

Studi Perbandingan Metode DCT dan SVD pada *Image Watermarking*

Shofi Nur Fathiya - 13508084
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18084@students.if.itb.ac.id

Abstract—Salah satu metode yang digunakan dalam menjaga keaslian media digital, misalnya citra, adalah dengan memberi watermark pada citra tersebut. Watermark menyisipkan logo atau penanda lain yang menjadi ciri khas pembuat citra tersebut. Logo ini tidak hilang walaupun citra yang diberi watermark diubah isinya. Watermark ini juga merupakan salah satu bentuk steganografi, yakni menyisipkan suatu data ke data lain. Ada beberapa pendekatan yang digunakan dalam watermarking, namun yang akan dibahas pada makalah ini adalah pendekatan Discrete Cosine Transform (DCT) dan Singular Value Decomposition (SVD).

Index Terms—citra, watermark, DCT, SVD.

I. PENDAHULUAN

Perkembangan informasi saat ini semakin bertambah pesat seiring dengan munculnya internet. Berbagai media, terutama media digital, dapat disebarluaskan kemana pun dengan cepat dan mudah. Salah satu media yang sering disebarluaskan adalah citra digital. Citra dapat dikirimkan, digandakan, diubah isinya, hingga disebarluaskan kembali. Hal ini menyebabkan citra menjadi hilang keasliannya. Pembuat citra tidak dapat memberikan bukti bahwa citra tersebut adalah miliknya.

Karena dirasakan pentingnya menjaga keaslian citra ini, akhirnya ditemukan sebuah metode yang tepat untuk menjaga keaslian citra tersebut. Metode ini adalah *watermarking*, yaitu menyisipkan logo atau tanda khas dari pembuat citra ke dalam citra. Tanda ini tidak akan hilang atau rusak walaupun citra digandakan ataupun diubah, sehingga originalitas citra tetap terjaga.

Dalam ilmu kriptografi, *watermark* termasuk dalam salah satu pengembangan steganografi. Namun, berbeda dengan kriptografi yang menyembunyikan pesan dengan cara mengubahnya menjadi bentuk lain, steganografi menyisipkan pesan ke dalam suatu media penampung tanpa merusak media tersebut sehingga pesan tidak terlihat dari luar. Jika media penampung yang digunakan adalah citra, maka keberadaan pesan akan lolos dari pandangan karena citra yang telah disisipi pesan tidak berbeda dengan citra aslinya bila dilihat dengan mata

manusia. Dari steganografi ini kemudian lahirlah pemikiran-pemikiran baru, dimana salah satunya adalah *watermark*. Pada *watermark*, pesan yang disisipkan merupakan logo atau tanda khas dari pembuat media sementara media yang disisipkan pesan tersebut adalah hasil karya pembuat media tersebut.

Watermarking pada media digital memiliki beberapa sifat umum yang harus dipenuhi, yaitu:

1. Invisibility

Watermark tidak terdeteksi oleh panca indra, misalkan tidak terlihat untuk media citra atau tidak terdengar untuk media audio. Namun sifat ini tidak berlaku untuk *visible watermarking*.

2. Robustness

Watermark tidak mudah diubah atau dihapus dari media akibat proses *editing*, kompresi, *filter*, dan sebagainya.

3. Trackable

Penyebaran citra ber-*watermark* dapat dilacak.

Watermarking pada citra terbagi menjadi dua domain berbeda, yaitu domain spasial dan domain transformasi. Pada domain spasial, penyisipan *watermark* dilakukan langsung pada nilai byte dari pixel citra, misalnya dengan mengganti nilai bit terakhir dari pixel citra atau yang dikenal dengan metode *Least Significant Bit (LSB)*. Namun terdapat beberapa kekurangan pada *watermark* yang dilakukan dalam domain spasial ini, salah satunya *watermark* mudah dihapuskan dari citra dengan mengganti semua bit LSB pada citra. Untuk mengantisipasi kekurangan tersebut, maka *watermarking* dapat dilakukan dengan cara lain, domain transformasi. Pada domain ini, penyisipan *watermark* dilakukan pada koefisien transformasi pada citra. Penggunaan domain transformasi ini memberikan nilai robust lebih daripada domain spasial. Beberapa metode yang dapat dilakukan pada domain transformasi adalah *Discrete Wavelet Transform (DWT)*, *Discrete Fourier Transform (DFT)*, *Discrete Cosine Transform (DCT)*, dan *Singular Value Decomposition (SVD)*. Namun yang akan dibahas dalam makalah ini hanyalah 2 metode, yaitu DCT dan SVD.

