



Watermarking pada Citra Warna Menggunakan Teknik SVD – DCT Berdasarkan Local Peak SNR

Daniel Setiadikarunia dan Frederick Michael

Jurusan Teknik Elektro, Universitas Kristen Maranatha, Bandung

Jl. Suria Sumantri 65, Bandung 40164, Indonesia

daniel.sk@ieee.org

Abstrak: Perkembangan teknologi terutama pada dunia digital pada saat ini memungkinkan informasi dalam berbagai bentuk dan media dapat tersebar dengan cepat tanpa batas ruang dan waktu serta rentan terhadap penggunaan secara ilegal. Salah satu solusi untuk mencegah penggunaan secara ilegal adalah dengan *watermarking*. *Watermarking* merupakan penyisipan tanda (*watermark*) ke dalam data digital yang bertujuan sebagai tanda identitas pemilik asli data digital tersebut. Dalam makalah ini dibahas penyisipan *watermark* pada citra digital menggunakan teknik *Singular Value Decomposition (SVD) – Discrete Cosine Transform (DCT)* berdasarkan *Local Peak Signal to Noise Ratio (LPSNR)* serta pengujian kualitas citra yang telah disisipi *watermark* dan ketahanan dari *watermark*. Proses penyisipan *watermark* dilakukan dengan menggabungkan nilai *singular* hasil SVD pada *watermark* ke dalam koefisien DCT dari citra asli. Untuk mendapat tingkat transparansi dan ketangguhan yang optimal dari *watermark*, digunakan LPSNR. Selanjutnya dilakukan inversi transformasi DCT untuk mendapatkan citra yang telah disisipi *watermark*. Dari hasil uji coba didapatkan bahwa, citra yang telah disisipi *watermark* memiliki kualitas yang baik dan *watermark* memiliki ketahanan yang kuat terhadap kompresi JPEG, *cropping*, *scaling* (diperbesar), dan *filtering*, tetapi tidak tahan terhadap proses *scaling* (diperkecil) dan rotasi.

Kata kunci: *Watermarking*, SVD, DCT, LPSNR

Abstract: Today technology development especially in digital world enable information in various forms and media can be disseminated rapidly without time and space boundaries and vulnerable to illegal use. One of solutions that can be used to avoid the illegal use is *watermarking*. *Watermarking* is the insertion of mark into a digital image with the aim to give a sign of original owner identity. In this paper, we discussed the insertion of *watermark* into digital image using *Singular Value Decomposition (SVD) – Discrete Cosine Transform (DCT)* technique based on *Local Peak Signal to Noise Ratio (LPSNR)*, and examination the quality of image that has been inserted *watermark* and the robustness of *watermark*. The *watermark* insertion process is performed by embedding the singular values from the SVD of *watermark* into DCT coefficient of original image. LPSNR is used to get transparency level and optimal robustness of *watermark*. Then inverse DCT transformation is performed to get image that has been *watermark* inserted. The results from the experiment showed that image quality which has been inserted by *watermark* has good quality and *watermark* has strong robustness to

JPEG compression, cropping process, scaling process (enlarged), and filtering process, but is not resist to scaling (reduced) process and rotate process.

Keywords: *Watermarking, SVD, DCT, LPSNR*

I. PENDAHULUAN

Dalam era digital saat ini, informasi dalam berbagai bentuk dan media dapat tersebar dengan cepat tanpa batas ruang dan waktu. Informasi dalam bentuk multimedia ini rentan terhadap perubahan dan memberikan kesempatan kepada pihak yang tidak berhak untuk melakukan duplikasi dan modifikasi data tanpa izin dari pemilik yang sah untuk berbagai kepentingan. Hal ini dapat menimbulkan persoalan hak cipta bagi data multimedia yang tersebar. Salah satu cara yang dapat digunakan untuk melindungi hak cipta pada data multimedia adalah dengan *watermarking*.

Watermarking merupakan teknik penyisipan data rahasia ke dalam sebuah sumber asli. Sumber yang dimaksud dapat berupa teks, gambar, suara dan video. Data yang telah disisipkan, harus dapat dideteksi atau diekstrak kembali. *Watermarking* dapat digolongkan menjadi dua bagian berdasarkan domain kerjanya, yaitu pada domain spasial dan domain transformasi. *Watermarking* yang bekerja dalam domain spasial bekerja dengan cara menanamkan *watermark* secara langsung ke dalam domain spasial dari suatu citra. Istilah domain spasial sendiri mengacu pada piksel-piksel penyusun sebuah citra. Pada teknik *watermarking* dalam domain transformasi, penanaman *watermark* dilakukan pada koefisien frekuensi hasil transformasi citra asalnya.

Algoritma pemberian/penyisipan *watermark Least Significant Bit (LSB) Coding*^[1] merupakan metode *watermarking* pada domain spasial. Metode ini sangat sederhana tetapi yang paling tidak tahan terhadap segala proses yang dapat mengubah nilai-nilai intensitas pada citra, terutama kompresi JPEG. Penyisipan *watermarking* dengan algoritma ini paling mudah diserang, karena data label akan hilang seluruhnya bila nilai dari LSB-nya dibalikkan. Metode ini mengubah nilai LSB (*Least Significant Bit*) komponen luminansi atau warna menjadi bit label yang akan disembunyikan. Hasil dari citra yang telah disisipi *watermark* sangat mirip dengan aslinya, karena hanya mengubah nilai bit terakhir dari data. Metode lain yang menggunakan label satu bit juga, yaitu metode *Patchwork*^[2]. Metode *Patchwork* didasarkan pada penanaman label 1 bit pada citra digital dengan menggunakan pendekatan statistik. Sebanyak n pasang titik (a_i, b_i) pada citra dipilih secara acak. *Brightness* dari a_i dinaikkan 1 (satu) dan *brightness* dari pasangannya b_i diturunkan 1 (satu). Nilai Harapan dari jumlah perbedaan n pasang titik tersebut adalah $2n$. Ketahanan metode ini terhadap kompresi JPEG dengan parameter kualitas 75% adalah label masih dapat dibaca dengan probabilitas kebenaran sebesar 85%.

Metode *Secure Spread Spectrum Watermarking for Multimedia*^[3] diperkenalkan oleh Ingemar J. Cox. Metode ini didasarkan pada domain frekuensi, dengan menanamkan sejumlah urutan bilangan riil sepanjang n pada citra $N \times N$ dengan mentransformasikan terlebih dahulu menjadi koefisien DCT $N \times N$. Bilangan tersebut ditanamkan pada n koefisien DCT yang paling besar, tidak termasuk komponen DC.

Zhao dan Koch mengusulkan *Randomly Sequenced Pulse Position Modulated Code (RSPPMC)*^[4] yang bekerja pada domain DCT seperti metode Cox. Namun, metode ini bekerja berdasarkan prinsip format citra JPEG, membagi citra menjadi blok-blok 8×8 , kemudian dilakukan transformasi DCT. Dengan menggunakan prinsip *spread spectrum* (metode *frequency*

hopped) dan RSPPMC, koefisien-koefisien DCT tersebut diubah sedemikian rupa sehingga akan mengandung informasi 1 bit dari label, seperti dipilih tiga koefisien untuk disesuaikan dengan bit label yang ingin ditanamkan. Contohnya untuk menanamkan bit '1' ke dalam suatu blok koefisien DCT 8×8 , koefisien ketiga dari ketiga koefisien yang terpilih harus diubah sedemikian rupa sehingga lebih kecil dari kedua koefisien lainnya.

Chandra dkk.^[5] mengusulkan algoritma berbasis SVD yang diterapkan pada citra dan pada *watermark*. Nilai *singular* dari *watermark* dikalikan dengan faktor *scaling* dan ditambahkan pada nilai *singular* dari citra. Metode ini tidak tergolong ke dalam *blind watermarking* karena membutuhkan citra asli dan citra yang telah disisipi *watermark* untuk melakukan ekstraksi *watermark*. Pada tahun 2002, Sun dkk.^[6] mengusulkan skema *watermarking* berbasis SVD dan kuantisasi dengan mengeksplorasi matriks nilai *singular* untuk menyisipkan *watermark*. Mekanisme dasar yang digunakan adalah kuantisasi koefisien terbesar pada matriks nilai *singular* dengan sebuah nilai konstan yang disebut koefisien kuantisasi. Terdapat suatu *trade-off* antara *invisibility* (ketidaktampakan) dan *robustness* (ketahanan) *watermark*. Bila diinginkan *robustness* yang tinggi maka akan semakin *visible*, dan sebaliknya semakin *invisible* maka *robustness* akan semakin menurun. Hasil terbaik yang diharapkan dapat dicapai dengan mengubah-ubah koefisien kuantisasi. Liu dan Tan^[7] mengaplikasikan transformasi SVD ke seluruh citra. *Watermark* berupa matriks *pseudo gaussian random number* dengan faktor *scaling* yang tepat ditambahkan pada matriks nilai *singular*. Kemudian matriks nilai *singular* yang telah dimodifikasi dimasukkan kembali pada citra.

