ISSN. 1412-0100 VOL 16, NO 1, APRIL 2015

Aplikasi Teknik *Adaptive Digital Image Watermarking* Untuk Proteksi Hak Cipta Citra Digital

Syanti Irviantina[1], Sunario Megawan[2], Jonni[3]
STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789
[1,2,3]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[1]syanti@mikroskil.ac.id, [2]sunario@mikroskil.ac.id, [3]071110519@students.mikroskil.ac.id

Abstrak

Seiring dengan semakin meluasnya jaringan multimedia, maka proses pengiriman dan pengaksesan dari data digital juga semakin mudah. Dengan adanya kemudahan ini menyebabkan mudahnya duplikasi data yang berimbas pada pemalsuan dan plagiat terhadap data digital. Hal ini menyebabkan penyalahgunaan dan pelanggaran hak cipta. Dalam mengatasi hal ini, telah banyak dikembangkan metode-metode yang digunakan untuk melindungi karya-karya cipta digital. Salah satunya adalah dengan menggunakan teknik watermarking. Watermarking yang menggunakan teknik thresholding pada digital watermarking memiliki problema dimana harus dipilih sebuah nilai threshold yang cocok. Proses pemilihan nilai threshold yang cocok ini relatif susah. Dengan menggunakan metode Adaptive Digital Image Watermarking, citra yang disisipkan tidak akan dapat dideteksi secara kasat mata dan tidak membutuhkan penentuan nilai threshold. Pada pengujian yang dilakukan terhadap citra watermark, citra yang disisipkan masih dapat diekstrak walaupun dilakukan perubahan contrast dan brightness ataupun kompresi pada kualitas minimal 60%.

Kata kunci— citra digital, *watermarking, Adaptive Digital Image Watermarking*

Abstract

Because of the expansion of multimedia networks the process of delivery and access of digital data become easier. It make easy duplication of data which impact on forgery and plagiarism of the digital data and make abuse and copyright infringement. There are several methods has been developed to protect digital copyright works, as is by using watermarking technique. Watermarking using thresholding techniques in digital watermarking has a problem which should have been a suitable threshold value and process of selecting a suitable threshold value is not easy. By using Adaptive Digital Image Watermarking, the inserted image will not be detected and does not require the determination of the threshold value. In tests performed on the watermarked image, the image still can be extracted even if change contrast and brightness or compression on the quality of at least 60%.

Keywords— *digital image, watermarking, Adaptive Digital Image Watermarking*

1. PENDAHULUAN

Proses evolusi yang pesat dari Internet membuat proses transmisi dari konten *digital multimedia* seperti teks, *audio*, citra digital dan *video* menjadi jauh lebih mudah. Media digital dapat diakses dan didistribusikan melalui jaringan. Akibatnya, proses duplikasi menjadi mudah dan tidak mengurangi isinya dimana hasil duplikasi menjadi identik dengan yang asli. Duplikasi identik yang tidak terbatas dari sebuah media digital dapat diproduksi dengan mudah [1]. Hal ini menjadi ancaman serius bagi hak cipta dari pemilik media. Oleh karena itu, proses proteksi terhadap kepemilikan intelektual merupakan sebuah hal penting bagi dunia digital. Teknik *digital watermarking*

diperkenalkan untuk proteksi hak cipta atau identifikasi kepemilikan dari media digital, seperti *audio*, citra digital, *video*, ataupun teks.

Kebanyakan algoritma *watermark* menggunakan sebuah *serial number* ataupun identitas pemilik sebagai *watermark*. Teknik *adaptive digital image watermarking* memiliki kelebihan seperti tidak dapat dideteksi oleh mata manusia, tahan terhadap operasi pengolahan citra *lossy* dan tidak perlu menggunakan nilai *threshold* sehingga hasil ekstraksi dapat lebih akurat.

Dengan adanya kemudahan dalam menduplikasi citra digital, maka diperlukan teknik-teknik tertentu untuk dapat menyisipkan hak kepemilikan citra digital untuk mengetahui identitas pemilik asli dari citra digital tersebut. Salah satu teknik tersebut adalah teknik *thresholding* pada digital *watermarking* dimana teknik ini melakukan pemilihan sebuah nilai threshold yang sesuai, dan untuk memilih nilai threshold ini bukanlah perkara yang mudah. Disamping itu dalam menyisipkan citra digital kedalam sebuah media citra sampul, akan menghadapi masalah dimana ukuran citra digital yang akan disisipkan biasanya berukuran besar. Untuk itu artikel ini akan menjelaskan bagaimana menyisipkan hak kepemilikan citra digital untuk mengetahui identitas pemilik asli dari citra digital tersebut sehingga dapat melakukan proteksi terhadap citra digital tersebut

Dalam artikel ini format citra sampul dan citra biner hanya yang memiliki format *.BMP dan *.JPG, dengan ukuran citra biner maksimal sebesar citra sampul dan eksitensi citra watermark sama dengan eksistensi citra sampul. PRNG (*Pseudo-Random Number Generator*) yang digunakan adalah algoritma LCG (*Linear Congruential Generator*) dengan nilai kunci k diantara -1 dan 1 serta nilai awal algoritma LCG a, b dan X_0 yang digunakan bertipe data bilangan *integer* dengan batasan minimal 2 dan maksimal 1.000.000.