II. DASAR TEORI

A. Discrete Cosine Transform (DCT)

Discrete Cosine Transform merupakan salah satu metode yang digunakan dalam melakukan *watermarking* yang dilakukan dengan cara memetakan himpunan nilai sebanyak N buah pada domain spasial menjadi himpunan lain dengan jumlah yang sama pada domain frekuensi. Nilai dari himpunan tersebut adalah bilangan real.

Konsep yang digunakan pada metode DCT adalah mengganti koefisien DCT pada citra asli menjadi koefisien baru dengan menggunakan fungsi gelombang cosinus diskrit. Pemilihan koefisien yang diganti dapat berbeda-beda tergantung pilihan frekuensi. Jika frekuensi yang dipilih adalah frekuensi tinggi dimana koefisien yang dihasilkan bernilai rendah, maka citra hasil *watermark* tidak akan mengalami perubahan berarti (nilai invisibility tinggi) sehingga tidak terlihat perubahannya oleh mata manusia. Namun penggunaan frekuensi ini menyebabkan citra lemah terhadap perubahan, misalkan *cropping*, *editing*, dan lain-lain (nilai robustness rendah). Dan jika yang dipilih merupakan frekuensi rendah, dimana koefisien citra yang dihasilkan adalah koefisien bernilai tinggi, maka citra hasil *watermarking* kuat terhadap perubahan (nilai robustness tinggi) namun perubahan pada citra ini mudah terlihat (nilai invisibility rendah).

Cara mendapatkan nilai koefisien DCT adalah dengan menghitung fungsi basis cosinus. Untuk matriks 1 dimensi yang berukuran N, rumus yang digunakan adalah :

$$C(x) = \alpha_x \sum_{n=0}^{N-1} I(n) \frac{\cos\pi(2n+1)x}{2N}$$

Persamaan 1. Mencari nilai koefisien DCT pada matriks 1 dimensi

dimana

C : koefisien pada indeks ke-x
 α_x : faktor skalar
 N : ukuran matriks
 I : image
 x : indeks yang sedang dicari

dan α_x didapatkan dari

$$\alpha_x = 1 / \sqrt{N} \quad , x = 0$$

$$\alpha_x = \sqrt{2/N} \quad , 1 \leq x \leq N-1$$

Persamaan 2. Mencari nilai faktor skalar untuk mencari koefisien DCT pada matriks 1 dimensi

Karena citra merupakan 2 dimensi, maka rumus yang digunakan adalah sebagai berikut.

$$C(x,y) = \alpha_x \alpha_y \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \frac{\cos\pi(2m+1)x}{2M} \frac{\cos\pi(2n+1)y}{2N}$$

Persamaan 3. Mencari nilai koefisien DCT pada matriks 2 dimensi

dimana

C : koefisien pada indeks ke-x
 α_x, α_y : faktor skalar
 M : ukuran tinggi matriks
 N : ukuran lebar matriks
 I : image (citra)
 x, y : indeks yang sedang dicari nilainya

dan α_x didapatkan dari

$$\alpha_x = 1 / \sqrt{M} \quad , x = 0$$

$$\alpha_x = \sqrt{2/M} \quad , 1 \leq x \leq M-1$$

$$\alpha_y = 1 / \sqrt{N} \quad , y = 0$$

$$\alpha_y = \sqrt{2/N} \quad , 1 \leq y \leq N-1$$

Persamaan 4. Mencari nilai faktor skalar untuk mencari koefisien DCT pada matriks 2 dimensi

1. Proses Penyisipan

Langkah-langkah yang dilakukan dalam menyisipkan *watermark* ke dalam citra adalah sebagai berikut :

