APLIKASI ALGORITMA *SEMI – FRAGILE IMAGE WATERMARKING* BERDASARKAN PADA *REGION SEGMENTATION*

Andri[1], Ng Poi Wong[2], Johnny Fransiscus[3]

STMIK Mikroskil
Jl. Thamrin No. 112, 124, 140 Medan 20212

andri@mikroskil.ac.id[1], poiwong@mikroskil.ac.id[2]

Abstrak

Kebanyakan algoritma *semi-fragile watermarking* yang telah ada memiliki kelemahan, seperti sifat tidak kelihatan (*invisibility*) yang jelek, *robustness* yang tidak sempurna pada beberapa rutin dari proses sinyal. Penyebab utamanya adalah karena kebanyakan algoritma menggunakan parameter kuantisasi tertentu tanpa mempertimbangkan perbedaan diantara citra.

Untuk menyelesaikan permasalahan ini, Shengbing Che, Bin Ma, Jinkai Luo dan Shaojun Yu dari China (2009) [1] memperkenalkan sebuah algoritma *image watermarking* berdasarkan pada segmentasi *region*. Proses kerja dari sistem dimulai dari pemilihan citra sampul dan citra *watermark* hitam putih. Setelah itu, dilanjutkan pengisian nilai kunci dan proses pengacakan terhadap citra hitam putih. Kemudian, citra sampul akan dikonversi ke bentuk *grayscale* dan terakhir citra hitam putih akan ditempelkan ke dalam citra sampul. Citra *watermark* yang diperoleh dapat disimpan ke dalam sebuah *file* yang akan digunakan pada proses ekstraksi untuk mengekstrak keluar citra *watermark* hitam putih yang disisipkan di dalamnya.

Hasil pengujian menunjukkan waktu eksekusi proses penempelan watermark relatif lebih cepat dari proses ekstraksi. Algoritma tidak bisa mengekstraksi citra hitam putih ketika citra *watermark* diberi efek *brightness* dan *contrast*. Citra hitam putih masih dapat terekstrak keluar walau citra *watermark* disisipkan *noise* atau di*crop*. Apabila ada kesalahan pengisian kunci maka akan menyebabkan citra hitam putih menjadi tidak terekstrak keluar.

Kata kunci : *semi-fragile watermarking*, *region segmentation*, *watermark*

1. Pendahuluan

1.1 Latar Belakang

Seiring dengan semakin berkembangnya popularitas *internet*, terutama dengan penerapan dari teknologi multimedia, masalah proteksi sekuritas dari informasi multimedia telah menjadi semakin penting. Sebagai proteksi efektif dari data multimedia, teknik *digital watermarking* telah menjadi fokus penelitian dari akademi internasional. Berdasarkan karakteristiknya, *digital watermarking* dapat dikelompokkan menjadi tiga tipe, yaitu *robust digital watermark*, *fragile digital watermark* dan *semi-fragile digital watermark*. Karena keterbatasan *bandwidth* jaringan, maka biasanya informasi multimedia ditransmisikan dalam bentuk kompresi data. Dalam hal ini, *semi fragile watermarking* memiliki keunggulan pada aspek *fragility* dan *robustness*.

Walaupun demikian, masih terdapat banyak kelemahan pada algoritma *semi-fragile watermarking* yang telah ada, seperti sifat tidak kelihatan (*invisibility*) yang jelek, *robustness* yang tidak sempurna pada beberapa rutin dari proses sinyal. Penyebab utamanya adalah karena kebanyakan algoritma menggunakan parameter kuantisasi tertentu tanpa

mempertimbangkan perbedaan diantara citra. Lebih lanjut lagi, ketika informasi *watermark* ditempelkan, kebanyakan algoritma tidak mempertimbangkan karakteristik baru yang dibawa oleh citra *carrier* oleh sebuah proses sinyal biasa, sehingga ketika *watermark* diekstraksi, *robustness* tidak dapat mencapai maksimum. Ataupun dapat dikatakan bahwa mereka tidak menempelkan informasi *watermark* berdasarkan pada karakteristik penyerangan. Untuk menyelesaikan permasalahan yang dihadapi, pada tahun 2009, Shengbing Che, Bin Ma, Jinkai Luo dan Shaojun Yu dari China memperkenalkan sebuah algoritma *image watermarking* yang berdasarkan pada segmentasi *region* [1].