2. KAJIAN PUSTAKA

2.1 *Digital Watermark*

Digital watermarking adalah penyisipan sinyal digital ke dalam suatu media digital. *Digital watermarking* ini berangkat dari proses-proses pengolahan sinyal digital, dimana sinyal digital dapat berupa gambar, *audio*, *video*, dan teks. Seperti yang telah disebutkan sebelumnya, bahwa *digital watermarking* ini diimplementasikan dengan memanfaatkan kekurangan dari indera manusia (indera penglihatan dan indera pendengaran) dimana indera manusia ini kurang sensitif terhadap perubahan yang terjadi, misalnya saja perubahan yang terjadi pada level bit (sampai batas tertentu), perubahan pada level frekuensi (di luar frekuensi yang diterima manusia

2.2 *Watermarking* untuk Pelabelan Hak Cipta

Masalah Hak Cipta dari dahulu sudah menjadi hal yang utama dalam segala ciptaan manusia, ini digunakan untuk menjaga *originalitas* atau *kreatifitas* pembuat akan hasil karyanya. Hak cipta terhadap data-data digital sampai saat ini belum terdapat suatu mekanisme atau cara yang handal dan efisien. Beberapa cara yang pernah dilakukan oleh orang-orang untuk mengatasi masalah pelabelan hak cipta pada data digital, antara lain [2]:

1. *Hearder Marking*, dengan memberikan keterangan atau informasi hak cipta pada header dari suatu data digital.
2. *Visible Marking*, merupakan cara dengan memberikan tanda hak cipta pada data digital secara eksplisit.
3. *Encryption*, mengkodekan data digital ke dalam representasi lain yang berbeda dengan representasi aslinya (tetapi dapat dikembalikan ke bentuk semula) dan memerlukan sebuah kunci dari pemegang hak cipta untuk mengembalikan ke representasi aslinya.
4. *Copy Protection*, memberikan proteksi pada data digital dengan membatasi atau dengan memberikan proteksi sedemikian rupa sehingga data digital tersebut tidak dapat diduplikasi.

Watermarking sebagai metoda untuk pelabelan hak cipta dituntut memiliki berbagai kriteria (ideal) sebagai berikut agar memberikan unjuk kerja yang bagus [2]:

1. Label Hak Cipta yang unik mengandung informasi pembuatan, seperti nama, tanggal, dst, atau sebuah kode hak cipta seperti halnya ISBN (*International Standard for Book Notation*) pada buku-buku.
2. Data terlabel tidak dapat diubah atau dihapus (*robustness*) secara langsung oleh orang lain atau dengan menggunakan software pengolahan sinyal sampai tingkatan tertentu.
3. Pelabelan yang lebih dari satu kali dapat merusak data digital aslinya, supaya orang lain tidak dapat melakukan pelabelan berulang terhadap data yang telah dilabel.

2.3 *Pseudo Random Number Generator* (PRNG)

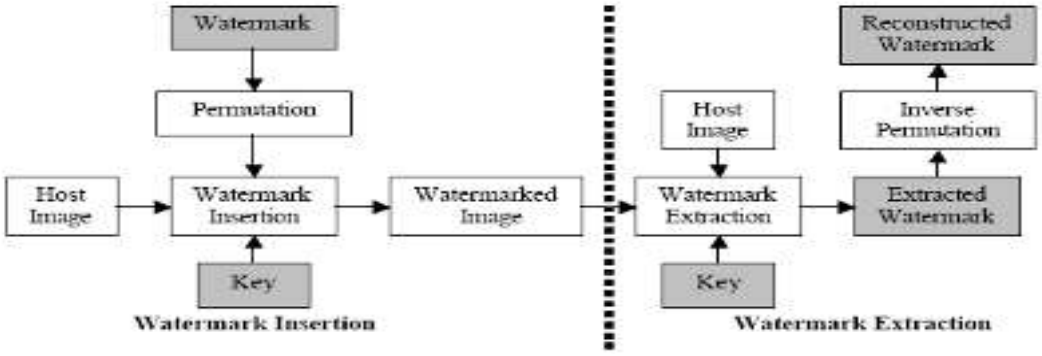
Suatu *pseudo random number generator* (PRNG) merupakan suatu algoritma yang menghasilkan suatu urutan nilai dimana elemen-elemennya bergantung pada setiap nilai yang dihasilkan. *Output* dari PRNG tidak betul-betul acak, tetapi hanya mirip dengan properti dari nilai acak. Kebanyakan algoritma dari *pseudo random number generator* ditujukan untuk menghasilkan suatu sampel yang secara seragam terdistribusi. Perlu diingat, semua deretan bilangan acak yang dibangkitkan dari rumus matematika, serumit apapun, dianggap sebagai deret acak semu, karena dapat diulang pembangkitannya. Sementara itu, banyak produk *software* yang dinyatakan aman karena menggunakan bilangan acak semu. Namun karena bilangan acak yang dibangkitkan bersifat semu, maka keamanan yang diperoleh juga semu. *Linear Congruential Generator* (LCG) mewakili salah satu algoritma *pseudo-random number* yang tertua dan paling populer. LCG dapat didefinisikan dengan rumusan pada persamaan (1).

