Modifikasi *Spread Spectrum Watermarking* dari Cox Berbasiskan pada Enkripsi *Chaotic*

Rinaldi Munir, Bambang Riyanto, Sarwono Sutikno, Wiseto P. Agung
Sekolah Teknik Elekto dan Informatika, ITB, Bandung 40132
e-mail: rinaldi-m@stei.itb.ac.id

Abstract*: Makalah ini memaparkan modifikasi algoritma* spread spectrum watermarking *yang diusulkan oleh Cox dengan penambahan fitur enkripsi pada* watermark. *Selain itu, tidak seperti metode Cox yang* watermark*-nya berupa barisan nilai acak yang tidak bermakna, maka pada modifikasi ini* watermark *yang digunakan adalah citra logo hitam-putih. Penyisipan dan pendeteksian watermark dilakukan pada ranah Discrete Cosine Tranform (DCT). Untuk meningkatkan keamanan algoritma* watermarking, watermark *dienkripsi terlebih dahulu*. Chaotic map *dioperasikan dua kali, pertama untuk membangkitkan barisan bit yang digunakan untuk mengenkripsi watermark, kedua untuk menghasilkan barisan nilai acak yang dimodulasikan dengan* watermark. *Simulasi dengan MATLAB menunjukkan bahwa teknik ini kokoh terhadap serangan seperti kompresi JPEG, tetapi kurang kokoh terhadap* cropping, resizing, *dan penambahan derau.*

Kata Kunci*: spread spectrum, watermarking,Cox, enkripsi, chaotic map. .*

1. PENDAHULUAN

Representasi digital dari produk multimedia seperti berkas audio, citra, dan video menjadi populer karena data digital mudah untuk digandakan (*copy*) dan didistribusikan. Namun, penggandaan dan pendistribusian yang tidak berizin menimbulkan masalah terhadap hak atas kekayaaan intelektual (HAKI). Permasalahan ini dapat diatasi dengan menggunakan *digital watermarking*. *Digital watermarking* adalah teknik untuk menyisipkan informasi yang menyatakan label kepemilikan (disebut *watermark*) ke dalam data digital. *Digital watermarking* mempunyai banyak aplikasi, antara lain untuk bukti kepemilikan, otentikasi, perlindungan *copyright*, *fingerprinting*, dan *tamper proofing*. Persyaratan umum *watermarking* adalah: 1) *imperceptible*: *watermark* tidak dapat dipersepsi secara inderawi, 2) *robustness*: kokoh terhadap serangan yang dapat merusak atau menghapus *watermark*, 3) *secure*: hanya pihak yang mempunyai otoritas yang dapat mengakses *watermark*.

Penyisipan *watermark* ke dalam citra digital umumnya dilakukan dalam dua ranah, yaitu ranah spasial dan ranah *transform*. Penyisipan dalam ranah spasial menyisipkan *watermark* secara langsung ke dalam *pixel* citra. Keuntungan cara ini adalah murah (cepat) tetapi umumnya *watermark* tidak kokoh terhadap manipulasi yang dilakukan kepada citra.

Kekokohan *watermark* dapat diperoleh jika penyisipan *watermark* dilakukan dalam ranah *transform*, artinya *watermark* disisipkan ke dalam koefisien transformasi. Kakas transformasi yang umum digunakan adalah *DFT* (*Discrete Fourier Transform*), *DCT* (*Discrete Cosine Transform*), dan *DWT* (*Discrete Wavelet Transform*). Kekokohan terhadap manipulasi *cropping* dapat diperoleh jika *watermark* disebar (*spread*) di antara seluruh komponen frekuensi. Kekokohan terhadap operasi geometri (seperti penskalaan, rotasi, atau pergeseran) dapat diperoleh dalam ranah *tranform* karena ranah *transform* dapat dirancang sedemikian sehingga *invariant* terhadap sekumpulan transformasi tertentu [1]. Misalnya, metode *watermarking* yang menggunakan tranformasi *DFT* kokoh terhadap operasi pergeseran karena pergeseran dalam ranah spasial tidak mempunyai pengaruh terhadap magnitudo *DFT* .