- Transformasikan nilai-nilai pada citra untuk mendapatkan nilai koefisien DCT citra. Rumus yang digunakan adalah rumus di atas.
- Tentukan frekuensi yang akan disisipi *watermark*. Pemilihan frekuensi akan mempengaruhi nilai robustness dan invisibility.
- Sisipkan *watermark* ke dalam koefisien DCT. Ubah nilai koefisien DCT dengan rumus

$$v'_i = v_i (1 + \alpha w_i)$$

Persamaan 5. Menyisipkan nilai *watermark* ke koefisien DCT

dimana

v'_i : nilai koefisien DCT setelah diubah dengan *watermark*
 v_i : nilai koefisien DCT sebelum diubah dengan *watermark*
 α : faktor skalar, nilai yang sering digunakan adalah 0.1
 w_i : *watermark* indeks ke-i

- Lakukan inversi DCT (IDCT) pada koefisien DCT agar gambar dapat terlihat kembali dengan rumus berikut :

$$I(m,n) = \alpha x \alpha y \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C(x,y) \frac{\cos\pi(2m+1)x}{2M} \frac{\cos\pi(2n+1)y}{2N}$$

Persamaan 6. Melakukan inversi DCT

2. Proses Ekstraksi

Untuk melakukan ekstraksi, diperlukan citra asli, citra ber-*watermark*, dan *watermark* asli. Langkah-langkah yang dilakukan dalam melakukan ekstraksi citra ber-*watermark* adalah sebagai berikut.

- Lakukan transformasi pada kedua citra, yaitu citra asli dan citra ber-*watermark*, seperti pada proses penyisipan.
- Bandingkan hasil koefisien DCT kedua citra. Jika hasil koefisien berbeda maka kemungkinan citra uji telah memiliki *watermark*.
- Cari nilai w' dengan menggubakan rumus yang sama saat mengubah koefisien DCT saat penyisipan.
- Buat sebuah himpunan bilangan acak yang sama dengan bilangan acak pada proses penyisipan. Cari nilai w dengan membagi nilai w' dengan bilangan acak.
- Ubah setiap nilai -1 menjadi 0.
- Susun nilai-nilai tersebut dengan mengubah nilai 0 menjadi hitam dan 1 menjadi putih.
- Bandingkan hasil *watermark* ini dengan *watermark* asli dan hitung BER nya. BER dihitung dengan rumus :

$$BER = \frac{\sum w_i w'_i}{\text{sqrt}(\sum w_i \sum w'^2_i)}$$

Persamaan 7. Mencari nilai BER untuk menghitung perbedaan nilai *watermark* masukan pengguna dengan *watermark* hasil ekstraksi

Jika BER masih dalam suatu batas toleransi tertentu, maka *watermark* dikatakan cocok. Namun jika nilai BER berada di luar batas toleransi, maka terdapat beberapa kemungkinan, yaitu *watermark* inputan berbeda dengan *watermark* asli ataupun citra telah diubah nilainya.

B. Singular Value Decomposition (SVD)

Singular Value Decomposition merupakan salah satu teknik analisis numerik dalam mendiagonalkan suatu matriks. Teknik ini kini digunakan dalam menyisipkan *watermark* ke dalam citra. SVD ini merupakan teknik yang lebih baru jika dibandingkan dengan teknik lainnya, misalkan DCT atau DFT.

Misalkan terdapat sebuah citra berukuran $M \times N$ yang akan diberi *watermark*. Citra tersebut dapat direpresentasikan dalam sebuah matriks bukan nol C dan dibuat menjadi persamaan :

$$C = USV^T$$

Persamaan 8. Mencari nilai matriks singular dari suatu citra

dimana

- C : matriks dari citra berukuran $M \times N$
- U : matriks ortogonal berukuran $M \times M$
- S : matriks singular dengan ukuran $M \times N$
- V : matriks orthogonal berukuran $N \times N$

1. Penyisipan *watermark*

Langkah-langkah yang dilakukan dalam melakukan penyisipan *watermark* dengan metode SVD adalah sebagai berikut:

- Bangkitkan himpunan bilangan random sejumlah himpunan pada citra *watermark*. Himpunan bilangan ini merupakan indeks pada citra yang akan diubah nilainya.
- Untuk setiap indeks yang akan diubah nilainya, ubah terlebih dahulu nilai LSB-nya menjadi 0.
- Cari nilai S dari persamaan di atas.
- Tambahkan nilai S ini dengan nilai *watermark* sehingga ditemukan S_t dengan persamaan:

$$S_t = S + aW$$

Persamaan 9. Menambahkan nilai *watermark* ke matriks singular

dimana a adalah faktor pengali yang menentukan kekuatan *watermark*.