$$x_n = (ax_{n-1} + b) \bmod m \tag{1}$$

Periode dari LCG umumnya adalah sebesar nilai m. Pada penelitian ini, nilai m adalah sebesar jumlah blok citra.

2.4 Algoritma *Adaptive Image Watermarking*

Watermark yang digunakan adalah sebuah citra biner. Oleh karena itu, mata manusia dapat dengan mudah mengidentifikasi *watermark* yang terekstrak. Kenyataannya, penempelan sebuah *watermark* pada *least significant bit* dari sebuah piksel kurang sensitif pada mata manusia. Namun, *watermark* akan rusak jika dilakukan operasi pengolahan citra seperti *low-pass filtering* pada citra hasil *watermarking*. Untuk membuat *watermark* yang ditempelkan tahan terhadap penyerangan, *watermark* harus ditempelkan pada *more significant bit*. Namun, hal ini akan mengakibatkan kualitas citra hasil *watermarking* menjadi kurang bagus dan akan terdeteksi oleh mata manusia. Agar dapat memenuhi persyaratan ketangguhan dan tidak kelihatan ini, [1] melakukan modifikasi secara adaptif terhadap nilai intensitas dari beberapa piksel yang dipilih semaksimal mungkin dan modifikasi ini tidak akan terdeteksi oleh mata manusia. Sebagai tambahan, untuk mencegah perusakan atau pengubahan citra, *watermark* akan dipermutasi terlebih dahulu menjadi data teracak. *Block diagram* dari sistem *watermark* yang dibahas dapat dilihat pada gambar 1.



Gambar 1. Block Diagram system watermarking

2.5 Algoritma Penempelan

Pada metode yang dikemukakan oleh [1] ini, *watermark* yang ditempelkan harus tidak dapat dideteksi oleh mata manusia dan tahan terhadap kebanyakan operasi pengolahan citra. Untuk memenuhi persyaratan ini, sebuah bit dari nilai piksel biner (0 atau 1) akan ditempelkan pada sebuah blok dari citra sampul. Sebelum penempelan, citra sampul akan dibagi menjadi N x N blok. Tergantung pada konstras dari sebuah blok, piksel pada blok akan dimodifikasi secara adaptif untuk memaksimalkan ketangguhan dan menjamin agar *watermark* tidak terdeteksi. Posisi atau blok yang digunakan untuk penempelan akan dipilih dengan menggunakan sebuah generator pembangkit bilangan acak (*Pseudo-Random Number Generator* / PRNG) dengan menggunakan sebuah nilai kunci k. Anggap B adalah blok yang dipilih, g_{max} adalah intensitas maksimal dari blok, g_{min} adalah intensitas minimal dari blok dan g_{mean} adalah intensitas rata-rata dari blok. Rumus yang digunakan pada g_{max}, g_{min}, g_{mean} dapat dilihat pada persamaan (2).

$$g_{max} = max(b_{ij}, 0 \le i,j < N)$$
$$g_{min} = min(b_{ij}, 0 \le i,j < N)$$
$$g_{mean} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} b_{ij} \quad (2)$$

dimana b_{ij} merepresentasikan intensitas dari piksel (i, j) pada blok B. Anggap bahwa nilai piksel yang ditempelkan b_w adalah 0 atau 1. Proses penempelan akan memodifikasi intensitas dari piksel pada blok tergantung pada aturan pada persamaan (3).