Kebanyakan metode *image watermarking* di dalam ranah *transform* menggunakan teknik *spread spectrum*. Istilah "*spread spectrum*" muncul karena penyisipan *watermark* ke dalam citra menggunakan teknik yang analog dengan komunikasi *spread spectrum*, yaitu *watermark* disebar (*spread*) di antara banyak komponen frekuensi.

Gagasan *spread spectrum watermarking* pertama kali diperkenalkan oleh Cox *et al* [2]. Pada metode Cox, *watermark* disebar ke dalam sekumpulan komponen frekuensi yang signifikan secara persepsi (*perceptually significant region*). Penyerang yang mencoba untuk menghapus *watermark* dari citra harus berhadapan dengan komponen tersebut, sebab menghapus *watermark* dapat menyebabkan kerusakan yang terlihat pada citra. Menyisipkan *watermark* ke dalam komponen frekuensi tersebut dapat mendistorsi kualitas citra itu sendiri, oleh karena itu harus dipilih parameter yang menyeimbangkan antara *robustness* dengan *invisibility*.

Watermark yang digunakan oleh Cox adalah barisan

nilai yang dipandang sebagai sinyal derau-semu (*pseudo-noise*) dan dimodulasi dengan koefisien *spread spectrum*. *Watermark* ini juga sekaligus berlaku sebagai kunci penyisipan dan pendeteksian *watermark*.
Pada beberapa sistem *watermarking*, *watermark* tidak selalu berupa barisan nilai acak yang tidak memiliki persepsi apa-apa tetapi diinginkan *watermark* berupa gambar/logo atau informasi yang menunjukkan identitas pemilik citra. Gambar banyak dipilih sebagai *watermark* karena mudah dipersepsi secara visual.

Makalah ini menyajikan metode *image watermarking* berbasis *chaos* yang diadaptasi dari metode yang diusulkan oleh Cox. *Chaos* diterapkan karena ia memiliki dua karakteristik penting untuk meningkatkan keamanan, yaitu sensitivitas pada kondisi awal dan sebarannya yang merata pada seluruh ruang yang ada [3]. Karakteristik ini cocok untuk enkripsi dan *watermarking*. Fungsi *chaos* di dalam *watermarking* digunakan untuk membangkikan barisan bilangan acak. Barisan bilangan acak di dalam metode ini digunakan untuk mengenkripsi *watermark* dan memodulasi *watermark* menjadi barisan bilangan riil sehingga masih tetap relevan disisipkan dengan metode Cox. *Watermark* dienkripsi sebelum disisipkan untuk memperoleh *watermarking* yang lebih aman.

2. ALGORITMA *SPREAD SPECTRUM*

Watermark (*w*) yang disisipkan oleh Cox adalah barisan bilangan riil yang mempunyai distribusi normal atau Gaussian, $N(0, \sigma^2)$, yang dalam hal ini distribusi normal mempunyai rerata 0 dan variansi σ^2. *Watermark* berdistribusi Gaussian dipilih karena ia lebih kokoh terhadap perubahan dibandingkan dengan menggunakan distribusi *uniform* [4]. *Watermark* juga berlaku sebagai kunci, karena hanya pemilik citra yang mengetahui *watermark* ini. *Watermark* harus disisipkan di dalam komponen sinyal yang signifikan secara persepsi (*perceptually significant region*), meskipun perubahan pada komponen ini dapat menyebabkan kerusakan yang tampak pada citra.

2.1 Penyisipan *Watermark*

Langkah-langkah penyisipan *watermark* ke dalam citra digital adalah sebagai berikut:

1. Citra asli dianggap sebagai sebuah blok, lalu ditansformasi ke dalam ranah frekuensi dengan menggunakan *DCT*. Tranformasi citra $I(m, n)$ yang berukuran $N \times M$ dihitung dengan menggunakan rumus:

$$C(p,q) = \alpha_p \alpha_q \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I(m,n) \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \quad (1)$$

yang dalam hal ini,

$$0 \le p \le M-1;\ 0 \le q \le N-1 \quad ;$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} ;$$

$$\alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

2. Temukan komponen frekuensi yang signifikan secara persepsi. Cox menggunakan 1000 koefisien terbesar. Inilah yang dinamakan *frequency spreading*.
3. *Watermark* $W = w_1, w_2, \ldots, w_n$ dibangkitkan sedemikian sehingga w_i mempunyai distribusi $N(0, 1)$, yaitu distribusi normal dengan rerata 0 dan variansi 1.*Watermark* disisipkan ke dalam koefisien *DCT* dengan cara mengubah komponen frekuensi v_i dari citra asal menjadi $\hat{v}_i$ dengan menggunakan persamaan:

$$\hat{v}_i = v_i(1 + \alpha w_i) \quad (2)$$

yang dalam hal ini α adalah faktor skalar. Cox memilih $\alpha = 0.1$.
4. Lakukan transformasi *DCT* inversi terhadap hasil langkah 4 untuk menghasilkan citra ber-*watermark*. Persamaan *DCT* inversi adalah:

$$I(m,n) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \alpha_p \alpha_q C(p,q) \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \quad (3)$$

2.2 Ekstraksi *Watermark*

Proses ekstraksi *watermark* membutuhkan citra asal, citra ber-*watermark*, dan *watermark* asal untuk pembandingan. Langkah-langkah ekstraksi *watermark* adalah sebagai berikut:

1. Citra ber-*watermark* dianggap sebagai sebuah blok, lalu ditansformasi ke dalam ranah frekuensi dengan menggunakan *DCT*.
2. Lakukan tranformasi *DCT* terhadap citra asli (yang belum diberi *watermark*).
3. Selisih langkah 1 dan 2 adalah *watermark* yang diekstraksi, *W** , dengan kata lain *W** diperoleh dengan menghitung w_i* kembali berdasarkan persamaan 2 menghasilkan persamaan berikut:

$$w_i^* = \frac{\frac{\hat{v}_i}{v_i} - 1}{\alpha} \quad (4)$$

4. *Watermark* *W** dibandingkan dengan *watermark* asli *W* dengan menggunakan korelasi berikut:

$$sim(W, W^*) = \frac{W \cdot W^*}{\sqrt{W \cdot W^*}} \quad (5)$$

Untuk memutuskan apakah W dan W^* sama, digunakan nilai ambang T, yaitu keduanya sama jika $sim(W, W^*) > T$.

Kelemahan metode Cox adalah dibutuhkannya citra asal untuk proses ekstraksi, sehingga metode ini tergolong ke dalam kelompok *non-blind watermarking*. Meskipun demikian, menurut penulisnya, *watermark* kokoh terhadap operasi pengolahan citra yang umum seperti konversi analog-ke-digital dan digital-ke-analog, *dithering*, *resampling*, kompresi, rotasi, translasi, dan pensakalaan [1, 4].

3. *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

$$x_{i+1} = r\, x_i\, (1 - x_i) \quad (6)$$

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. Ketika $r = 4$, iterasi bergantung sepenuhnya pada nilai awal x_0 dan nilai-nilai yang dihasilkan muncul acak meskipun sistem ini deterministik [5]. Nilai-nilai *chaos* yang dihasilkan bertipe bilangan riil dan 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* [3] atau menyatakan lokasi penyisipan *watermark* di dalam citra [6]. Di dalam metode ini barisan nilai *chaos* digunakan untuk mengenkripsi *watermark* sebelum disisipkan dan memodulasi *watermark* yang berupa citra biner menjadi barisan nilai riil acak yang siap disisipkan ke dalam koefisien hasil transformasi.

4. METODE YANG DIUSULKAN

Metode *image watermarking* berbasiskan pada enkripsi *chaos* yang dipaparkan di dalam makalah ini diadaptasi sepenuhnya dari metode Cox. Perbedaannya, di dalam metode ini watermark dienkripsi sebelum disisipkan dengan barisan nilai *chaos*. Selain itu, *watermark* adalah berupa citra biner berupa logo atau gambar bermakna lainnya (di dalam [1] *watermark* adalah barisan nilai acak yang mempunyai distribusi Gaussian). Fungsi *chaos* yang digunakan adalah persamaan logistik seperti yang telah dijelaskan sebelumnya.

Untuk mengenkripsi *watermark* (**w**) yang berupa citra biner, maka persamaan logistik dengan nilai awal x_0 diterapkan untuk membangkitkan barisan nilai *chaos* sebanyak elemen *pixel* citra biner. Jika citra biner berukuran 32×32 *pixel*, maka panjang barisan nilai *chaos* adalah $64 \times 64 = 1024$. Karena persamaan logistik menghasilkan nilai-nilai bertipe riil (x_i), maka nilai-nilai riil ini ditransformasi ke nilai biner dengan fungsi pengambangan berikut:

$$g(x_i) = \begin{cases} 0, & x_i \le t \\ 1, & x_i > t \end{cases} \quad (7)$$

yang dalam hal ini nilai ambang t dapat diambil dari nilai rata-rata seluruh barisan *chaos* yang telah dibangkitkan.