Permasalahan yang dibahas dalam artikel ini bagaimana menyisipkan *watermark* pada citra digital berwarna (RGB) menggunakan teknik *Singular Value Decomposition – Discrete Cosine Transform* berdasarkan *Local Peak Signal to Noise Ratio (LPSNR)*. Bagaimana kualitas citra yang telah disisipi *watermark* dan bagaimana ketahanan *watermark* terhadap pengolahan citra yang umum dilakukan.

II. METODE WATERMARKING

Penyisipan *watermark* yang dibuat termasuk dalam teknik *watermarking* dalam domain frekuensi. *Watermark* yang akan disisipkan, pertama-tama dilakukan transformasi SVD. Transformasi atau dekomposisi SVD pada *watermark* \mathbf{W} yang berukuran $M \times M$ dapat dituliskan sebagai^[8]

$$\begin{aligned} \mathbf{W} &= \mathbf{U}\mathbf{S}\mathbf{V}^T \\ &= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_M] \times \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_M \end{bmatrix} \times [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_M]^T \\ &= \sum_{i=1}^M \lambda_i \mathbf{u}_i \mathbf{v}_i^T \end{aligned} \quad (1)$$

dengan $\mathbf{S} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$ adalah matriks diagonal $M \times M$, yang mempunyai elemen diagonal non negatif, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M \geq 0$. λ_i adalah nilai *singular* dari matriks \mathbf{W} . \mathbf{U} adalah matriks $M \times M$

yang kolomnya terdiri dari vektor *singular* kiri \mathbf{u}_i dari matriks \mathbf{W} . \mathbf{V} adalah matriks $M \times M$ yang kolomnya terdiri dari vektor *singular* kanan \mathbf{v}_i dari matriks \mathbf{W} . \mathbf{U} dan \mathbf{V} adalah matriks unitari, yang mempunyai sifat $\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$, dan $\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}$, dengan \mathbf{I} adalah matriks identitas $M \times M$.

Nilai *singular* hasil SVD dari *watermark* inilah yang akan disisipkan pada citra asli/semula. Citra asli dibagi ke dalam blok-blok berukuran 8×8 piksel yang tidak saling overlap. Pada blok citra ini dilakukan proses DCT menggunakan persamaan^[9]

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right], \quad (2)$$

dengan

$$\alpha(u) = \alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{untuk } u = v = 0 \\ \sqrt{\frac{2}{N}} & \text{untuk } u, v \neq 0 \end{cases}$$

Nilai *singular* disisipkan ke dalam koefisien DCT $C(u, v)$ dari blok 8×8 yang terpilih dari citra yang akan diberi *watermark*. Penentuan blok 8×8 mana yang dipilih untuk disisipi nilai *singular* dari *watermark* dilakukan berdasarkan persamaan *chaotic sequence* berikut.^[10]

$$X_{n+1} = \mu X_n (1 - X_n) \quad (3)$$

dengan $\mu \in [0, 4]$ dan dipilih $X_0 \in (0, 1)$.

Untuk mengurangi blok artefak dalam gambar yang telah disisipi *watermark*, digunakan LPSNR. Dengan pemilihan nilai LPSNR yang tepat, maka akan didapat tingkat transparansi yang baik dan ketangguhan maksimal dari *watermark* terhadap distorsi. LPSNR didefinisikan sebagai^[11]

$$\text{LPSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{8} \sum_{x=1}^8 \sum_{y=1}^8 [I'(x, y) - I(x, y)]^2} \quad (4)$$

dengan $I'(x, y)$ adalah nilai intensitas piksel blok citra yang sudah disisipi *watermark* dan $I(x, y)$ adalah nilai intensitas piksel blok citra asli pada posisi (x, y) dalam domain spasial.

II.1. Penyisipan Watermark

Diagram blok proses penyisipan *watermark* ditunjukkan pada Gambar 1. Prinsip kerjanya dapat dijelaskan sebagai berikut:

1. Mula-mula dilakukan transformasi SVD pada *watermark*, sehingga didapat *singular value* (λ) dari *watermark*.
2. Citra asli/semula dibagi menjadi blok berukuran 8×8 piksel. Bila citra asli berukuran 512×512 piksel, maka akan dibagi menjadi 64×64 blok berukuran 8×8 piksel.
3. Blok-blok citra asli yang telah dibagi, dipilih dengan *chaotic sequence*.
4. Kemudian blok-blok yang telah dipilih dilakukan DCT.

5. Nilai *singular* dari *watermark* disisipkan pada koefisien DCT dari blok-blok terpilih berdasarkan,

$$f'(1,1) = f(1,1) + (m \cdot \lambda) \quad (5)$$

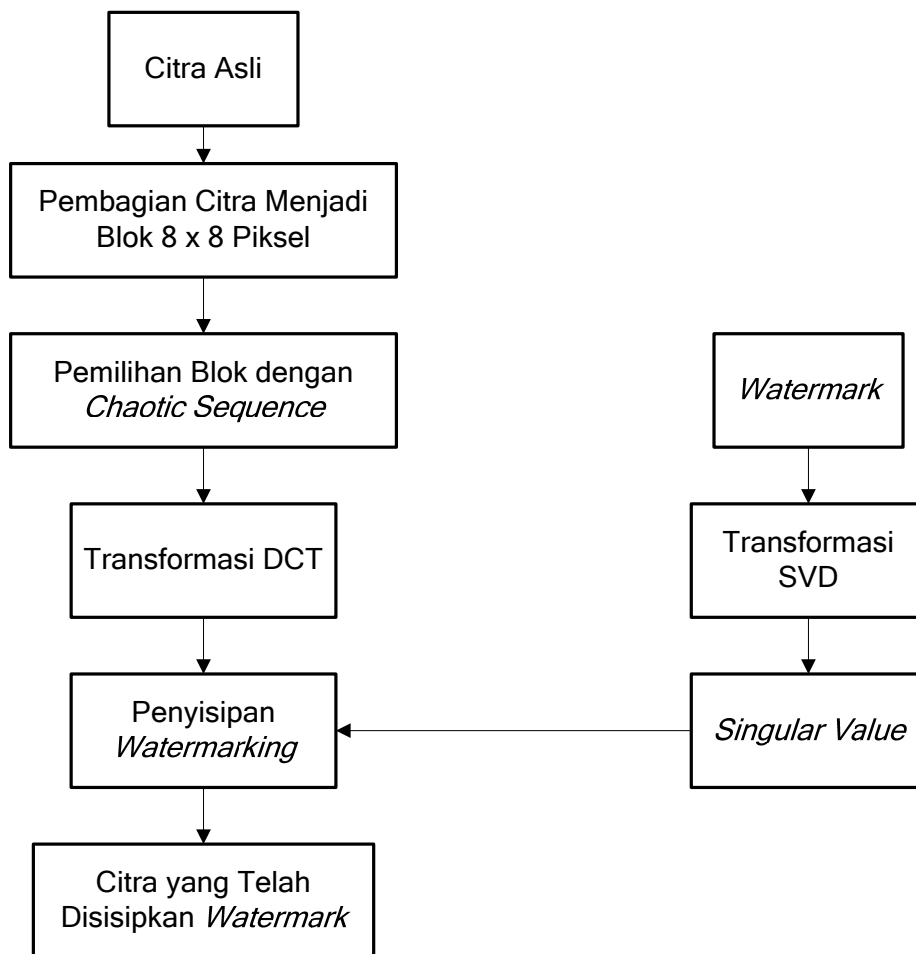
dengan

$f(1,1)$ = komponen DC dari matriks citra hasil DCT

m = faktor pengali untuk nilai *singular*

λ = nilai *singular*

6. Kemudian dilakukan inversi DCT untuk mendapatkan citra yang telah disisipi *watermark*.



Gambar 1. Diagram Blok Penyisipan *Watermark*

II.2. Ekstraksi *Watermark*

Diagram blok proses ekstraksi *watermark* ditunjukkan pada Gambar 2. Prinsip kerja dari proses ekstraksi *watermark* dapat dijelaskan sebagai berikut:

1. Citra yang telah disisipi *watermark*, dibagi menjadi blok berukuran 8×8 piksel.
2. Citra asli/semula dibagi menjadi blok berukuran 8×8 piksel.
3. Blok citra yang telah disisipi *watermark* dan blok citra asli, dipilih dengan *chaotic sequence*.