$$b_w = 1 ; g' = g_{max} \ if \ g > g_{mean}$$
$$g' = g + \delta \quad ig \ g < g_{mean}$$
$$b_w = 0; \ g' = g_{min} \quad if \ g < g_{mean}$$
$$g' = g - \delta \quad ig \ g \ge g_{mean} \quad (3)$$

dimana g' adalah intensitas hasil modifikasi dan δ adalah sebuah nilai kecil yang digunakan untuk memperbaiki intensitas. Proses penempelan dari *watermark* tergantung pada isi dari setiap blok. Jika blok merupakan konstras tinggi, maka intensitas dari piksel akan dimodifikasi secara besar-besaran. Jika tidak, maka intensitas akan dimodifikasi sedikit saja. Oleh karena itu, algoritma ini dapat memodifikasi isi dari sebuah blok secara adaptif. Anggap B adalah blok citra asli dan dan B' adalah blok citra hasil *watermark*, penjumlahan dari intensitas piksel dari B' akan lebih besar daripada B jika nilai piksel *watermark* yang dimasukkan b_w bernilai 1.

2.6 Algoritma Ekstraksi *Watermark*

Pada algoritma ini, algoritma ekstraksi dari sebuah *watermark* memerlukan input berupa citra asli. Pertama, gunakan nilai k untuk menghasilkan sebuah deretan dari posisi atau blok dimana *watermark* ditempelkan. Anggap B adalah blok citra asli dan dan B' adalah blok citra hasil *watermark*, untuk setiap posisi yang terpilih, hitunglah total penjumlahan dari intensitas piksel, S_0 dan S_w dari B dan B'. Nilai bit *watermark* yang terekstrak keluar b_w dapat ditentukan dengan menggunakan aturan pada persamaan (4).

$$b_w = 1 \quad if \ S_w > S_o$$
$$b_w = 0 \quad if \ S_w \le S_o \quad (4)$$

Nilai bit *watermark* terekstrak b_w akan dipermutasi secara kebalikan untuk memperoleh *watermark* yang direkonstruksi ulang.

2.7 *Mean Square Error* (MSE)

Mean Square Error (MSE) adalah kesalahan kuadrat rata-rata. Nilai MSE didapat dengan membandingkan nilai selisih piksel-piksel citra asal dengan citra hasil pada posisi piksel yang sama. Semakin besar nilai MSE, maka tampilan pada citra hasil akan semakin buruk. Perhitungan MSE dilakukan dengan menggunakan persamaan (5).

$$MSE = \frac{1}{MN} \sum_{i-0}^{M-1} \sum_{j-0}^{N-1} (x[i,j] - y\,[i,j])^2 \quad (5)$$

dimana x[i,j] adalah citra asal dengan dimensi M x N, dan y[i,j] adalah citra hasil *watermarking* [2].

2.8 *Peak Signal to Noise Ratio* (PSNR)

Peak Signal to Noise Ratio (PSNR) digunakan untuk menentukan kualitas citra. Nilai PSNR diperoleh dengan membandingkan citra asli dan citra rekonstruksi. Untuk menentukan nilai PSNR digunakan persamaan (6).

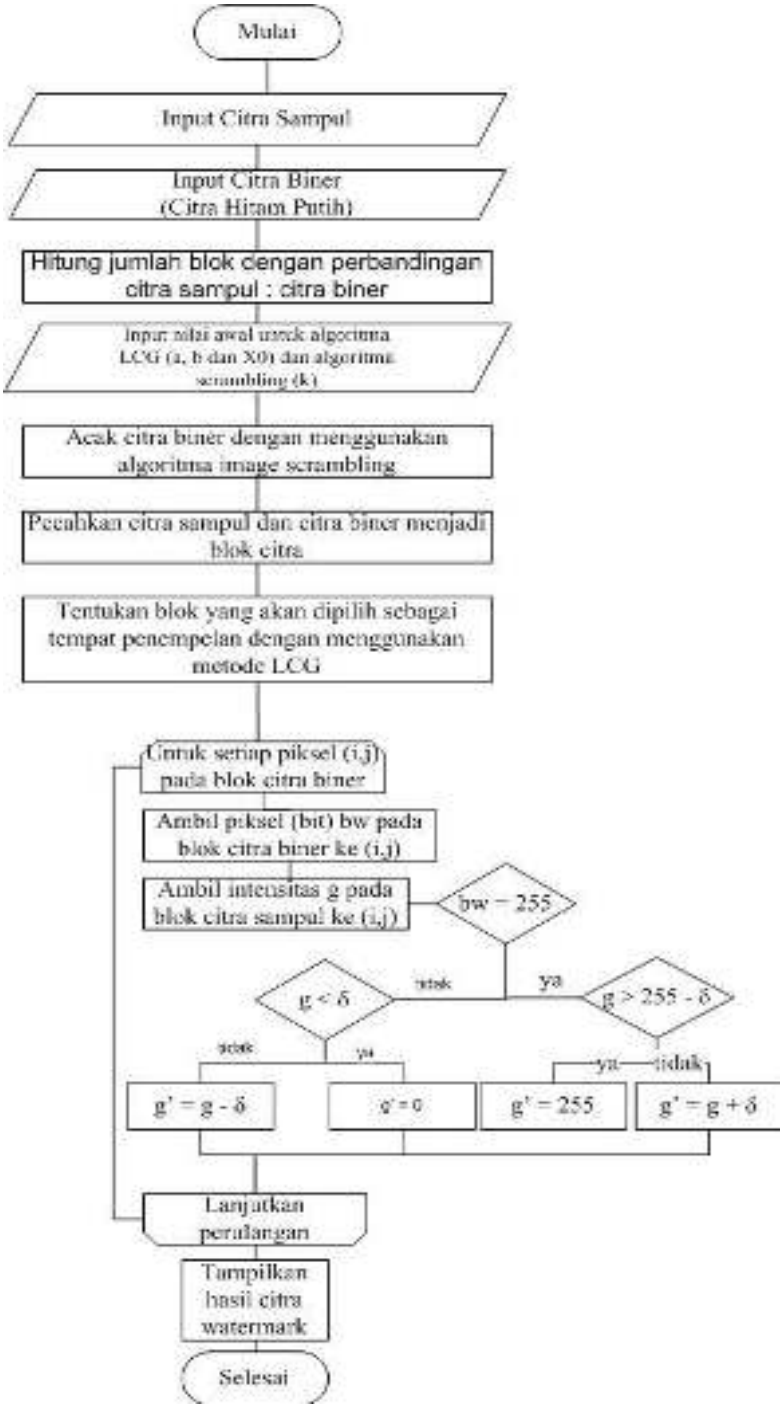
$$PSNR = 10 * \log_{10}\left(\frac{MAX}{MSE}\right) \quad (6)$$