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka yang menjadi permasalahan adalah:

1. Adanya masalah pada proteksi hak cipta citra digital, sehingga perlu diterapkan algoritma *watermarking* untuk menyembunyikan label hak cipta dalam citra sehingga dapat dijadikan sebagai bukti otentik kepemilikan karya digital tersebut.
2. Penerapan algoritma *watermarking* sering menghadapi beberapa masalah seperti kebanyakan algoritma menggunakan parameter kuantisasi tertentu tanpa mempertimbangkan perbedaan diantara citra dan tidak mempertimbangkan karakteristik baru yang dibawa oleh citra *carrier* oleh sebuah proses sinyal biasa, sehingga kelihatan perbedaan warna pada citra hasil *watermarking*.

1.3 Tujuan dan Manfaat

Tujuan dari penulisan ini adalah untuk membangun sebuah aplikasi yang menerapkan algoritma *semi-fragile watermarking* untuk menghasilkan citra *watermarking* untuk menyediakan proteksi hak cipta citra digital.

Manfaat dari penulisan ini, yaitu:

a. Aplikasi dapat digunakan untuk melindungi hak cipta citra dengan menyembunyikan citra *black and white* unik dalam sebuah citra digital.
b. Aplikasi dapat digunakan sebagai referensi dalam pengembangan aplikasi *watermarking* citra lainnya.

1.4 Batasan Masalah

Ruang lingkup pembahasan mencakup:

1. Data yang disisipkan berupa citra *black and white*, dimana ukuran citra *black and white* harus setengah dari ukuran citra sampul.
2. Citra masukan (citra sampul) dalam format BMP dan JPEG.
3. Format citra *watermark* (citra hasil) sesuai dengan format citra masukan (citra sampul).
4. Ukuran citra yang dapat diproses dengan batasan minimal 50 x 50 piksel dan maksimal sebesar 800 x 800.
5. Metode *scrambling* yang digunakan adalah metode pengacakan citra berdasarkan pada teori *chaos* dan transformasi pengurutan.

1.5 Metodologi Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Kajian Teoritis
   Mencari dan membaca buku teks serta bahan-bahan lainnya yang diperlukan.
2. Analisis
   Pada proses ini, dideskripsikan hasil analisis cara kerja sistem dalam bentuk *activity diagram*. Kemudian, proses dilanjutkan dengan memodelkan sistem dengan menggunakan *use case diagram*.
3. Perancangan

Membuat rancangan *interface* (masukan dan keluaran dari perangkat lunak).

4. Konstruksi Sistem
   Membangun perangkat lunak dengan bahasa pemrograman MS Visual Basic 2010.
5. Pengujian Data
   Pengujian yang dilakukan:
   a. Kecepatan proses penempelan dan ekstraksi dengan ukuran citra sampul dan *watermark* yang berbeda beda.
   b. Pemberian efek *brightness* dan *contrast* terhadap citra hasil *watermarking*.
   c. Pemberian *noise* pada bagian citra hasil *watermarking*.
   d. Menggunakan nilai kunci yang berbeda saat penempelan dan ekstraksi *watermark*.

2. Kajian Pustaka

2.1 *Digital Watermarking*

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). Ide *watermarking* pada data *digital* (sehingga disebut *digital watermarking)* dikembangkan di J*epang* tahun 1990 dan di *Swiss* tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti *video,* citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan. Hal yang memisahkan *watermarking* dari *steganografi* adalah dalam implementasinya, *steganografi* digunakan untuk mengamankan informasi yang ditumpangkan pada suatu media digital, sedangkan *watermarking* dapat dimanfaatkan untuk berbagai tujuan seperti:

1. ***Tamper-proofing***: *watermarking* digunakan sebagai indikator yang menunjukkan ada tidaknya perubahan pada data yang di*watermarking*.
2. ***Feature location***: menggunakan metode *watermarking* sebagai alat untuk mengidentifikasi isi dari data *digital* pada lokasi-lokasi tertentu, seperti contohnya penamaan objek tertentu dari beberapa objek yang lain pada suatu citra *digital*.
3. ***Annotation/caption***: *watermarking* yang digunakan sebagai keterangan tentang data *digital* itu sendiri atau informasi lain yang dipandang perlu untuk ditanamkan kedalam media yang bersangkutan.
4. ***Secure and invisible communications*** atau komunikasi yang aman.
5. ***Copyright-Labeling***: *watermarking* dapat digunakan sebagai metode untuk penyembunyian label hak cipta pada data *digital* sebagai bukti otentik kepemilikan karya *digital* tersebut.