4. Kemudian blok-blok yang telah dipilih dilakukan DCT dan diekstrak nilai *singularnya* berdasarkan,

$$\lambda' = [f'(1,1) - f(1,1)] / m \quad (6)$$

dengan

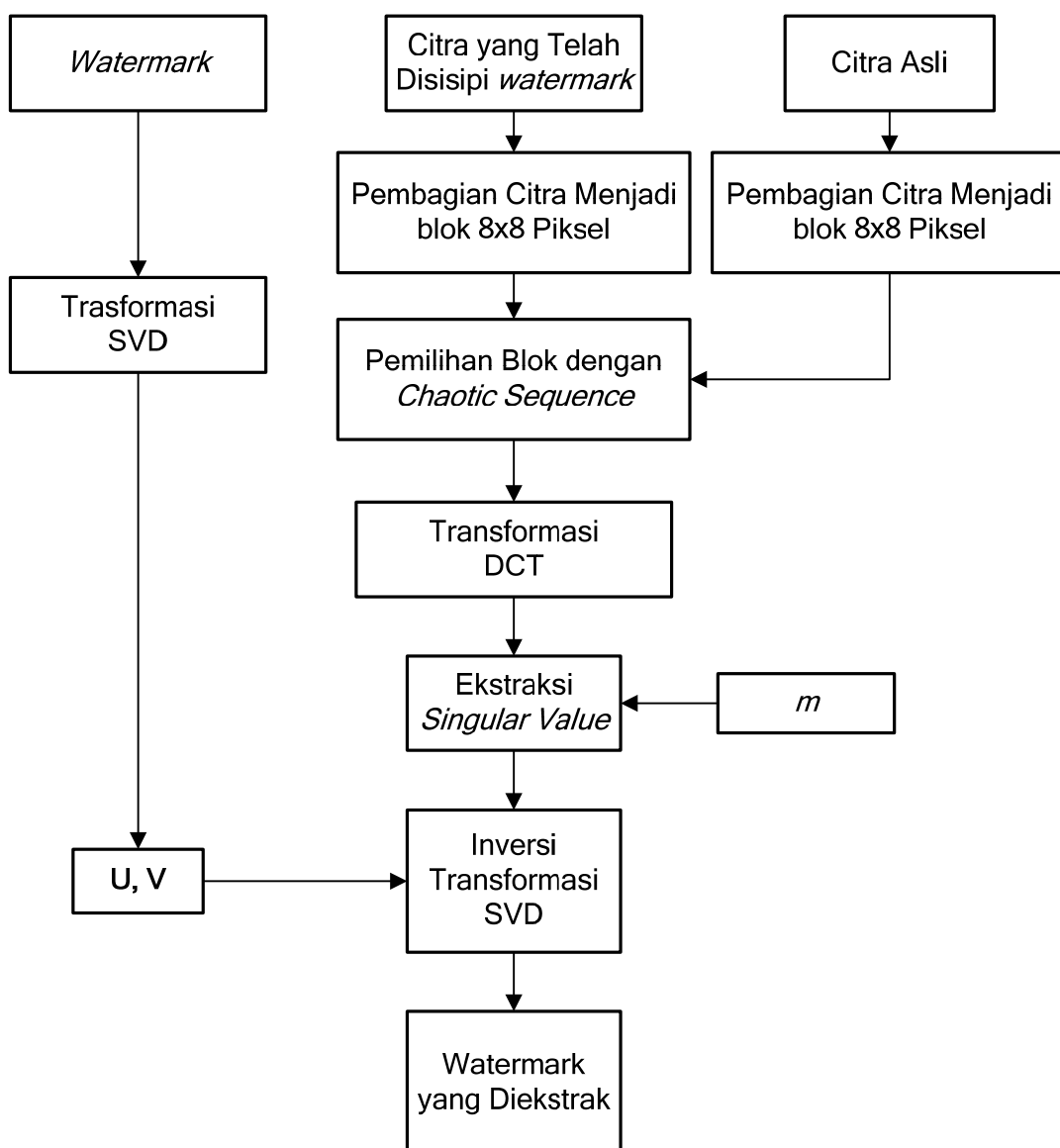
λ' = nilai *singular* hasil ekstraksi

$f'(1,1)$ = komponen DC dari matriks citra yang telah disisipi *watermark*

$f(1,1)$ = komponen DC dari matriks citra asli

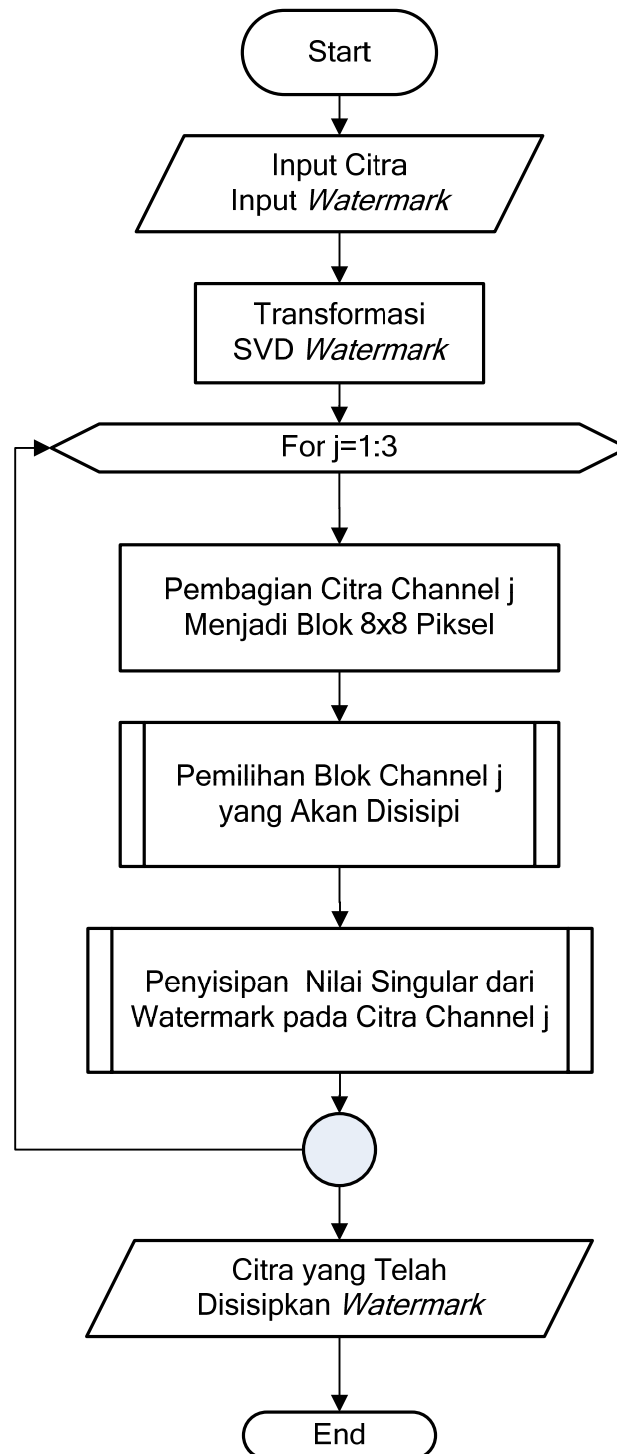
m = faktor pengali untuk nilai *singular*

5. Kemudian dilakukan transformasi SVD untuk mendapatkan matriks **U** dan **V**. Kemudian dilakukan inversi transformasi SVD dengan menggunakan matriks **U**, **V** dan *singular value* hasil ekstraksi untuk mendapatkan citra *watermark*.



Gambar 2. Diagram Blok Ekstraksi *Watermark*

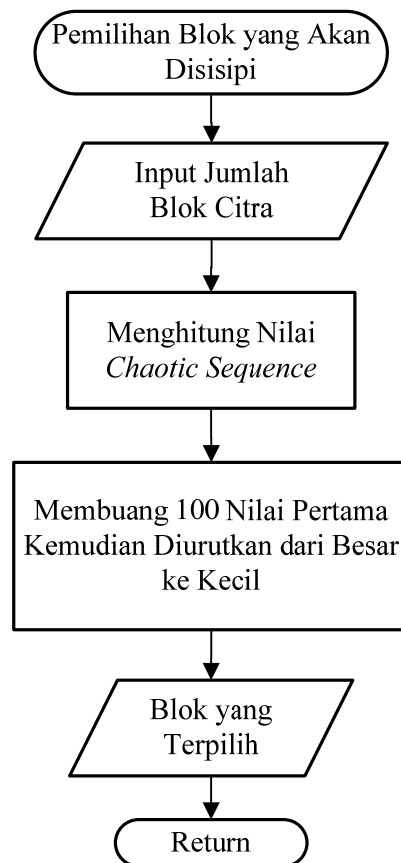
III. PERANCANGAN PERANGKAT LUNAK

III.1. Diagram Alir Penyisipan *Watermark*Gambar 3. Diagram Alir Penyisipan *Watermark*

Algoritma penyisipan *watermark* yang ditunjukkan pada Gambar 3 adalah sebagai berikut:

1. Mula-mula *watermark* ditransformasikan menggunakan SVD, sehingga didapat nilai *singular* dari *watermark*.
2. Memasukkan input j , dengan j didefinisikan sebagai penunjuk *channel* RGB dari citra. Citra asli dibagi menjadi blok-blok berukuran 8×8 piksel.
3. Kemudian blok-blok citra dipilih untuk disisipi *watermark* berdasarkan *chaotic sequence*.
4. Kemudian dilakukan proses penyisipan *watermark* pada blok-blok citra yang telah dipilih.
5. Proses pemilihan blok dan penyisipan *watermark* terus berulang sampai matriks RGB citra selesai disisipi dengan *watermark*. Setelah proses selesai, didapat citra yang telah disisipi *watermark*.