Misalkan **b** adalah barisan nilai *chaos* hasil persamaan (7), maka *watermark* **w** dienkripsi dengan cara meng-*xor*-kannya dengan **b**:

$$\mathbf{w'} = \mathbf{w} \oplus \mathbf{b} \quad (8)$$

Selanjutnya, *watermark* **w'** yang elemen-elemennya terdiri dari 0 dan 1 dikonversi menjadi nilai bipolar (-1 +1) dengan cara mengkonversi setiap 0 menjadi –1 sedangkan bit 1 tetap.

Persamaan logistik dengan nilai awal y_0 (sebaiknya $x_0 \ne y_0$) kembali diterapkan untuk menghasilkan barisan nilai *chaos* kedua, **y**. Barisan nilai *chaos* yang bertipe bilangan riil ini kemudian dikalikan dengan *watermark* **w'**:

$$\mathbf{w''} = \mathbf{w'} \cdot \mathbf{y} \quad (9)$$

Watermark **w''** inilah yang dikalikan dengan persamaan (2) di atas untuk memperoleh koefisien frekuensi ter-*watermark*. Langkah-langkah selanjutnya sama seperti pada metode Cox yang asli.

Proses ekstraksi watermark Untuk merekonstruksi *watermark* pada proses ekstraksi, *watermark* dari hasil persamaan (4) di-*xor*-kan kembali dengan **y** untuk memperoleh *watermark* terekstraksi:

$$\mathbf{w^{**} = w^{*} \oplus y} \qquad (10)$$

Jadi, pada proses ekstraksi *watermark* hanya dibutuhkan satu barisan *chaos* saja. Selanjutnya **w**** dibandingkan dengan **w** untuk verifikasi.

5. HASIL EKSPERIMEN DAN PEMBAHASAN

Modifikasi metode *spread spectrum watermarking* yang berbasiskan pada enkripsi *chaotic* 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* 'lada' dengan format *bitmap* dan berukuran 256 × 256, sedangkan *watermak* yang disisipkan adalah citra 'alpha' yang bertipe biner dan berukuran 64 × 64 (Gambar 2a). Fungsi *chaos* yang digunakan adalah *logistic map* dengan r = 4.0, nilai awal barisan *chaos* (kunci pertama) adalah x_0 = 0.2785, dan nilai awal barisan chaos kedua adalah y_0 = 0.4368. Parameter α yang dipilih untuk penyisipan *watermark* adalah 0.1.

Citra 'lada' (256 × 256)

Watermark(64 x 64)

(a)

Citra ber-*watermark*

Watermark ekstraksi

(b)

Gambar 2. Penyisipan dan pengekstraksian *watermark*

Gambar 2(b) 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 sama dengan *watermark* asal.

5.1 Pengaruh Perubahan Nilai Awal *Chaos*

Karena *chaos* peka terhadap nilai awal, maka pengubahan sedikit saja pada nilai awal menghasilkan kesalahan yang signifikan pada saat pengekstraksian

Watermark asli yang

Watermark salah diekstraksi

Gambar 3. *Watermark* asli dan *watermark* salah yang diekstraksi dengan x_0 yang diubah sedikit saja.

watermark. Gambar 3 memperlihatkan *watermark* asli dan *watermark* hasil ektstraksi bila nilai awal x_0 yang digunakan pada pendeteksian diubah sedikit dari 0.2785 menjadi 0.2785001. Terlihat *watermark* hasil ekstraksi berupa pola acak yang tidak bermakna.

5.2 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* dengan *quality* 100%. Selanjutnya, citra dalam format *jpeg* dikembalikanlagi ke format *bitmap* untuk digunakan pada waktu pendeteksian *watermark*. Hasil pendeteksian memperlihatkan bahwa kompresi *JPEG* memang merusak *watermark* (Gambar 4), tetapi *watermark* tersebut masih dapat dikenali dengan baik.