Nilai PSNR ditentukan oleh besar kecilnya nilai MSE yang terjadi pada citra. Semakin besar nilai PSNR, semakin baik pula hasil citra *watermark* yang diperoleh dari citra asli.

3. METODE PENELITIAN

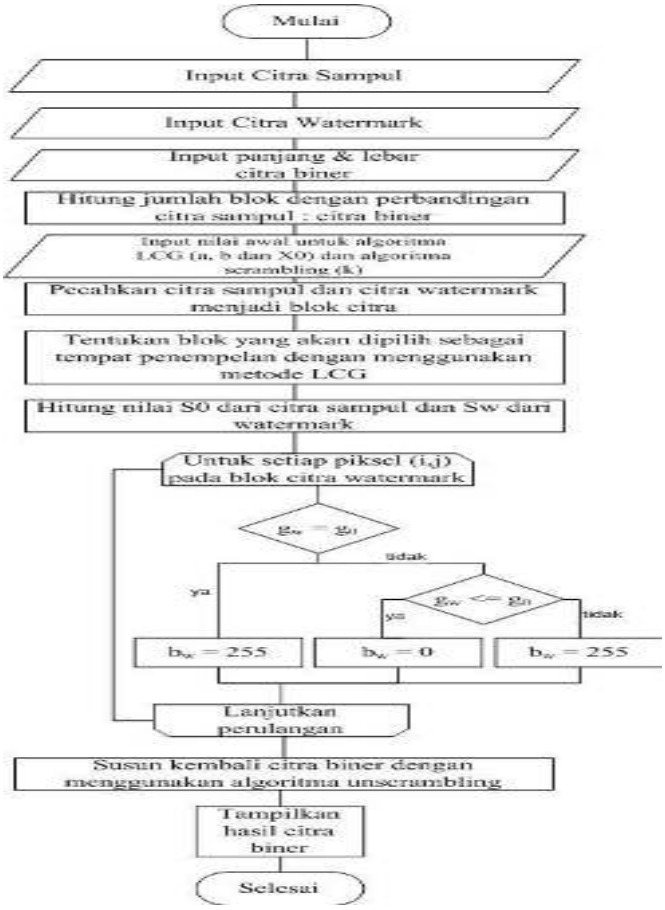
Secara umum, metode yang digunakan pada penelitian ini terdiri dari beberapa tahapan sebagai berikut:

1. Proses penyisipan *watermark* akan dimulai dari pengisian semua nilai yang diperlukan, seperti citra asli, citra biner dan parameter input lainnya seperti nilai k untuk algoritma *scrambling*, nilai a, b dan X_0 untuk algoritma LCG seperti diperlihatkan pada gambar berikut ini.