- Dari S_t yang didapat, dicari kembali nilai singularnya yaitu :

$$S_t = U_w S_w V_w^T$$

Persamaan 10. Mencari nilai matriks singular dari suatu citra setelah diberi *watermark*

- S_w yang diperoleh ini kemudian digunakan untuk menyisipkan *watermark* bersama dengan matriks U dan V di awal.

$$A_w = U S_w V^T$$

Persamaan 11. Menyisipkan *watermark*

2. Ekstraksi *watermark*

Untuk melakukan ekstraksi, yang diperlukan adalah citra ber-*watermark* dan *watermark* asli. Citra asli tidak diperlukan dalam proses ekstraksi, namun sebagai gantinya, nilai U_w dan V_w dari S_t harus dimiliki. Proses ekstraksi *watermark* dapat dilakukan hanya dengan membalikkan langkah-langkah pada proses penyisipan saja hingga *watermark* berhasil diekstrak dari citra.

Langkah berikutnya adalah dengan membandingkan *watermark* hasil ekstraksi dengan *watermark* asli dengan rumus :

$$Err = \frac{\sum_{i=1}^N \sum_{j=1}^N w'(i,j) w(i,j)}{\|W'\| \|W\|}$$

Persamaan 12. Melakukan perbandingan *watermark* hasil ekstraksi dengan *watermark* asli.

dimana

$W' = \{w'(i,j)\}_{1 \leq i,j \leq N}$ dan

$W = \{w(i,j)\}_{1 \leq i,j \leq N}$

Apabila nilai Err masih berada dalam batas toleransi, maka *watermark* hasil ekstraksi dapat dianggap cocok dengan *watermark* asli.

III. PENGUJIAN PROGRAM

Pada pembuatan makalah ini, dibuat sebuah program yang melakukan *watermarking* dengan metode DCT dan SVD. Program dibuat dengan bahasa C# dan tools Microsoft Visual Studio 2010.

Untuk penyisipan metode DCT, program menerima masukan berupa :

- Citra yang akan disisipkan *watermark*
- *Watermark*

Sedangkan untuk ekstraksinya, program menerima masukan berupa :

- Citra asli
- Citra yang akan diujikan/diidentifikasi
- *Watermark*

Untuk penyisipan metode SVD, program menerima masukan berupa :

- Citra yang akan disisipkan *watermark*
- *Watermark*
- Kunci (untuk pembangkitan nilai acak)

Sedangkan untuk proses ekstraksi, program menerima masukan berupa :

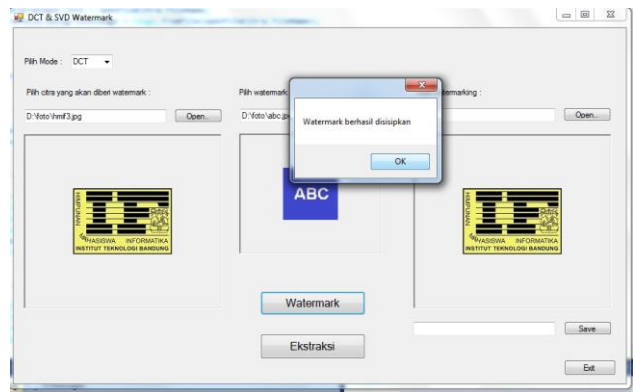
- Citra yang akan diujikan
- *Watermark* asli
- Kunci (untuk pembangkitan nilai acak)

Berikut adalah beberapa hasil percobaan yang dilakukan terhadap program.