Watermarking dapat juga dikategorikan sebagai *visible watermarking* (watermark terlihat oleh indera manusia) dan *invisible watermarking* (*watermark* tidak tampak). *Watermarking* dapat juga dikategorikan sebagai *blind watermarking* (proses verifikasi watermark tidak membutuhkan citra asal) dan *non blind watermarking* (proses verifikasi watermark membutuhkan citra asal). [2]

2.2 Algoritma *Scrambling* dan *Unscrambling*

Berikut algoritma *scrambling*:

Langkah 1: Diberikan sebuah nilai awal x_1 yang diasosiasikan dengan kunci rahasia. Anggap k = 1.

Langkah 2: Iterasikan sebanyak N – 1 kali dengan iterasi *chaotic* (1), dengan mengambil deretan dari nilai riil $\{x_1, x_2, .., x_N\}$

Langkah 3: Urutkan deretan$\{x_1, x_2, .., x_N\}$ dari kecil ke besar, sehingga diperoleh deretan terurut $\{x_1' x_2', \ldots, x_N'\}$

Langkah 4: Hitung kode alamat dari kumpulan *scrambling* dimana $\{t_1, t_2, \ldots, t_N\}$, dimana $t_i \in \{t_1, t_2, \ldots, t_N\}$. t_i adalah *subscript* baru dari x_i pada deretan terurut $\{x_1', x_2', \ldots, x_N'\}$.

Langkah 5: Permutasikan baris ke –k dari citra I dengan kode alamat permutasi $\{t_1, t_2, \ldots, t_N\}$, dengan mengganti piksel pada kolom ke t_i dengan piksel kolom ke i untuk i = 1 sampai N.

Langkah 6: Jika k = M, maka proses telah selesai. Jika tidak, anggap $x_1 = x_N$ dan k = k + 1. Ulangi langkah 2 sampai langkah 5.

Berikut algoritma *unscrambling*: dengan diberikan sebuah nilai awal x_1, prosedur dekripsi sama seperti dengan *scrambling*, kecuali langkah 5 yang diubah menjadi:
Tukar piksel kolom ke-i dengan piksel kolom ke-t_i untuk i dari 1 sampai N. [3]

2.3 Algoritma *Semi-Fragile Image Watermarking* Berdasarkan Segmentasi *Region*

Berdasarkan ciri visual manusia Shengbing Che, Bin Ma, Jinkai Luo dan Shaojun Yu dari China (2009) memperkenalkan sebuah model baru dari ciri visual dan menggunakan *Quantized Central Limit Theorem*, operator *coefficient redressal*, probabilitas pengembalian terbaik dan metode penempelan dua langkah. Berdasarkan teorema ini, para ahli tersebut menggunakan pendekatan kuantisasi dinamik untuk menempelkan *watermark* pada transformasi *wavelet*.

Pengujian yang dilakukan oleh para ahli tersebut menunjukkan ketidaknampakan yang lebih bagus dari citra pembawa dan ketangguhan yang lebih baik pada pemrosesan citra seperti kompresi JPEG, penambahan noise dan filter penghalusan dapat diperoleh. Sementara itu, informasi yang ditempelkan dalam skala yang besar, jumlah bit data mencapai seperempat dari piksel citra asli.