Gambar 2. Algoritma proses penyisipan watermark

2. Proses ekstraksi *watermark* akan dimulai dari pengisian semua nilai input yang diperlukan, seperti citra asli, citra hasil *watermark* dan parameter input lainnya seperti panjang dan lebar citra biner, nilai k untuk algoritma *scrambling*, nilai a, b dan X_0 untuk algoritma LCG

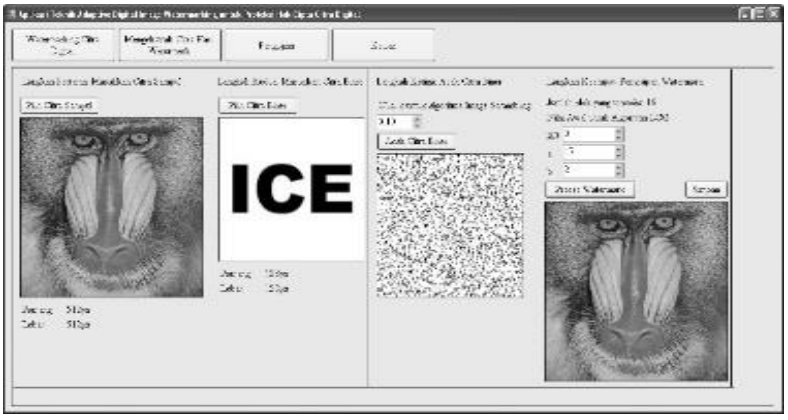


Gambar 3. Algoritma proses ekstraksi watermark

4. HASIL DAN PEMBAHASAN

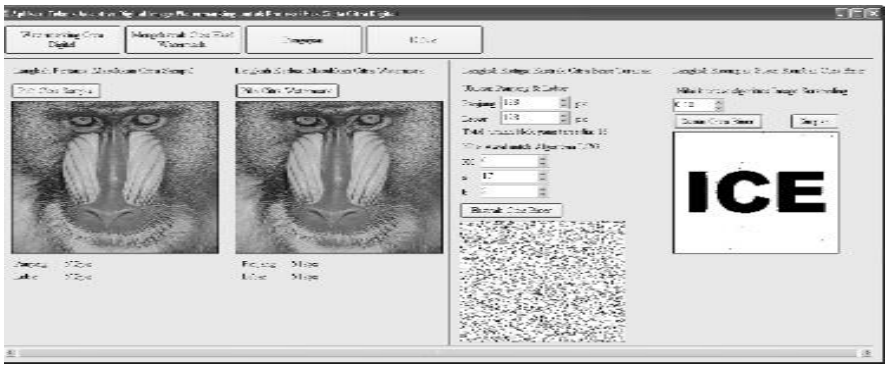
4.1 Hasil

Cara kerja aplikasi ini adalah dengan menyisipkan citra biner ke dalam citra asli yang kemudian akan ditampilkan kembali menjadi sebuah citra ter-*watermark*.



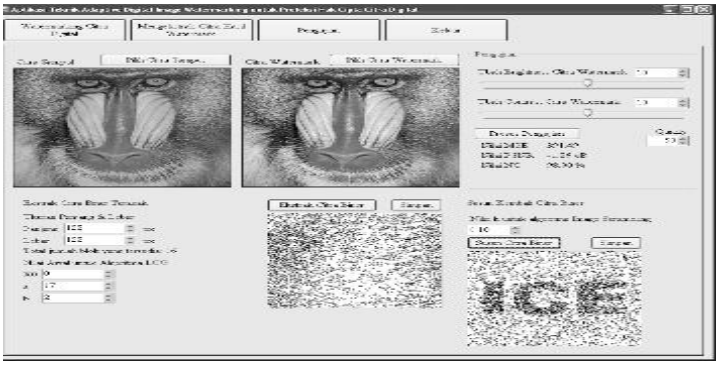
Gambar 4. Penyisipan citra digital

Kemudian dilakukan pengekstraksian citra *watermark* yang kemudian akan ditampilkan kembali menjadi sebuah citra biner.



Gambar 5. Ekstraksi citra watermark

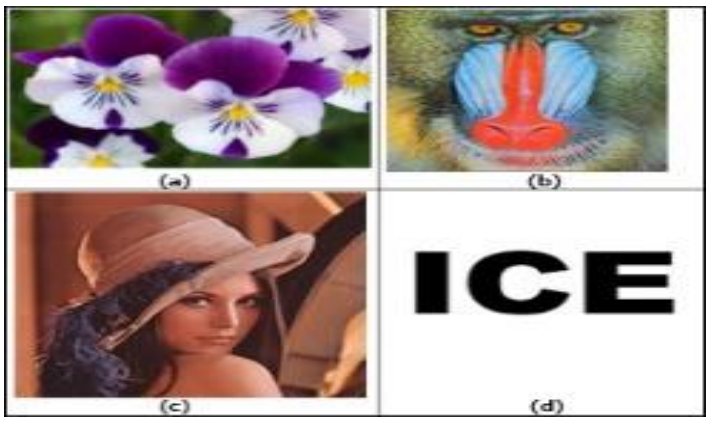
Kemudian dilakukan pengujian citra *watermark* terhadap citra sampul yang akan menampilkan nilai MSE, PSNR dan NC. Tampilan pengujian disertai fitur ekstraksi *watermark* untuk memudahkan pengguna mengekstrak citra biner dari citra *watermark* yang telah dimodifikasi



