METODE *ASYMETRIC WATERMARKING* DENGAN PENJUMLAHAN *CHAOS* DALAM RANAH *DCT*

Rinaldi Munir, Bambang Riyanto, Sarwono Sutikno, Wiseto P. Agung

Sekolah Teknik Elektro dan Informatika ITB

E-mail: rinaldi@informatika.org

Abstrak

Asymmetric watermarking *menggunakan kunci yang berbeda untuk menyisipkan dan mendeteksi watermark. Di dalam makalah ini disajikan metode* asymmetric watermarking *berbasis* chaos pada citra digital*. Sifat* chaos *yang peka terhadap perubahan kecil nilai awal cocok untuk meningkatkan keamanan* watermark*. Dalam hal ini, kunci privat adalah* watermark *privat berupa barisan nilai riil yang berdistribusi normal N(0, 1), sedangkan kunci publik adalah* watermark *publik yang diperoleh dengan menjumlahkan* watermark *privat dengan sebuah barsian* chaos*.* Watermark *privat 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 *dan* malicious attack

Kata Kunci*: asymmtric watermarking, chaos, DCT, korelasi, robust.*

1. PENDAHULUAN

Data multimedia seperti citra digital, audio, video, dan lain-lain mudah ditransmisikan, diubah, dan digandakan dengan kualitas yang sama dengan data asal. Masalah yang muncul dari distribusi dan penggandaan ilegal adalah *copyright*. *Digital watermarking* merupakan teknik yang digunakan untuk mengontrol penggandaan dan distribusi data multrimedia [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 otorita yang dapat melakukanya.

Sejumlah skema *watermarking* sudah banyak dipublikasikan dalam beberapa tahun terakhir. Tetapi, satu masalah di dalam *state-of-the-art* skema *watermarking* tersebut adalah kebanyakan skema itu simetri, yaitu 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.

Masalah ini dapat diselesaikan dengan menggunakan skema *asymmetric watermarking*. Pada skema ini digunakan kunci (*watermark*) yang berbeda untuk penyisipan dan pendeteksian. Skema *asymmetric watermarking* disebut *public-key watermarking* jika *watermark* yang digunakan untuk

pendeteksian dipublikasikan, maka kunci tersebut dinamakan kunci (*watermark*) publik. Kunci (*watermark*) yang digunakan pada proses penyisipan dirahasiakan sehingga dinamakan kunci (*watermark*) privat). Skema *public-key watermarking* ini 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].

Secara umum, di dalam skema *asymmetric watermarking*, deteksi *watermark* biasanya direalisasikan dengan uji korelasi antara *watermark* publik dengan data multimedia yang diterima [6]. Hasil pendeteksian adalah keputusan biner yang mengindikasikan apakah data multimedia tersebut mengandung *watermark* atau tidak.

Baik *watermark* privat (yang disisipkan) maupun *watermark* publik (referensi untuk pendeteksian) keduanya harus berkorelasi. Ada banyak cara untuk membangkitkan dua buah *watermark* yang berkorelasi. Di dalam skema [7], *watermark* privat dan publik dibangkitkan dengan menggunakan trannsformasi linier dan balikan *transpose*-nya. Di dalam skema [5], penulisnya menggunakan permutasi rahasia (dari himpunan permutasi yang dibangkitkan) untuk membangkitkan watermark privat dari watermark publik.

Di dalam makalah ini dipresentasikan skema *asymmetric watermarking* berbasiskan *chaos*. *Chaos* diterapkan karena ia mempunyai karakteristik penting untuk meningkatkan keamanan, yaitu sensitivitas pada kondisi awal. Karakteristik ini cocok untuk enkripsi dan *watermarking* [8] Fungsi *chaos* digunakan untuk membangkikan nilai-nlai bobot. *Watermark* publik dibangkitkan dengan menjumlahkan barisan *chaos* ini kepada *watermark* privat. Data multmedia yang disisipi *watermark* adalah citra *greyscale*. Baik penyisipan maupun pendeteksian *watermark* keduanya dilakukan pada ranah *discrete cosine transform* (*DCT*).

2. *CHAOS* DAN *WATERMARKING*

Karakteristik umum pada 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].

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 teletak di antara 0 dan 1 dan tersebar secara merata serta tidak ada dua nilai yang sama.

Sifat *chaos* yang peka terhadap perubahan kecil berarti jika nilai awal x_0 diubah sedikit, misalnya sebesar 0.00001, maka setelah beberapa kali iterasu diperoleh barisan nilai *chaos* yang berbeda dan semakin lama nilainya mengalami divergensi dari nilai-nilai *chaos* semula.