Metode segmentasi region, seperti ekstraksi sisi dan metode *threshold gray* adalah metode yang berdasarkan pada domain jarak (*space domain*). Metode tersebut hanya mengamati perbedaan dari nilai piksel dan tidak mempertimbangkan perubahan dari nilai tersebut. Oleh karena itu, metode ini akan memecahkan banyak titik piksel yang sebenarnya tidak perlu dipecahkan keluar dari area halus. Metode segmentasi region yang diperkenalkan oleh Shengbing Che, Bin Ma, Jinkai Luo dan Shojun Yu berdasarkan pada domain frekuensi, dengan mempertimbangkan perubahan pada tiga arah, yaitu horizontal, vertikal dan diagonal [1].

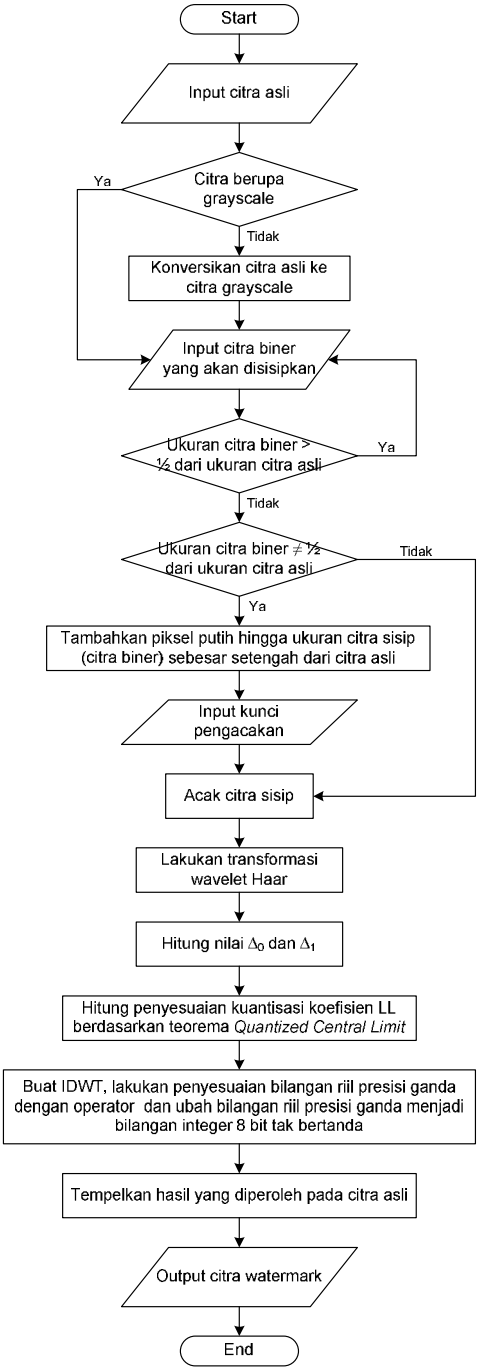
3 Metode Penelitian

Proses kerja dari algoritma *Semi-Fragile Image Watermarking* berdasarkan pada *Region Segmentation* dapat dibagi menjadi dua tahapan, yaitu:

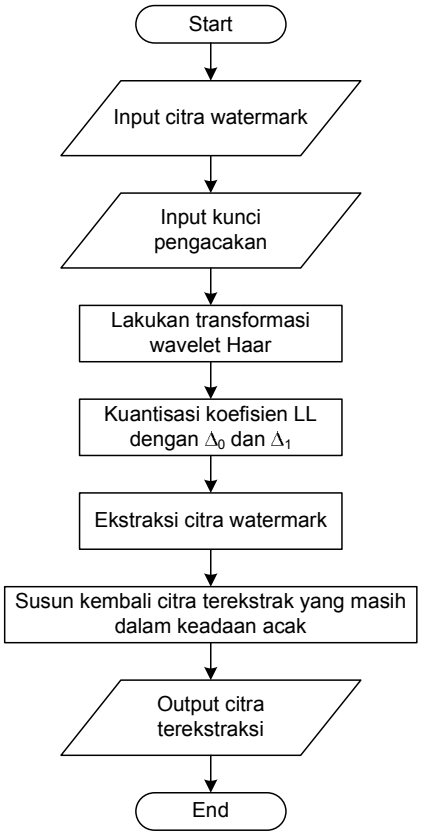
1. Proses Pembuatan dan Penempelan *Watermark*
   Proses ini berfungsi untuk menempelkan citra biner ke dalam citra sampul (berbentuk citra *grayscale*).
2. Proses Ekstraksi *Watermark*

Proses ini berfungsi untuk mengekstrak keluar citra biner yang disisipkan sebelumnya.

