhadap *Cropping*

Operasi *cropping* pada pengolahan citra umumnya bertujuan untuk mengambil bagian tertentu dari gambar. Pada pengujian ini, citra ber-*watermark* dipotong sekitar 30% pada bagian bawah. Bagian yang dipotong diisi dengan *pixel-pixel* yang berwarna hitam. Hasil pendeteksian menunjukkan bahwa *watermark* yang diekstraksi masih dapat dikenali dengan baik (Gambar 9).



dipotong ber-*watermark* yang telah dipotong sebesar 30%

Watermark ekstraksi

Gambar 9. Pengujian *cropping* sebesar 30% terhadap citra ber-*watermark*

4. KESIMPULAN

Di dalam makalah ini telah disajikan metode *image watermarking* berbasis *chaos* dalam ranah *DCT*. Hasil pengujian menunjukkan bahwa citra ber-*watermark* tidak dapat dibedakan dengan citra asalnya (syarat *invisibility* terpenuhi). Pengujian dengan bermacam-macam serangan terhadap citra ber-*watermark* menunjukkan bahwa metode *watermarking* yang dikembangkan ini kokoh terhadap serangan seperti kompresi *JPEG*, *cropping*, *resizing*, dan penambahan derau (syarat *robustness* terpenuhi). Kelebihan lainnya, pendeteksian *watermark* tidak membutuhkan citra asal sehingga dinamakan *blind watermarking*. Penggunaan *chaos* dalam *watermarking* bertujuan untuk meningkatkan keamanan metode sehingga metode tetap aman terhadap perubahan kecil pada nilai awal (syarat *secure* terpenuji).

Kelemaham metode ini terletak pada ukuran citra yang harus merupakan perpangkatan dari 2. Untuk citra yang ukurannya bukan perpangkatan dari 2 sebenarnya masih dapat dilakukan dengan terlebih dahulu menambah pixel-pixel semu sehingga ukuran citra menjadi perpangkatan dari 2. Pengembangan lebih lanjut dapat dilakukan terhadap citra berwarna maupun video.

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] Frank Hartung, Bern Girod, "*Fast Public-Key Watermarking of Compressed Video*", Proceeding of International Conference on Image Processing 1997.
[3] Saraju P. Mohanty, "*Digital Watermarking: A Tutorial Review*", Dept. of Computer Scieence and Engineering, University of South Florida.
[4] Zhao Dawei, dkk, *"A Chaos-Based Robust Wavelet-Dmain Watermarking Algorithm"*, Jurnal Chaos Solitons and Fractals 22 (2004) 47-54.
[5] S. Mabtoul, dkk, *"A Blind Chaos-Based Complex Wavelet-Domain Image Watermarking Technique"*, International Journal of Computer Science and Network Security, Vol. 6 No.3, March 2006.
[6] Hongxia Wang, dkk, "*Public Watermarking Based on Chaotic Map"*, IEICE Trans. Fundamentals, Vol. E87-A, No. August 2004.
[7] James Lampton, "*Chaos Cryptography: Protecting Data Using Chaos*", Mississippi School for Mathematics and Science.
[8] R. Clarck Robinson, *An Introduction to Dynamical Systems, Continuous and Discrete*, Pearson Prentice Hall, 2004.