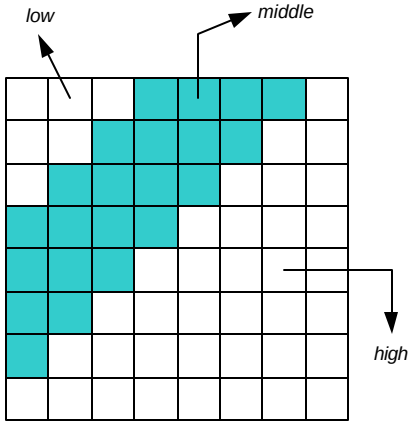
Beberapa tahun terakhir teori *chaos* banyak digunakan di dalam *digital watermarking*. *Chaos* digunakan khususnya sebagai pembangkit bilangan acak. Barisan nilai *chaos* digunakan langsung sebagai *watermark* [4] atau menyatakan lokasi penyisipan *watermark* di dalam citra [6].

3. *WATERMARKING* DALAM RANAH *DCT*

Menyisipkan dan mendeteksi *watermark* dalam ranah *transform* menghasilkan *robustness*

yang lebih tinggi dibandingkan dalam 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*. Kakas yang digunakan untuk transformasi adalah *FFT*, *DCT*, *DWT*, Fourier-Melin, dan lain-lain. Skema *watermarking* yang digunakan di dalam skema ini adalah *DCT*. Transformasi *DCT* dapat dilakukan terhadap keseluruhan citra atau pada blok-blok citra berukuran 8 x 8. Dengan mengacu pada kompresi JPEG, *watermarking* berbasis blok berukuran 8 x 8 umumnya lebih *robust* [12]. Metode *watermarking* dalam makalah ini berbasis blok 8 x 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).



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

Koefisien *DCT* pada area *middle* dapat diperoleh dengan cara berikut: mula-mula semua koefisien *DCT* (kecuali nilai *DC*) dari blok citra dibaca dengan memindai secara zig-zag, seperti halnya pada kompresi *JPEG*. Kemudian semua koefisien *DCT* dari indeks ke-(*s* + 1) hingga indeks ke-($s + 64 \times \frac{M}{N_1 \times N_2}$) diambil dari susunan zig-zag tadi [12]. *s* adalah jumlah koefisien *DCT* yang dilompati, *M* adalah panjang *watermark* dan N_1 x N_2 adalah ukuran citra. *Watermark* disisipkan pada koefisien-koefisien *DCT* yang terpilih ini (akan dijelaskan di bawah ini).

4. METODE YANG DIUSULKAN

Ada tiga tahapan proses yang dilakukan di dalam metode *asymmetric watermarking* yang diusulkan di dalam makalah ini. Masing-masing tahap dijelaskan di dalam sub-bab berikut.

4.1 Pembangkitan *Watermark* Privat dan Publik

Watermark yang akan disisipkan memiliki ukuran (*d*) kira-kira seperempat dari ukuran citra. Jika citra berukuran N_1 x 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* privat $\mathbf{w_s}$ berdasarkan *N*(0, 1):

$$\mathbf{w_s} = (w_s(1), w_s(2), ..., w_s(M))$$

Selanjutnya bangkitkan barisan *chaos* (rahasia) dengan nilai awal *i*:

$$\mathbf{k} = (k(1), k(2), ..., k(M))$$

Barisan *chaos* ini dijumlahkan ke *watermark* privat untuk menghasilkan *watermark* publik $\mathbf{w_p}$:

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

yang dalam hal ini

$$w_p(i) = k(i) + \cdot w_s(i),\ i = 1, 2, ..., M \quad (2)$$

Persamaan (2) ini mengindikasikan bahwa kedua *watermark* berkorelasi satu sama lain.

4.2 Penyisipan *Watermark*

Citra *I* yang berukuran *N* x *M* dibagi menjadi blok-blok kecil berukuran 8 x 8. Setiap blok ditransformasi dengan *DCT*, lalu koefisien DCT dipindai secara *zig-zag* dan semua koefisien DCT pada bagian middle frequency diambil (seeprti dijelaskan di dalam sub-bab 3). Misalkan koefisien-koefisien *DCT* yang terpilih ini disimpan di dalam larik *f*. Penyisipan *watermark* ke dalam *f* dilakukan dengan persamaan berikut [12]:

$$f_w(i) = f(i) + \alpha\, |f(i)|\, w_s(i) \quad (3)$$

yang dalam hal ini α adalah faktor kekuatan *watermark* ($0 < \alpha < 1$) yang dipilih sedemikian rupa sehingga *watermark* tidak dapat dipersepsi secara visual namun masih dapat dideteksi. Di dalam persamaan (3), *watermark* diskalakan dengan nilai mutlak koefisien *DCT* sebelum dijumlahkan ke dalam koefisien tersebut.

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

4.3 Pendeteksian *Watermark*

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

Citra yang diterima dibagi menjadi blok-blok berukuran 8 x 8, lalu koefisien DCT pada bagian *middle frequency* (yang mungkin mengalami kerusakan karena *non-malicious attack*) diekstraksi. Misalkan koefisien-koefisien *DCT* yang diekstrakasi ini disimpan di dalam larik *f**.