A. Diagram Alir Proses Pemilihan Blok yang Akan Disisipi



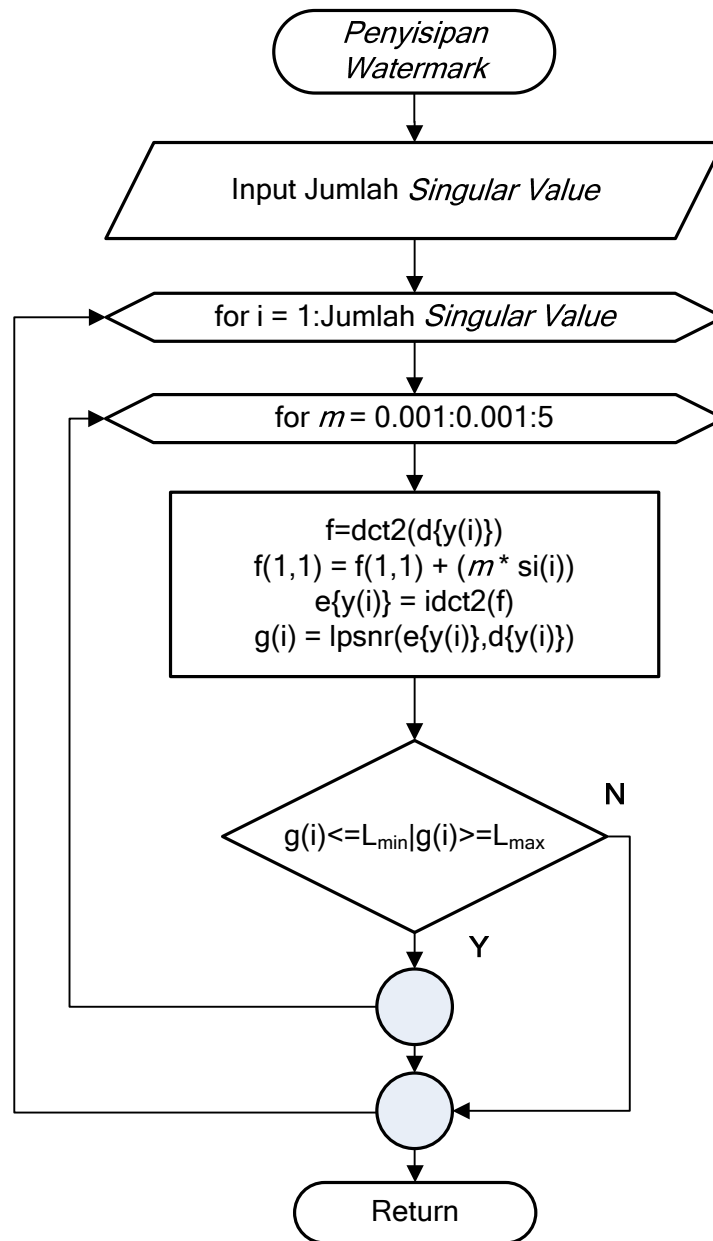
Gambar 4. Diagram Alir Proses Pemilihan Blok yang Akan Disisipi

Algoritma proses pemilihan blok yang akan disisipi seperti yang ditunjukkan pada Gambar 4 adalah sebagai berikut:

1. Memasukkan input jumlah blok citra yang berukuran 8×8 piksel ditambah 100. Bila citra yang digunakan 512×512 piksel, maka input jumlah blok adalah 4196.
2. Hitung nilai *chaotic sequence* berdasarkan persamaan (3)
3. 100 nilai pertama dibuang, kemudian diurutkan berdasarkan nilai *chaotic sequence* secara *descending*.

4. Setelah proses pengurutan selesai, didapat blok–blok citra yang akan disisipi *watermark*.

B. Diagram Alir Proses Penyisipan Nilai Singular



Gambar 5. Diagram Alir Proses Penyisipan Nilai Singular dari *Watermark*

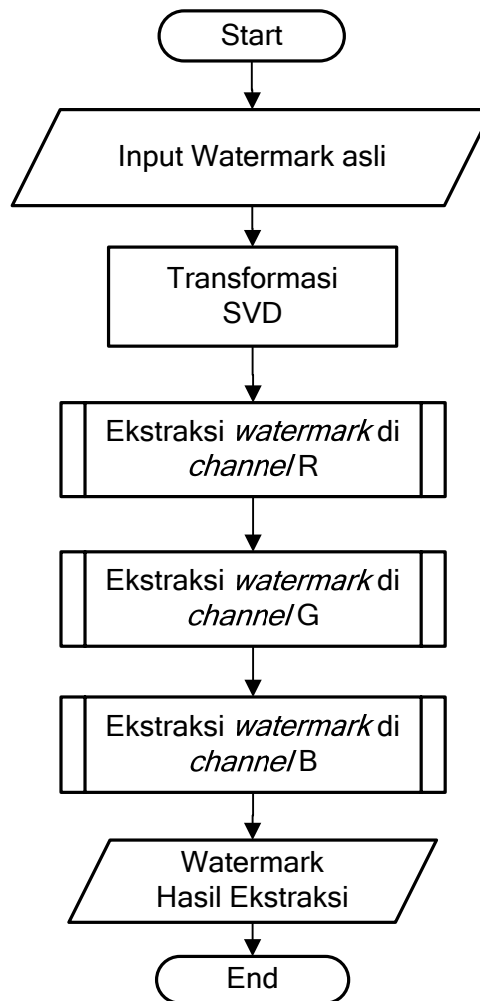
Algoritma proses penyisipan *watermark* seperti yang ditunjukkan pada Gambar 5 adalah sebagai berikut:

1. Memasukkan input jumlah *singular value* dari *watermark*. Memasukkan input m , dengan m adalah faktor pengali untuk *singular value*.
2. Kemudian blok citra yang terpilih dilakukan DCT. Matriks koefisien DCT dari blok yang terpilih akan dimodifikasi menurut *singular value* dari *watermark* yang akan disisipkan. Elemen matriks yang akan diubah adalah elemen baris pertama dari kolom

pertama (komponen DC). Modifikasi koefisien-koefisien tersebut dilakukan menurut persamaan (5). Blok citra yg telah diubah, dilakukan inversi DCT.

3. Kemudian hitung nilai LPSNR, jika nilai $LPSNR \leq L_{min}$ atau $LPSNR \geq L_{max}$ maka nilai m akan diubah dan dilakukan modifikasi kembali. Jika nilai LPSNR telah berada di dalam batas, maka nilai m yang terakhir digunakan, disimpan dan proses selesai.
4. Proses penyisipan *watermark* ini akan terus berulang hingga seluruh *singular value* disisipkan.