Gambar 1 menunjukkan proses pembuatan *watermark*. Gambar 2 menunjukkan proses ekstraksi citra biner terhadap citra *watermark* yang dihasilkan.



Gambar 1 Proses Pembuatan *Watermark*



Gambar 2 Proses Ekstraksi *Watermark*

4 Hasil dan Pembahasan

4.1 Hasil

Tampilan aplikasi untuk pembuatan *watermark* bisa dilihat pada Gambar 3.

Gambar 3. Tampilan Aplikasi Pembuatan *Watermark*

Langkah pertama yang harus dilakukan adalah memilih citra sampul yang akan disisipkan citra hitam putih didalamnya. Caranya adalah dengan mengklik tombol '+' sehingga sistem akan menampilkan kotak dialog *Open*. Pilih *file* citra yang diinginkan dan klik tombol 'Open'. Sistem akan membaca dan menampilkan citra yang dipilih. Setelah itu, proses akan dilanjutkan dengan pemilihan citra hitam putih yang ingin disisipkan ke dalam citra asli. Setelah itu, proses dilanjutkan mengisi nilai kunci acak dan mengklik tombol 'Acak'. Kemudian, klik tombol 'Konversi ke *Grayscale*' untuk mengkonversi citra asli ke bentuk *grayscale*. Setelah selesai melakukan pengisian data, maka kliklah tombol 'Proses' sehingga sistem akan menempelkan citra hitam putih ke dalam citra asli *grayscale*.

Tampilan aplikasi untuk ekstraksi *watermark* bisa dilihat pada Gambar 4.

Gambar 4. Tampilan Aplikasi Ekstraksi *Watermark*

Langkah yang harus dilakukan adalah memilih citra *watermarking* untuk mengekstraksi citra hitam putih. Setelah itu, masukkan kunci *anti-scrambling*. Setelah selesai mengisi semua data yang diperlukan, klik tombol 'Proses'. Sistem akan menampilkan citra terekstrak keluar, citra yang disisipkan dan lama waktu eksekusi yang diperlukan.

4.2 Pengujian

Pengujian 1:

Pengujian dilakukan untuk mengamati kecepatan proses penempelan dan ekstraksi dengan ukuran citra asli dan citra *watermark* yang berbeda-beda seperti yang terlihat pada Tabel 1.

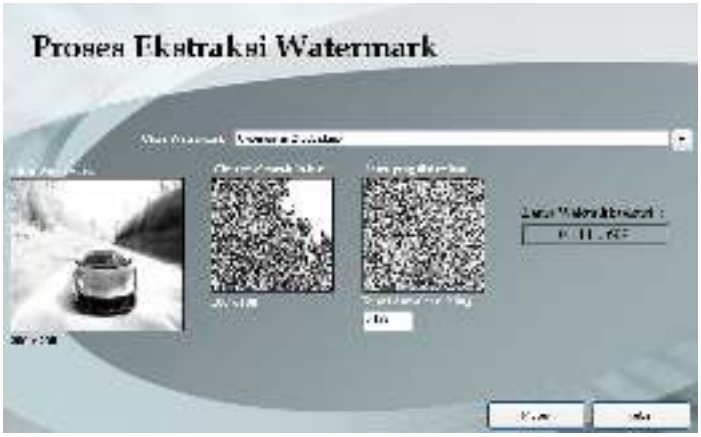
Tabel 1. Hasil Pengujian Kecepatan Proses Penempelan dan Ekstraksi dengan Ukuran Citra Sampul dan Citra *Watermark* yang Berbeda-beda