Gambar 6. Pengujian citra digital

4.2 Pembahasan

Didalam pengujian ini dibagi tiga tahap, yaitu tahap pengujian penyisipan *watermark*, tahap pengujian ekstraksi *watermark* dan tahap pengujian ketahanan *watermark*. Citra sampul yang akan diuji seperti pada gambar 7 adalah berukuran 1024px * 1024px (a), 512px * 512px (b) dan 256px * 256px (c)dan untuk citra biner digunakan ukuran 128px * 128px (d).



Gambar 7. Citra digital yang diuji

4.2.1 Pengujian Penyisipan *Watermark*

Proses pengujian penyisipan *watermark* dilakukan dengan menguji beberapa citra untuk mengetahui apakah citra biner yang telah disisipkan dapat terlihat secara kasat mata atau tidak. Berikut tabel pengujian penyisipan *watermark* dengan setiap citra memiliki kuncinya masing-masing.

Tabel 1. Pengujian penyisipan watermark

Citra Sampul	Citra Biner	Nilai k dari Algoritma *Image Scrambling*	Hasil *Image Scrambling*	Nilai x_0, a dan b dari Algoritma LCG	Hasil Penyisipan
		k = 0.1		$x_0 = 0$; a = 25 ; b = 3	
		k = 0.2		$x_0 = 0$; a = 33 ; b = 5	
		k = 0.3		$x_0 = 0$; a = 33 ; b = 7	

4.2.2 Pengujian Ekstraksi Watermark

Setelah proses pengujian penyisipan *watermark*, dilakukan proses ekstraksi *watermark* untuk mengetahui apakah citra biner yang telah disisipkan dapat diekstrak kembali. Berikut tabel hasil pengujian ekstraksi *watermark* dengan nilai kunci algoritma yang sesuai dengan kunci proses penyisipan *watermark*

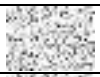
Tabel 2. Pengujian ekstraksi watermark

Citra Sampul	Citra *Watermark*	Nilai x_0, a dan b dari Algoritma LCG	Citra Hasil Ekstraksi	Nilai k dari Algoritma *Image Scrambling*	Citra Biner
		$x_0 = 0$; a = 25 ; b = 3		k = 0.1	
		$x_0 = 0$; a = 33 ; b = 5		k = 0.2	
		$x_0 = 0$; a = 33 ; b = 7		k = 0.3	

4.2.3 Pengujian Ketahanan Watermark

Ketahanan *watermark* diuji dengan 4 cara, yaitu pengujian ketahanan *watermark* terhadap nilai k dari Algoritma *Image Scrambling* (table 3), pengujian ketahanan *watermark* terhadap nilai x_0, a dan b dari Algoritma LCG (table 4), pengujian ketahanan *watermark* terhadap kompresi citra (table 5) dan pengujian ketahanan *watermark* terhadap serangan dari kombinasi *brightness* dan *contrast* (table 6)

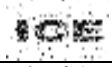
Tabel 3. Pengujian ketahanan watermark terhadap nilai k dari algoritma Image Scrambling

Citra Sampul	Citra *Watermark*	Nilai x_0, a dan b dari Algoritma LCG	Citra Hasil Ekstraksi	Nilai k dari Algoritma *Image Scrambling*	Citra Biner
		$x_0 = 0$; a = 25 ; b = 3		k = 0.1	
		$x_0 = 0$; a = 33 ; b = 5		k = 0.1	
		$x_0 = 0$; a = 33 ; b = 7		k = 0.1	

Tabel 4. Pengujian ketahanan watermark terhadap nilai x0,a dan b dari algoritma LCG

Citra Sampul	Citra Watermark	Nilai x_0, a dan b dari Algoritma LCG	Citra Hasil Ekstraksi	Nilai k dari Algoritma Image Scrambling	Citra Biner
		$x_0 = 0$; a = 25 ; b = 3		k = 0.1	
		$x_0 = 0$; a = 33 ; b = 5		k = 0.2	
		$x_0 = 0$; a = 33 ; b = 7		k = 0.3	