III.2. Diagram Alir Ekstraksi Watermark

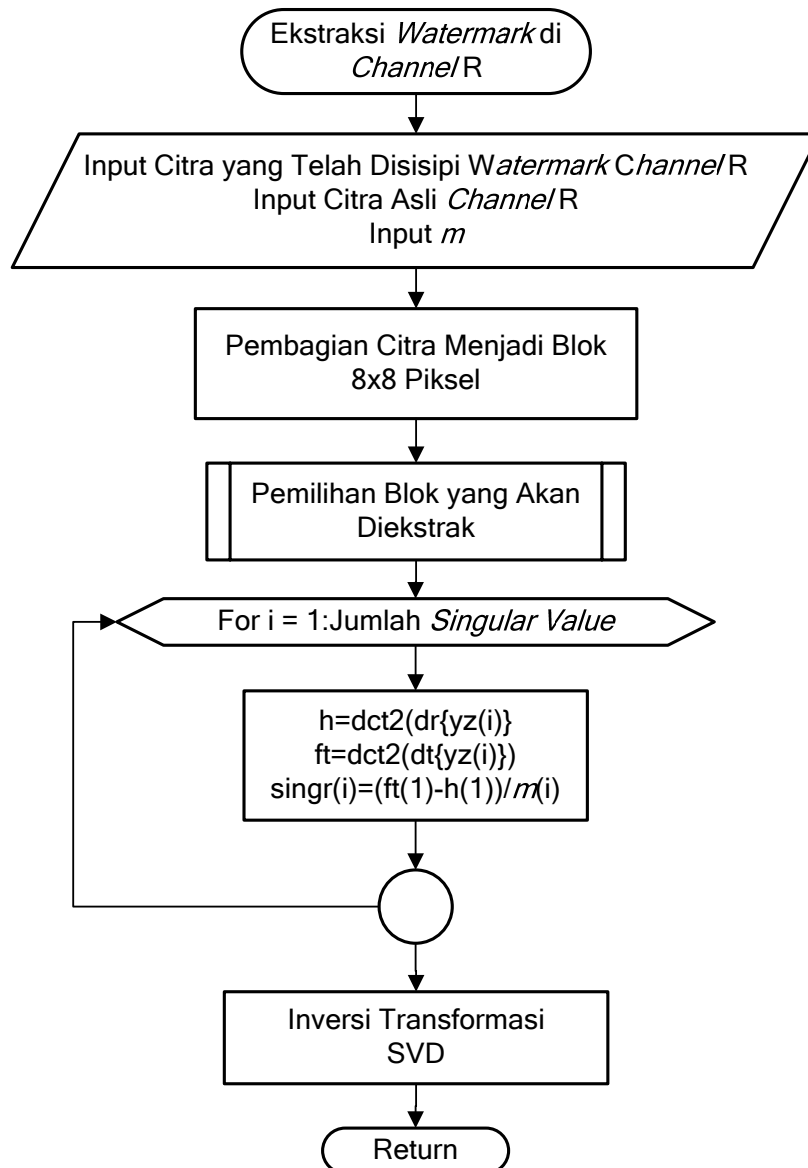


Gambar 6. Diagram Alir Ekstraksi Watermark

Algoritma ekstraksi *watermark* seperti yang ditunjukkan pada Gambar 6 adalah sebagai berikut:

1. *Watermark* asli ditransformasi SVD sehingga didapat matriks \mathbf{U} , \mathbf{S} dan \mathbf{V} . Matriks \mathbf{U} dan \mathbf{V} digunakan untuk inversi transformasi SVD pada proses ekstraksi *watermark*.
2. Proses ekstraksi *watermark* dilakukan di ketiga *channel* citra berwarna (RGB) yang telah disisipi *watermark*, yaitu *channel R*, *channel G*, *channel B*.
3. Setelah proses ekstraksi selesai, didapat *watermark* hasil ekstraksi dari ketiga *channel*.

A. Diagram Alir Ekstraksi Watermark di Channel R



Gambar 7. Diagram Alir Ekstraksi Watermark di Channel R

Algoritma ekstraksi *watermark* di *channel R* seperti yang ditunjukkan pada Gambar 7 adalah sebagai berikut:

1. Matriks *channel R* dari citra yang telah disisipi *watermark* dan citra asli, dibagi menjadi blok berukuran 8 x 8 piksel.
2. Kemudian blok-blok citra yang telah disisipi *watermark* dan citra asli, dipilih untuk diekstrak berdasarkan *chaotic sequence*.
3. Masukkan input *m*, dengan *m* didefinisikan sebagai faktor pengali untuk *singular value*.
4. Kemudian blok citra yang telah disisipi *watermark* dan citra asli yang terpilih dilakukan DCT. Elemen matriks baris pertama dari kolom pertama (komponen DC) dari citra yang telah disisipi *watermark* dan citra asli diselisihkan kemudian dibagi dengan *m*. Proses ini dilakukan menurut persamaan (6) untuk mendapatkan nilai *singular* yang disisipi.

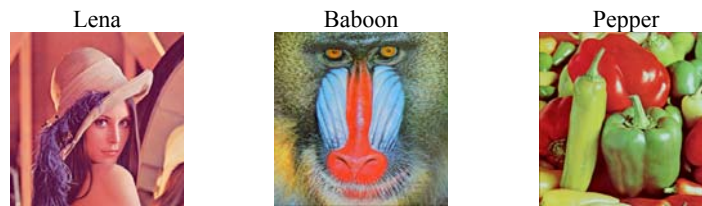
5. Kemudian dilakukan inversi transformasi SVD.

B. Diagram Alir Ekstraksi Watermark di Channel G dan B

Untuk ekstraksi *watermark* di *channel* G dan *channel* B dilakukan dengan cara yang sama dengan yang dilakukan di *channel* R.

IV. HASIL DAN ANALISIS

Proses *watermarking* diterapkan kepada tiga buah citra RGB (24 bit) berformat BMP dengan dua ukuran yaitu 256x256 dan 512x512 piksel seperti ditunjukkan pada Gambar 8.



Gambar 8. Citra-citra yang digunakan untuk pengujian

IV.1. Pemilihan Batas Nilai LPSNR

Dalam proses penyisipan *watermark* ke dalam citra digunakan suatu batas nilai LPSNR tertentu, agar *watermark* yang disisipkan ke dalam citra tidak terlalu mempengaruhi citra aslinya, akan tetapi harus dapat dideteksi kembali walaupun citra yang telah disisipi *watermark* mengalami berbagai distorsi. Cara pemilihan batas nilai LPSNR dilakukan berdasarkan percobaan, hasil yang diinginkan adalah *watermark* tidak merusak citra aslinya ($PSNR > 40$ dB), caranya dengan membandingkan antara citra asli dengan citra yang telah disisipi *watermark*. Selain itu *watermark* harus dapat dideteksi kembali walaupun telah mengalami berbagai distorsi. Caranya dengan membandingkan antara *watermark* yang diekstrak dari citra terdistorsi dengan *watermark* semula. Perbandingan dilakukan dengan menghitung nilai korelasinya. Nilai korelasi yang dinormalisasi yang diharapkan adalah mendekati satu, yang berarti *watermark* yang disisipkan pada citra yang terdistorsi masih dapat diekstrak dengan baik. Perhitungan nilai korelasi yang dinormalisasi ($NC = Normalized Corelation$) dilakukan dengan mengikuti persamaan^[12]

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N W_{(i,j)} W'_{(i,j)}}{\sum_{i=1}^M \sum_{j=1}^N [W_{(i,j)}]^2} \quad (7)$$

M dan N adalah lebar dan tinggi citra *watermark*, $W_{(i,j)}$ adalah elemen (i,j) dari matriks untuk *watermark* asli, dan $W'_{(i,j)}$ adalah elemen (i,j) dari matriks untuk *watermark* hasil ekstraksi.

Berdasarkan hasil percobaan, pada citra berukuran 256 x 256 dengan batas nilai $32 \leq LPSNR \leq 34$ keberadaan *watermark* relatif mengubah citra aslinya dan dengan batas $40 \leq LPSNR \leq 42$ keberadaan *watermark* relatif tidak mengubah citra, tetapi setelah didistorsi nilai korelasi (NC) relatif kecil. Oleh karena itu dipilih batas nilai $35 \leq LPSNR \leq 37$. Dengan batas nilai LPSNR ini, keberadaan *watermark* relatif tidak mengubah citra aslinya dan masih


memiliki nilai korelasi (*NC*) relatif besar walaupun telah mengalami berbagai distorsi.

Sedangkan pada citra berukuran 512 x 512 dengan batas nilai $29 \leq \text{LPSNR} \leq 31$ keberadaan *watermark* relatif mengubah citra aslinya dan dengan batas $39 \leq \text{LPSNR} \leq 42$ keberadaan *watermark* relatif tidak mengubah citra, tetapi setelah didistorsi nilai korelasi (*NC*) relatif kecil. Oleh karena itu dipilih batas nilai $32 \leq \text{LPSNR} \leq 34$. Dengan batas nilai LPSNR ini, keberadaan *watermark* relatif tidak mengubah citra aslinya dan masih memiliki nilai korelasi (*NC*) relatif besar walaupun telah mengalami berbagai distorsi.

IV.2. Bentuk Watermark

Watermark yang digunakan berupa citra *grayscale*. Pada citra berukuran 256 x 256 piksel dan 512 x 512 piksel disisipkan *watermark* berukuran 52 x 52 piksel seperti terlihat pada Tabel 1.



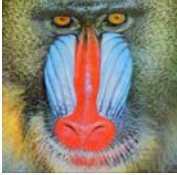
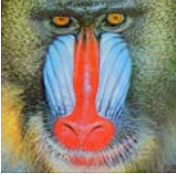








TABEL 1. WATERMARK YANG DIGUNAKAN

| Watermark | Ukuran |
|---|----------------|
|  | 52 x 52 piksel |

IV.3. Pengujian Kualitas Citra yang Telah Disisipi Watermark

Pengujian kualitas citra yang sudah disisipi *watermark* dilakukan dengan penilaian obyektif, yaitu dengan menghitung nilai PSNR (*Peak Signal to Noise Ratio*), dan penilaian subyektif dengan kriteria penilaian MOS (*Mean Opinion Score*). Nilai PSNR yang baik bagi suatu citra yang telah disisipi *watermark* adalah lebih besar dari 45 dB. Hal ini menunjukkan keberadaan *watermark* tidak merusak citra atau perubahan yang ditimbulkan *watermark* itu kecil sekali. Tabel 2 menunjukkan beberapa contoh citra yang telah disisipi *watermark*. Tabel 3 menunjukkan hasil nilai MOS dan PSNR dari masing-masing citra yang telah disisipi *watermark*. Nilai MOS diperoleh dari penilaian subyektif yang dilakukan oleh 20 orang responden.