Pendeteksian dilakukan dengan menghitung korelasi antara *f** dan *watermak* publik $\mathbf{w_p}$:

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

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 citra tidak mengandung *watermark*. *T* bergantung pada citra yang diuji dan dapat dihampiri dengan persamaan [12]:

$$T = a \cdot m_{|f^*|} \cdot s^2_{W_p}/2 \quad (5)$$

yang dalam hal ini $m_{|f^*|}$ adalah nilai rata-rata $|f^*|$ dan $s^2_{W_p}$ adalah variansi *watermark* publik $\mathbf{w_p}$.

5. EKSPERIMEN DAN HASIL

Metode ini diuji dengan menggunakan kakas MATLAB 7. Citra yang digunakan adalah citra *Lena* (256 x 256). *Watermark* yang disisipkan berukuran 128 x 128 dan mempunyai distribusi normal berdasarkan N(0, 1). Nilai $s = 9$, $\alpha = 0.15$. Nilai awal *logistic map* yang digunakan adalah $i = 0.1$. Gambar 2(a) memperlihatkan citra asal dan Gambar 2(b) adalah citra yang telah mengandung *watermark* (PSNR = 47,0).

(a) (b)

Gambar 2. (a) Citra Lena asli, (b) citra Lena yang sudah mengandung *watermark* (PSNR = 47,0).

Pada kasus tidak ada serangan, nilai korelasi yang dihasilkan adalah $c = 0.4180$ (lebih besar dari $T = 0.2145$). Jika citra Lena yang diuji tidak

mengandung *watermark*, maka c = -0.0841 (nilai mutlaknya lebih kecil dari T = 0.2147).

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 hingga 50% (Gambar 3).

(a) c = 0.7424, T = 0.4402 (b) c = 0.9627, T = 0.5725

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

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).

(a) Kualitas 100% c = 0.9604, T = 0.5761 (b Kualitas 60% c = 0.9746, T = 0.4454 (c) Kualitas 20% c = 0.9627, T = 0.5725

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

Eksperimen 3: Penajaman dan derau

Citra ber-*watermark* dipertajam sehingga tepi-tepi di dalam citra terlihat lebih menonjol. kompresi ke format *JPEG* dengan kualitas kompresi 100%, 60%, 20%. *Watermark* masih dapat dideteksi dari citra hasil kompresi (Gambar 5). Sedangkan untuk menguji ketahanan terhadap derau, citra ditambahkan dengan derau berupa *salt and peppers* 2%, ternyata *watermark* masih dapat dideteksi. Untuk derau lebih besar dari 2%, nilai korelasi c sedikit di bawah nilai ambang T (jika nilai T diturunkan, *watermark* masih dapat dianggap berhasil dideteksi). Lihat Gambar 5.

(b) c = 5.5455, T = 2.9624 (b) c = 1.2841, T = 1.3816

Gambar 5. (a) Penajaman citra, (b) Distorsi karena derau *salt and peppers* sebesar 8%.

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. Untuk perbesaran hingga 2 kali ukuran semula, *watermark* juga masih dapat dideteksi (c = 0.7535, T = 0.5525).

(a) c = 0.6145, T = 0.4581

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

6. ANALISIS *MALICIOUS ATTACK*

Serangan *malicious attack* bertujuan untuk menghapus *watermark* dari citra dengan melakukan manipulasi persamaan (3) untuk memperoleh $f(i)$. Informasi lain seperti $f_w(i)$, $\mathbf{w_p}$, dan α dimiliki oleh penyerang. Tetapi, penyerang harus mengetahu $\mathbf{w_s}$

agar bisa menghapus *watermark* dari dalam citra. Untuk mendapatkan $\mathbf{w_s}$, penyerang melakukan operasi pengurangan berikut:

$$\mathbf{w}_s = \boldsymbol{w}_p - \mathbf{k} \quad (7)$$

Oleh karena vektor **k** rahasia, maka penyerang tidak dapat melakukan hal ini. Jika penyerang mencoba membangkitkan **k**, maka ada tidak berhingga kemungkinan **k** yang 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.

7. KESIMPULAN

Di dalam makalah ini telah dipresentasikan metode *asymmetric watermarking* berbasis *chaos* dalam ranah *DCT* pada citra digital Penyisipan dilakukan pada area *middle frequency* dari ranah *DCT* untuk memperoleh keseimbangan antara *imperceptibility* dan *robustness*. *Watermark* publik diperoleh dengan menjumlahkan *watermark* privat dengan barisan *chaos*. Hasil eksperimen menunjukkan bahwa metode ini terbukti *robust* terhadap serangan *non-malicious attack* (kompresi, *cropping*, *resizing*, *sharpening*, distorsi karena derau) dan *malicious attack* (*exhaustive search* untuk menemukan barisan nilai *chaos*).

8. DAFTAR PUSTAKA

[1] Ingemar J. Cox, dkk, "*Secure Spread Spectrum Watermarking for Multimedia*", 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.