Citra ber-*watermark* dalam format JPEG

Watermark ekstraksi

Gambar 4. Pengujian kompresi *JPEG* terhadap citra ber-

Citra ber-*watermark* yang telah ditambah derau sebesar 1%

Watermark ekstraksi

Gambar 5. Pengujian penambahan derau *salt and peppers* 1% terhadap citra ber-*watermark*

watermark

5.2 Kekokohan Terhadap Penambahan Derau

Program *Jasc PaintShopPro* kembali digunakan untuk menambahkan derau (*salt and peppers*) sebesar 1% pada citra ber-*watermark*. Hasil pendeteksian memperlihatkan bahwa *watermark* yang diekstraksi memang mengalami kerusakan parah tetapi masih dapat dikenali (Gambar 5).

5.3 Kekokohan Terhadap *Resize*

Citra ber-*watermark* (256 × 256) diperkecil menjadi 80% dari ukuran semula (205 × 205) mengunakan *Jasc PaintShop Pro*. Untuk mendeteksi *watermark*, citra yang sudah diperkecil tadi dikembalikan lagi ke ukuran semula. Hasil pendeteksian memperlihatkan bahwa *watermark* mengalami kerusakan tetapi sebagian masih dapat dikenali (Gambar 6). Eksperimen dengan pengecilan hingga 50% membuat *watermark* hasil ekstraksi lebih sulit dikenali.

3.5 Kekokohan Terhadap *Cropping*

Operasi *cropping* pada pengolahan citra umumnya

Citra ber-*watermark* yang telah diperkecil menjadi 80%

Watermark ekstraksi

Gambar 6. Pengujian *resize* sebesar 50% terhadap citra ber-*watermark*

bertujuan untuk mengambil bagian tertentu dari gambar. Pada pengujian ini, citra ber-*watermark* dipotong sekitar 15% pada bagian bawah. Bagian yang dipotong diisi dengan *pixel-pixel* yang berwarna

Citra ber-*watermark* yang telah dipotong 15%

Watermark ekstraksi

Gambar 7. Pengujian *cropping* sebesar 30% terhadap citra ber-*watermark*

hitam. Hasil pendeteksian menunjukkan bahwa *watermark* yang diekstraksi mengaklami kerusakan parah dan sangat sulit dikenali (Gambar 7).

4. KESIMPULAN

Di dalam makalah ini telah disajikan modifikasi metode *spread spectrum watermarking* dari Cox dengan penambahan fitur enkripsi. *Watermark* dienkripsi dengan barisan nilai *chaos* sebelum disisipkan ke dalam citra. Selain itu, *watermark* adalah berupa citra biner, berbeda dengan metode Cox yang asli dimana *watermark* adalah barisan nilai acak yang tidak memiliki persepsi apapun. Hasil pengujian menunjukkan bahwa citra ber-*watermark* tidak dapat dibedakan dengan citra asalnya. Pengujian dengan bermacam-macam serangan terhadap citra ber-*watermark* menunjukkan bahwa metode *watermarking* yang dikembangkan ini kokoh terhadap serangan seperti kompresi *JPEG*, namun kurang kokoh terhadap *cropping*, *resizing*, dan penambahan derau. Kesimpulan ini agak berbeda dengan kesimpulan Cox karena pengukuran *robustness* pada metode Cox menggunakan ukuran korelasi berdasarkan nilai ambang tertentu, sedangkan pengukuran *robustness* di dalam makalah ini berdasarkan pengamatan visual semata. Penggunaan *chaos* untuk mengenkripsi *watermark* bertujuan untuk meningkatkan keamanan metode sehingga metode tetap aman terhadap perubahan kecil pada nilai awal.

Kelemahan metode ini adalah diperlukannya citra asal untuk melakukan ekstraksi *watermark*.

REFERENSI

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

[2] 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.

[3] Zhao Dawei, dkk, "*A Chaos-Based Robust*

Wavelet-Dmain Watermarking Algorithm", Jurnal Chaos Solitons and Fractals 22 (2004) 47-54.

[4] Saraju P. Mohanty, "*Digital Watermarking: A Tutorial Review*", Dept. of Computer Scieence and Engineering, University of South Florida.

[5] R. Clarck Robinson, *An Introduction to Dynamical Systems, Continuous and Discrete*, Pearson Prentice Hall, 2004.

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