TABEL 2. CONTOH CITRA ASLI DAN CITRA YANG TELAH DISISIPI WATERMARK

| Citra Asli | Citra Setelah Disisipi Watermark | Citra Asli | Citra Setelah Disisipi Watermark | Citra Asli | Citra Setelah Disisipi Watermark |
|---|---|---|---|--|---|
| Lena 256  | Lena 256  | Baboon 256  | Baboon 256  | Pepper 256  | Pepper 256  |
| Lena 512  | Lena 512  | Baboon 512  | Baboon 512  | Pepper 512  | Pepper 512  |

TABEL 3. NILAI MOS DAN PSNR CITRA YANG TELAH DISISIPI *WATERMARK*

| Citra yang Telah Disisipkan <i>Watermark</i> | Nilai Skala | | | | | MOS | PSNR (dB) |
|--|-------------|---|---|---|---|------|-----------|
| | 5 | 4 | 3 | 2 | 1 | | |
| Lena 256 | 11 | 7 | 2 | 0 | 0 | 4.45 | 50.87 |
| Baboon 256 | 14 | 5 | 1 | 0 | 0 | 4.65 | 50.87 |
| Pepper 256 | 15 | 5 | 0 | 0 | 0 | 4.75 | 50.87 |
| Lena 512 | 14 | 4 | 2 | 0 | 0 | 4.60 | 55.00 |
| Baboon 512 | 16 | 4 | 0 | 0 | 0 | 4.80 | 55.00 |
| Pepper 512 | 17 | 3 | 0 | 0 | 0 | 4.85 | 55.00 |

Berdasarkan nilai MOS yang ditunjukkan dalam Tabel 3 diperoleh:

- Citra Lena yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.45, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya. Citra Lena yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.65, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya.
- Citra Baboon yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.60, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya. Citra Baboon yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.8, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya.
- Citra Pepper yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.75, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya. Citra Pepper yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai MOS 4.85, berada pada skala penilaian *good* (sama), yang berarti keberadaan *watermark* relatif tidak mengubah citra aslinya.

Berdasarkan nilai PSNR ditunjukkan dalam Tabel 3 diperoleh:

- Citra Lena yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR 50.87 dB. Pada citra Lena yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR yang lebih baik yaitu 55.00 dB.
- Citra Baboon yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR 50.87 dB. Pada citra Baboon yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR yang lebih baik yaitu 55.00 dB.
- Citra Pepper yang berukuran 256 x 256 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR 50.87 dB. Pada citra Pepper yang berukuran 512 x 512 piksel setelah disisipkan *watermark* menghasilkan nilai PSNR yang lebih baik yaitu 55.00 dB.

Berdasarkan nilai MOS dan PSNR didapat bahwa ukuran citra menentukan kualitas citra yang disisipi *watermark*. Lebih besar ukuran citra yang disisipi *watermark*, maka kualitas citra tersebut akan lebih baik, demikian juga sebaliknya.

IV.4. Pengujian Ketahanan *Watermark*

Untuk menguji ketahanan *watermark* maka citra yang telah disisipi *watermark* diberi distorsi berupa pengolahan citra yang umum dilakukan yaitu kompresi JPEG, *cropping*, *scaling* (diperbesar dan diperkecil), *filtering*, *rotate*, dan pemberian *noise*.

Untuk mengetahui ketahanan, *watermark* diekstrak dari citra yang telah diberi distorsi. Kemudian dibandingkan dengan *watermark* asli dengan menghitung nilai korelasinya menggunakan persamaan (7). *Watermark* dapat dikategorikan tahan terhadap distorsi yang telah dilakukan pada citra jika *watermark* tersebut masih terlihat jelas ketika dilihat secara visual. *Watermark* yang terlihat jelas, umumnya memiliki nilai korelasi (*NC*) relatif besar (maksimal sama dengan satu).
















A. Kompresi JPEG

Citra yang telah disisipi *watermark* dikompres ke dalam format JPEG dengan faktor kualitas (*Q*) = 3, 5, 10. Semakin besar nilai faktor kualitas, maka kualitas gambar hasil kompresi semakin baik. Nilai *NC watermark* yang diekstrak dari citra yang telah dilakukan proses kompresi JPEG dapat dilihat pada Tabel 4. Contoh *watermark* hasil ekstraksi dari citra yang dikompres ditunjukkan pada Tabel 5.

TABEL 4. NILAI *NC WATERMARK* YANG DIEKSTRAK DARI CITRA YANG TELAH DIKOMPRES (*Q* = 3, 5, & 10)

| Citra yang Telah Disisipi Watermark | Nilai korelasi (<i>NC</i>) | | | | | | | | |
|-------------------------------------|------------------------------|--------|--------|--------------|--------|--------|---------------|--------|--------|
| | <i>Q</i> = 3 | | | <i>Q</i> = 5 | | | <i>Q</i> = 10 | | |
| | R | G | B | R | G | B | R | G | B |
| Lena 256 | 0.9606 | 0.9824 | 0.9524 | 0.9835 | 0.9902 | 0.9583 | 0.9938 | 0.9972 | 0.9994 |
| Baboon 256 | 0.9520 | 0.9505 | 0.9653 | 0.9898 | 0.9939 | 0.8929 | 0.9996 | 0.9973 | 0.9949 |
| Pepper 256 | 0.9018 | 0.9137 | 0.9882 | 0.9643 | 0.9761 | 0.9927 | 0.9986 | 0.9979 | 0.9987 |
| Lena 512 | 0.9574 | 0.9969 | 0.9910 | 0.9952 | 0.9955 | 0.9927 | 0.9992 | 0.9996 | 0.9982 |
| Baboon 512 | 0.6260 | 0.9859 | 0.8454 | 0.9991 | 0.9817 | 0.9903 | 0.9976 | 0.9998 | 0.9987 |
| Pepper 512 | 0.9774 | 0.9979 | 0.7589 | 0.9935 | 0.9952 | 0.9939 | 0.9995 | 0.9998 | 0.9997 |

TABEL 5. CONTOH *WATERMARK* HASIL EKSTRAKSI DARI CITRA YANG DIKOMPRES (*Q* = 10)

| Citra yang Telah Disisipkan Watermark | Citra Setelah Dikompres (<i>Q</i> = 10) | Watermark Hasil Ekstraksi | | |
|---|---|--|---|---|
| | | R | G | B |
| Lena 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9992 | ekstraksi watermark green  <i>NC</i> = 0.9996 | ekstraksi watermark blue  <i>NC</i> = 0.9982 |
| Baboon 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9976 | ekstraksi watermark green  <i>NC</i> = 0.9998 | ekstraksi watermark blue  <i>NC</i> = 0.9987 |
| Pepper 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9995 | ekstraksi watermark green  <i>NC</i> = 0.9998 | ekstraksi watermark blue  <i>NC</i> = 0.9997 |

Dari Tabel 4 dapat dilihat bahwa nilai korelasi (NC) yang diperoleh umumnya lebih besar dari 0.9. Walaupun nilai NC ada yang lebih kecil dari 0.9 pada salah satu *channel* warna, namun pada *channel* warna yang lain nilainya lebih besar dari 0.9. Untuk nilai NC 0.62, *watermark* hasil ekstraksi masih terlihat cukup jelas secara visual. Oleh karena itu *watermark* masih tahan terhadap proses kompresi JPEG.



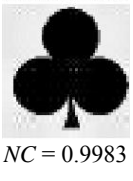
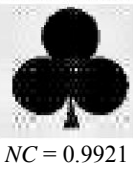
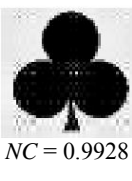



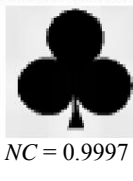
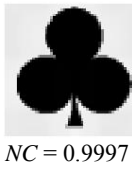


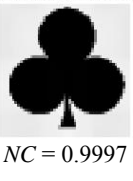

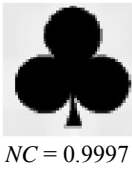
B. Cropping

Cropping adalah proses pemotongan citra. *Cropping* dilakukan pada citra yang telah disisipi *watermark*. Tabel 6 menunjukkan nilai NC *watermark* yang diekstrak dari citra yang telah dilakukan proses *cropping*. Tabel 7 memperlihatkan contoh *watermark* hasil ekstraksi dari citra yang telah dilakukan *cropping*.