Tabel 5. Pengujian ketahanan watermark terhadap kompresi citra

Citra Sampul	Nilai Kualitas Citra (q) dan Citra Watermark	Nilai MSE, PSNR dan NC	Citra Hasil Ekstraksi	Nilai k dari Algoritma Image Scrambling dan Citra Biner
	q = 90 % 	MSE = 1.70 ; PSNR = 21.73 NC = 99 %		k = 0.1 
	q = 90 % 	MSE = 5.5 ; PSNR = 16.24 NC = 99 %		k = 0.1 
	q = 90 % 	MSE = 1.74 ; PSNR = 21.32 NC = 99 %		k = 0.1 

Tabel 6. Pengujian ketahanan watermark terhadap serangan dari kombinasi brightness dan contrass

Citra Sampul	Citra Watermark	Nilai brightness(b), contrast (c), MSE, PSNR dan NC	Nilai x_0, a dan b dari Algoritma LCG	Citra Hasil Ekstraksi	Citra Biner
		b = 1 ; c = 1 ; MSE = 3.66 ; PSNR = 18.43 ; NC = 98.99 %	$x_0 = 0$; a = 25 ; b = 3		k = 0.1 
		b = 1 ; c = 1 ; MSE = 2.27 ; PSNR = 20.13 ; NC = 98.99 %	$x_0 = 0$; a = 33 ; b = 5		k = 0.2 
		b = 1 ; c = 1 ; MSE = 0.25 PSNR = 29.77 ; NC = 99 %	$x_0 = 0$; a = 33 ; b = 7		k = 0.3 

Tabel 7. Pengujian ekstraksi watermark dengan ukuran citra biner yang berbeda.

Citra Sampul	Citra *Watermark*	Nilai x_0, a dan b dari Algoritma LCG	Ukuran panjang dan lebar Citra Biner	Citra Biner Hasil Ekstraksi	Citra Biner
		$x_0 = 0$; a = 25 ; b = 3	p = 128 px ; l = 128 px		k = 0.1 
		$x_0 = 0$; a = 33 ; b = 5	p = 128 px ; l = 128 px		k = 0.1
		$x_0 = 0$; a = 33 ; b = 7	p = 128 px ; l = 128 px		k = 0.1

5. KESIMPULAN

Berdasarkan hasil dan pengujian yang dilakukan pada bab sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Dengan menggunakan algoritma LCG dengan nilai kunci diantara -1 dan 1 sudah dapat menentukan posisi citra biner.

2. Dari hasil pengujian serangan *brightness* dan *contrast* terhadap citra hasil penyisipan, dengan nilai *brightness* dan *contrast* lebih besar dari 5 akan sulit mengekstrak kembali citra binernya.
3. Dari hasil pengujian serangan kompresi terhadap citra hasil penyisipan, citra *watermark* dapat bertahan pada kualitas diatas 60%.

6. SARAN

Berikut ini beberapa saran untuk pengembangan penelitian berikutnya:

1. Proses ekstraksi terhadap citra *watermark* memerlukan ukuran yang sama dengan ukuran citra sampul. Diharapkan dalam penelitian selanjutnya citra biner dapat terekstrak walaupun ukuran citra watermark berbeda dengan ukuran citra sampul.
2. Proses penentuan blok citra selain menggunakan metode LCG, bisa diganti dengan metode PRNG yang lain seperti LFG (*Lagged Fibonacci Generator*) atau *Mersenne Twister*.

DAFTAR PUSTAKA

[1] C.H. Lee & Y.K Lee, 1999, *An Adaptive Digital Image Watermarking Technique for Copyright Protection*, IEEE Xplore, No.4, Vol.45,1005-1015, http://ieeexplore.ieee.org/.

[2] Sutoyo, T., E. Mulyanto, V. Suhartono, O.D. Nurhayati,Wijanarto, 2009, Teori Pengolahan Citra, Ed.1,Penerbit Andi, Yogyakarta

[3] L. Xiangdong, Z. Junxing, Z. Jinhai dan H. Xiqin, 2008, *Image Scrambling Algorithm Based on Chaos Theory and Sorting Transformation*, IJCSNS International Journal of Comptuer Sicence and Network Security, No.1,Vol. 8,64-68, http://paper.ijcsns.org/07_book/200801/20080110.pdf

[4] Murni, A., 1992, Pengantar Pengolahan Citra, PT. Elex Media Komputindo, Jakarta.

[5] H.O Nasereddin,2011, Digital Watermarking A Technology Overview, IJRRAS International Journal of Research and Reviewa in Applied Sciences, No.1, Vol.6,89-93, http://www.arpapress.com/Volumes/Vol6Issue1/IJRRAS_6_1_10.pdf

[6] E.Ariwibowo, 2006, Implementasi Metode Steganography Transformasi Dalam Penyembunyian label Hak Cipta Data Digital, Telkomnika, No.3, Vol.2,101-108, http://portalgaruda.org/.