Ukuran Citra Asli	Ukuran Citra Watermark	Key	Waktu Penempelan	Waktu Ekstrasi
80 x 80	40 x 40	0.8888	0.93 detik	0.265 detik
120 x120	40 x 40	0.123	0.250 detik	0.859 detik
140 x 140	40 x 40	0.678	0.296 detik	1.593 detik
160 x 160	50 x 50	0.9	0.421 detik	2.375 detik
180 x 180	90 x 90	0.95	0.765 detik	6.62 detik
200 x 200	80 x 80	0.111	0.875 detik	5.609 detik
200 x 200	80 x 80	0.43	0.937 detik	5.609 detik
220 x 220	90 x 90	0.78	0.828 detik	8.140 detik
300 x 300	110 x 110	0.8	2.265 detik	30.171 detik
320 x 320	120 x 120	0.941	3.531 detik	48.953 detik
380 x 380	120 x 120	0.3333	6.31 detik	1 menit 47.234 detik
380 x 380	140 x 140	0.59	5.171 detik	1 menit 12.671 detik
400 x 400	180 x 180	0.3	8.531 detik	2 menit 7.437 detik
480 x 480	220 x 220	0.159	15.78 detik	3 menit 22.140 detik
500 x 500	250 x 250	0.1	15.828 detik	3 menit 58.359 detik
550 x 550	240 x 240	0.851	20.312 detik	5 menit 40.921 detik
600 x 600	280 x 280	0.39	28.468 detik	8 menit 1.890 detik
650 x 650	300 x 300	0.279	41.484 detik	11 menit 25.640 detik
700 x 700	80 x 80	0.8	52.312 detik	14 menit 58.203 detik
750 x 750	170 x 170	0.75	1 menit 7.578 detik	18 menit 48.203 detik

Pengujian 2 :

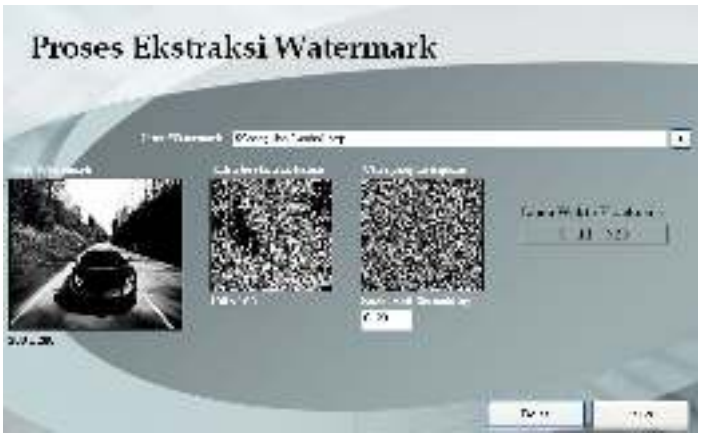
Pengujian dilakukan dengan mengubah *brightness* dan *contrast* dari citra hasil dan dilakukan proses ekstraksi untuk melihat pengaruhnya terhadap citra hasil ekstrak.

Hasil ekstraksi apabila citra hasil diberi efek *brightness*, seperti terlihat pada Gambar 5.



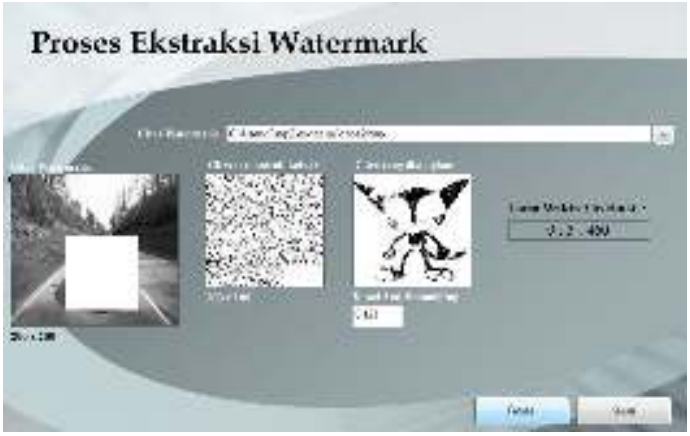
Gambar 5. Hasil Ekstraksi Citra Hasil Diberi Efek *Brightness*

Hasil ekstraksi apabila citra hasil diberi efek *contrast*, seperti terlihat pada Gambar 6:



Gambar 6. Hasil Ekstraksi Citra Hasil Diberi Efek *Contrast*

Pengujian 3:
Pengujian dengan proses *crop* dan pemberian *noise* terhadap bagian tertentu pada citra hasil. Hasil ekstraksi apabila citra hasil yang telah di*crop* pada bagian tengah, seperti terlihat pada Gambar 7:



Gambar 7. Hasil Ekstraksi Citra Hasil Di*crop* pada Bagian Tengah

Hasil ekstraksi apabila citra hasil yang telah di*crop* pada bagian sekeliling sisi, seperti terlihat pada Gambar 8:

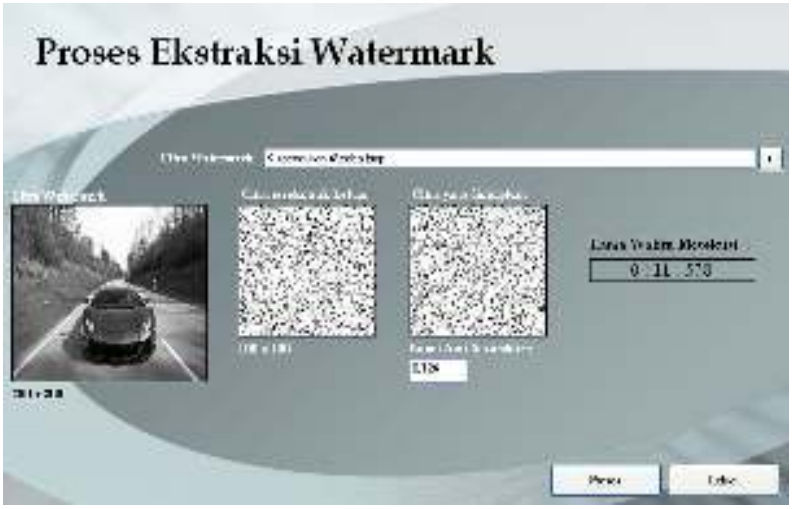
Gambar 8. Hasil Ekstraksi Citra Hasil Di*crop* pada Bagian Sekeliling Sisi

Hasil ekstraksi apabila citra hasil yang telah diberi *noise* terlihat pada Gambar 9:

Gambar 9. Hasil Ekstraksi Citra Hasil Diberi *Noise*

Pengujian 4 :

Pengujian terhadap kesalahan pengisian kunci. Apabila terjadi kesalahan dalam pengisian nilai kunci yang dimasukkan, misalkan nilai 0.123 diganti menjadi 0.124, maka hasil ekstraksi menjadi terlihat seperti Gambar 10:



Gambar 10. Hasil Ekstraksi Citra Jika Terjadi Kesalahan Pengisian Kunci

5 Kesimpulan

Setelah melakukan pengujian, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Waktu eksekusi dari proses penempelan *watermark* relatif lebih cepat dari proses ekstraksi *watermark*.
2. Algoritma tidak bisa mengekstraksi citra hitam putih yang ketika efek *brightness* dan *contrast* diberikan pada citra hasil *watermark*.
3. Citra hitam putih masih dapat terekstrak keluar (namun tidak sempurna) walaupun citra *watermark* telah disisipkan *noise* ataupun dilakukan proses *crop* disisipkan terhadap citra *watermark*.
4. Algoritma yang digunakan bersifat sensitif terhadap kunci yang digunakan. Apabila ada kesalahan pengisian kunci maka akan menyebabkan citra hitam putih menjadi tidak terekstrak keluar.

Referensi

[1] Che, S., Ma, B., Luo, J. dam Yu, S. 2009. *Semi-Fragile Image Watermarking Algorithm Based on Region-Segmentation*, I.J Image, Graphics and Signal Processing, 1, 59-66, China.

[2] Yullinda, C.D. 2008. Implementasi *Watermarking* dengan Metode *Discrete Cosine Transform* (DCT) pada Citra Digital, Digital Library-Perpustakaan Pusat UNIKOM.

[3] Xiangdong, L., Z. Junxing, Z. Jinhai dan H. Xiqin. 2008. *Image Scrambling Algorithm Based on Chaos Theory and Sorting Transformation*, IJCSNS International Journal of Computer Sicence and Network Security, Vol 8 No.1.