TABEL 6. NILAI NC WATERMARK YANG DIEKSTRAK DARI CITRA YANG TELAH DILAKUKAN *CROPPING*

| Citra yang Telah Disisipi Watermark | Nilai Korelasi (NC) | | | | | | | | |
|-------------------------------------|-------------------------|--------|--------|------------------|--------|--------|------------------|--------|--------|
| | Sudut Kiri Atas | | | Sudut Kanan Atas | | | Sudut Kiri Bawah | | |
| | R | G | B | R | G | B | R | G | B |
| Lena 256 | 0.9979 | 0.9973 | 0.9975 | 0.8228 | 0.7220 | 0.7655 | 0.9978 | 0.9976 | 0.9977 |
| Baboon 256 | 0.9979 | 0.9979 | 0.9976 | 0.8816 | 0.8941 | 0.8251 | 0.9981 | 0.9981 | 0.9981 |
| Pepper 256 | 0.9981 | 0.9981 | 0.9981 | 0.8528 | 0.9320 | 0.8013 | 0.9981 | 0.9981 | 0.9981 |
| Lena 512 | 0.9983 | 0.9921 | 0.9928 | 0.9995 | 0.9207 | 0.7370 | 0.9983 | 0.9966 | 0.9974 |
| Baboon 512 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9996 | 0.9995 | 0.9993 |
| Pepper 512 | 0.9997 | 0.9997 | 0.9997 | 0.7940 | 0.8954 | 0.6967 | 0.9996 | 0.9996 | 0.9991 |

TABEL 7. CONTOH WATERMARK HASIL EKSTRAKSI DARI CITRA YANG TELAH DILAKUKAN *CROPPING*

| Citra yang Telah Disisipkan Watermark | Citra Setelah Dilakukan <i>Cropping</i> | Watermark Hasil Ekstraksi | | |
|---|---|---|--|--|
| | | R | G | B |
| Lena 512  |  | ekstraksi watermark red  $NC = 0.9983$ | ekstraksi watermark greer  $NC = 0.9921$ | ekstraksi watermark blue  $NC = 0.9928$ |
| Baboon 512  |  | ekstraksi watermark red  $NC = 0.9997$ | ekstraksi watermark gree  $NC = 0.9997$ | ekstraksi watermark blue  $NC = 0.9997$ |
| Pepper 512  |  | ekstraksi watermark red  $NC = 0.9997$ | ekstraksi watermark green  $NC = 0.9997$ | ekstraksi watermark blue  $NC = 0.9997$ |

Dari Tabel 6 dapat dilihat bahwa nilai korelasi (NC) yang diperoleh umumnya lebih

besar dari 0.9. Walaupun nilai *NC* ada yang lebih kecil dari 0.9 pada salah satu *channel* warna, namun pada *channel* warna yang lain nilainya lebih besar dari 0.8. Untuk nilai *NC* 0.72, *watermark* hasil ekstraksi masih terlihat cukup jelas secara visual. Oleh karena itu *watermark* masih tahan terhadap proses *copping* yang dilakukan.



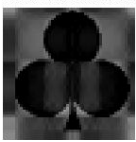
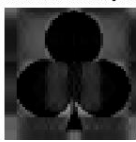
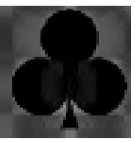


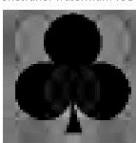
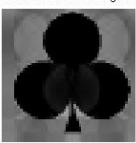
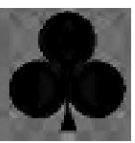
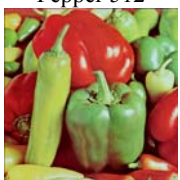


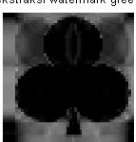
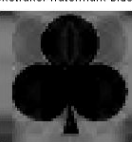
C. Scaling

Citra yang telah disisipkan *watermark* akan diperbesar atau diperkecil dengan faktor skala tertentu. Hasilnya kemudian disimpan lalu dikembalikan ke ukuran semula, kemudian *watermark* diekstrak dari citra tersebut. Nilai korelasi (*NC*) *watermark* yang diekstrak dari citra yang telah dilakukan proses *scaling* ditunjukkan pada Tabel 9. Tabel 10 memperlihatkan contoh *watermark* hasil ekstraksi dari citra yang telah dilakukan *scaling*.

TABEL 9. NILAI *NC WATERMARK* YANG DIEKSTRAK DARI CITRA YANG TELAH DILAKUKAN *SCALING*

| Citra yang Telah Disisipkan <i>Watermark</i> | Nilai Korelasi (<i>NC</i>) | | | | | |
|--|------------------------------|--------|--------|----------------------|---------|---------|
| | Faktor Skala = 1.5x | | | Faktor Skala = 0.25x | | |
| | R | G | B | R | G | B |
| Lena 256 | 0.7774 | 0.7326 | 0.9388 | -0.6545 | -0.6511 | -0.5471 |
| Baboon 256 | 0.9782 | 0.9872 | 0.9900 | 0.2145 | -0.2290 | -0.3835 |
| Pepper 256 | 0.8754 | 0.8064 | 0.9469 | -0.1139 | 0.2762 | 0.3250 |
| Lena 512 | 0.9967 | 0.9953 | 0.9951 | 0.2249 | 0.0588 | -0.0090 |
| Baboon 512 | 0.9917 | 0.9964 | 0.9901 | -0.2111 | -0.2906 | 0.2055 |
| Pepper 512 | 0.9983 | 0.9977 | 0.9499 | -0.0233 | -0.5799 | -0.0111 |

TABEL 10. CONTOH *WATERMARK* HASIL EKSTRAKSI DARI CITRA YANG TELAH DILAKUKAN *SCALING* 1.5X

| Citra yang Telah Disisipkan <i>Watermark</i> | Citra Setelah Dilakukan <i>Scaling</i> | <i>Watermark</i> Hasil Ekstraksi | | |
|---|---|--|---|---|
| | | R | G | B |
| Lena 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9967 | ekstraksi watermark greer  <i>NC</i> = 0.9953 | ekstraksi watermark blue  <i>NC</i> = 0.9951 |
| Baboon 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9917 | ekstraksi watermark green  <i>NC</i> = 0.9964 | ekstraksi watermark blue  <i>NC</i> = 0.9901 |
| Pepper 512  |  | ekstraksi watermark red  <i>NC</i> = 0.9983 | ekstraksi watermark greer  <i>NC</i> = 0.9977 | ekstraksi watermark blue  <i>NC</i> = 0.9499 |

Pada proses *scaling* (diperbesar), umumnya nilai korelasi (*NC*) lebih besar dari 0.9.

Walaupun ada yang lebih kecil pada salah satu *channel* warna, tetapi *channel* warna lain mempunyai nilai *NC* yang lebih besar dari 0.9, sehingga *watermark* hasil ekstraksi masih dapat dilihat dengan jelas secara visual. Sedangkan untuk proses *scaling* (diperkecil) *watermark* yang diekstrak memiliki nilai korelasi yang kecil dan jika dilihat secara visual *watermark* tidak terlihat dengan jelas. Jadi *watermark* tahan terhadap proses *scaling* (diperbesar), tetapi tidak tahan terhadap proses *scaling* (diperkecil) yang dilakukan.


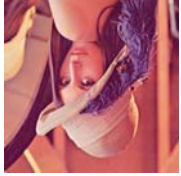
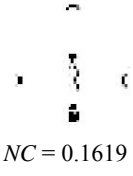
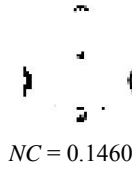
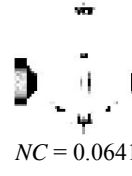
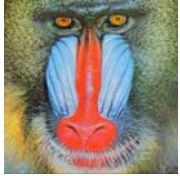


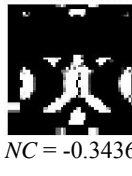
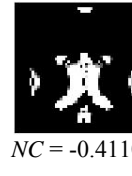




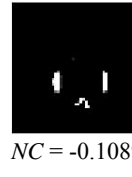
D. Rotasi

Citra yang telah disisipkan *watermark* akan dirotasi dengan sudut putar 90° , -90° , 180° . Nilai korelasi (*NC*) *watermark* yang diekstrak dari citra yang telah dilakukan proses rotasi ditunjukkan pada Tabel 11. Tabel 12 memperlihatkan contoh *watermark* hasil ekstraksi dari citra yang telah dilakukan rotasi.

TABEL 11. NILAI *NC WATERMARK* YANG DIEKSTRAK DARI CITRA YANG TELAH DILAKUKAN ROTASI

| Citra yang Telah Disisipi Watermark | Nilai Korelasi (<i>NC</i>) | | | | | | | | |
|-------------------------------------|------------------------------|---------|---------|--------------------|---------|---------|--------------------|---------|---------|
| | Rotasi 90° | | | Rotasi -90° | | | Rotasi 180° | | |
| | R | G | B | R | G | B | R | G | B |
| Lena 256 | 0.0201 | 0.3122 | 0.3000 | 0.3509 | 0.4157 | 0.5888 | 0.0024 | -0.0894 | -0.1108 |
| Baboon 256 | 0.7337 | -0.0825 | -0.4128 | 0.8359 | -0.2508 | -0.5199 | -0.2657 | -0.2855 | -0.2896 |
| Pepper 256 | 0.0823 | 0.0587 | 0.1369 | -0.6666 | -0.0243 | -0.0293 | -0.1779 | 0.5695 | 0.4379 |
| Lena 512 | -0.0946 | -0.1593 | -0.1928 | -0.0571 | -0.1094 | -0.0996 | 0.1619 | 0.1460 | 0.0641 |
| Baboon 512 | -0.4573 | -0.1492 | -0.2543 | -0.3180 | -0.7565 | -0.4798 | 0.0437 | -0.3436 | -0.4110 |
| Pepper 512 | 0.4127 | -0.3943 | -0.2455 | 0.2627 | -0.0009 | -0.1190 | 0.2223 | -0.2304 | -0.1089 |

TABEL 12. CONTOH *WATERMARK* HASIL EKSTRAKSI DARI CITRA YANG TELAH DILAKUKAN ROTASI 180°

| Citra yang Telah Disisipkan Watermark | Citra Setelah Dilakukan Rotasi | Watermark Hasil Ekstraksi | | |
|---|---|---|---|---|
| | | R | G | B |
| Lena 512  |  | ekstraksi watermark red  $NC = 0.1619$ | ekstraksi watermark green  $NC = 0.1460$ | ekstraksi watermark blue  $NC = 0.0641$ |
| Baboon 512  |  | ekstraksi watermark red  $NC = 0.0437$ | ekstraksi watermark green  $NC = -0.3436$ | ekstraksi watermark blue  $NC = -0.4110$ |
| Pepper 512  |  | ekstraksi watermark red  $NC = 0.2223$ | ekstraksi watermark green  $NC = -0.2304$ | ekstraksi watermark blue  $NC = -0.1089$ |

Nilai *NC watermark* yang diekstrak dari citra yang dirotasi umumnya lebih kecil dari 0.5 dan *watermark* hasil ekstraksi tidak terlihat jelas secara visual. Jadi *watermark* tidak tahan terhadap proses rotasi yang dilakukan.



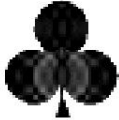


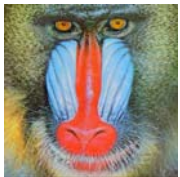


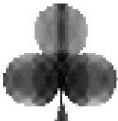

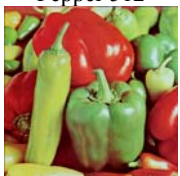




E. Filtering

Citra yang sudah disisipi *watermark* difilter dengan *median filter*, kemudian dari citra yang telah difilter tersebut *watermark* diekstrak kembali. Tabel 13 menunjukkan nilai *NC watermark* yang diekstrak dari citra yang telah difilter. Tabel 14 memperlihatkan contoh *watermark* hasil ekstraksi dari citra yang telah difilter menggunakan filter median dengan *mask* 5x5.

TABEL 13. NILAI *NC WATERMARK* YANG DIEKSTRAK DARI CITRA YANG TELAH DIFILTER DENGAN MEDIAN FILTERING

| Citra yang Telah Disisipi Watermark | Nilai Korelasi (NC) | | | | | | | | |
|-------------------------------------|---------------------|--------|--------|----------|--------|--------|----------|--------|--------|
| | Mask 3x3 | | | Mask 5x5 | | | Mask 7x7 | | |
| | R | G | B | R | G | B | R | G | B |
| Lena 256 | 0.9746 | 0.9759 | 0.9781 | 0.9499 | 0.9629 | 0.9721 | 0.7226 | 0.8516 | 0.9445 |
| Baboon 256 | 0.9750 | 0.9700 | 0.9473 | 0.9774 | 0.9768 | 0.9764 | 0.9784 | 0.9801 | 0.9815 |
| Pepper 256 | 0.9611 | 0.9648 | 0.9742 | 0.9546 | 0.9542 | 0.9633 | 0.9340 | 0.9652 | 0.9622 |
| Lena 512 | 0.9620 | 0.9786 | 0.9794 | 0.9682 | 0.9793 | 0.9805 | 0.9702 | 0.9740 | 0.9784 |
| Baboon 512 | 0.8952 | 0.9030 | 0.9376 | 0.9443 | 0.9095 | 0.9495 | 0.9607 | 0.6655 | 0.8791 |
| Pepper 512 | 0.9756 | 0.9783 | 0.9729 | 0.9791 | 0.9805 | 0.9741 | 0.9825 | 0.9827 | 0.9734 |

TABEL 14. CONTOH *WATERMARK* HASIL EKSTRAKSI DARI CITRA YANG TELAH DIFILTER DENGAN *MASK* 5X5

| Citra yang Telah Disisipkan Watermark | Citra Setelah Difilter | Watermark Hasil Ekstraksi | | |
|---|---|---|---|--|
| | | R | G | B |
| Lena 512  |  | ekstraksi watermark red  NC = 0.9682 | ekstraksi watermark gree  NC = 0.9793 | ekstraksi watermark blue  NC = 0.9805 |
| Baboon 512  |  | ekstraksi watermark red  NC = 0.9443 | ekstraksi watermark gree  NC = 0.9095 | ekstraksi watermark blue  NC = 0.9495 |
| Pepper 512  |  | ekstraksi watermark red  NC = 0.9791 | ekstraksi watermark green  NC = 0.9805 | ekstraksi watermark blue  NC = 0.9741 |

Tabel 13 menunjukkan bahwa nilai korelasi (*NC watermark*) yang diekstrak dari citra

yang telah difilter dengan *median filter* umumnya mempunyai nilai lebih besar dari 0.9. Paling sedikit satu *channel* warna mempunyai nilai *NC* lebih besar dari 0.9 dan *watermark* hasil ekstraksi terlihat jelas secara visual. Jadi *watermark* tahan terhadap proses pemfilteran yang dilakukan.

V. KESIMPULAN

Dari hasil ujicoba yang telah dilakukan, *watermaking* dengan teknik yang digunakan menghasilkan citra yang telah disisipi *watermark* mempunyai kualitas yang baik (PSNR > 50 dB dan MOS > 4). Ukuran citra yang lebih besar akan menghasilkan kualitas citra yang telah disisipi *watermark* lebih baik.

Penyisipan *watermark* dengan teknik yang digunakan menghasilkan ketahanan *watermark* yang baik terhadap beberapa proses pengolahan citra seperti kompresi JPEG, *cropping*, *scaling* (diperbesar), dan pemfilteran. *Watermark* yang disisipkan dengan teknik yang digunakan tidak tahan terhadap proses *scaling* (diperkecil) dan rotasi.

DAFTAR PUSTAKA

- [1] F. Hartung and M. Kutter, "Multimedia watermarking Techniques", *Proceeding of the IEEE*, vol. 87, No. 7, July 1999
- [2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding", *IBM System Journal*, Vol.35, NOS 3&4, 1996
- [3] I.J. Cox, J. Killian, T. Leighton, and T. Shamoan, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Transactions of Image Processing*, Vol. 6 No. 12, pp.1673-1687, Dec. 1997.
- [4] E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling", *IEEE Workshop on Nonlinear Signal and Image Processing* (Neos Marmaras, Greece, June 20-22, 1995
- [5] D.V.S. Chandra, "Digital Image Watermarking Using Singular Value Decomposition", *Proceedings of 45th IEEE Midwest Symposium on Circuits and Systems*, Tulsa, OK, 2002, pp. 264-267.
- [6] R. Sun, H. Sun, and T. Yao, "A SVD and quantization based semi-fragile watermarking technique for image authentication" *Proc. IEEE International Conf. Signal Processing*, 2, 2002, pp. 1592-1595
- [7] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership", *IEEE Trans. Multimedia*, (4), 1, pp, 121-128, 2002
- [8] P.C. Hansen, "*Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*", Phildelphia: SIAM, 1998.
- [9] R.C. Gonzales and R.E. Woods, "*Digital Image Processing*" 3rd ed., New Jersey: Pearson Prentice Hall, 2010.
- [10] H.O. Peitgen, H. Jurgens, and D. Sanpe, "*Chaos and Fractals*", New York: Springer-Verlag, 1992.
- [11] F.J. Huang and Z.H. Guan, "A Hybrid SVD-DCT Watermarking Method Based on LPSNR" *Pattern Recognition Letters* 25, 1769-1775, 2004
- [12] C.T. Hsu and J.L. Wu, "Hidden signatures in images", in *Proceedings of International Conference on Image Processing*, 1996, pp. 223-226.