A. Pengujian DCT

Hasil uji coba yang dilakukan untuk program ini adalah sebagai berikut :

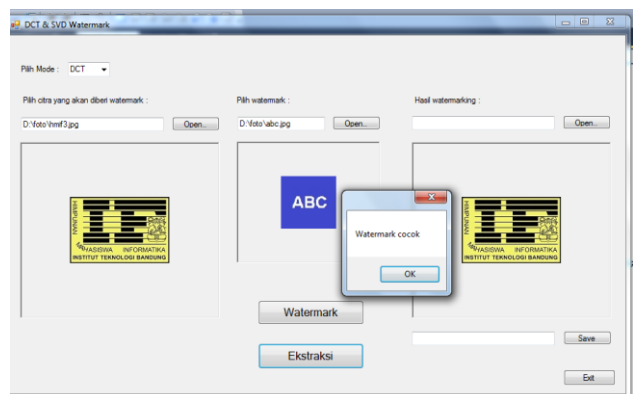
1. Proses Penyisipan



Gambar 1. Proses penyisipan *watermark* dengan metode DCT

Pada uji coba ini, *watermark* berhasil disisipkan ke dalam citra.

2. Proses Ekstraksi



Gambar 2. Proses ekstraksi *watermark* dengan metode DCT

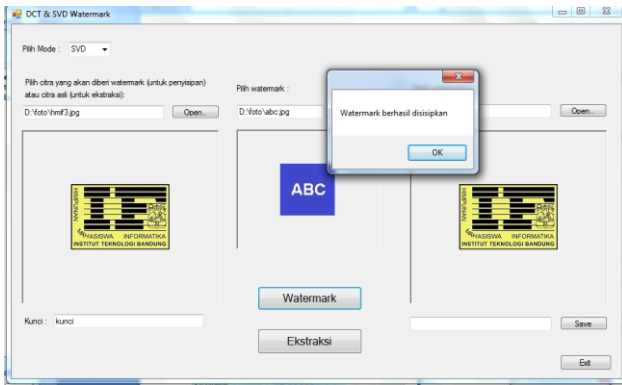
Pada uji coba ini program dapat mendeteksi bahwa terdapat *watermark* yang disisipkan ke dalam citra dan *watermark* inputan sama dengan *watermark* pada citra yang diujikan. Namun pada program ini, perhitungan perbedaan *watermark* asli dan *watermark* pada citra terkadang masih salah.

Program memerlukan waktu yang cukup lama dalam mengeksekusi *watermarking* DCT, baik dalam proses penyisipan maupun proses ekstraksi.

B. Pengujian SVD

Hasil uji coba yang dilakukan terhadap program mode SVD adalah sebagai berikut :

1. Proses Penyisipan

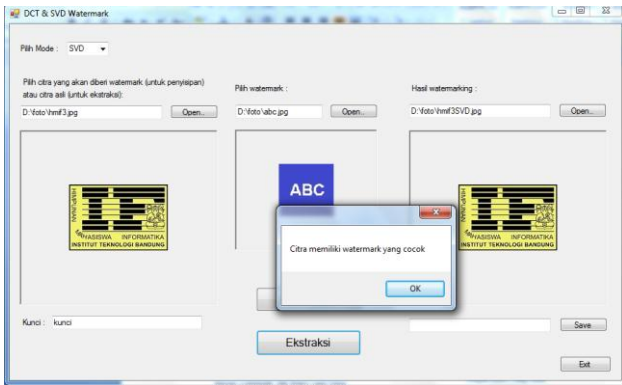


Gambar 3. Proses penyisipan *watermark* pada metode SVD

Pada proses ini program berhasil menyisipkan *watermark* ke dalam citra.

2. Proses Ekstraksi

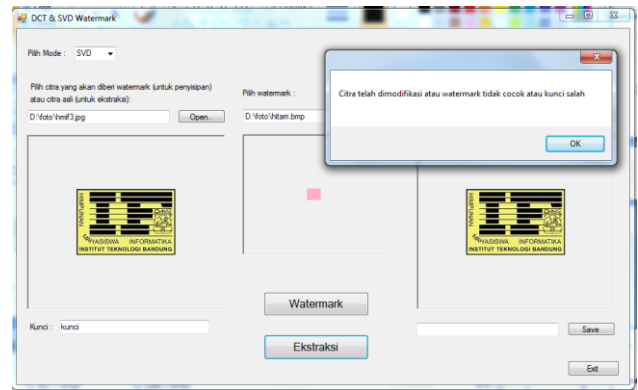
a. Kasus benar (*watermark* benar)



Gambar 4. Proses ekstraksi *watermark* dengan metode SVD dengan *watermark* dan kunci yang benar

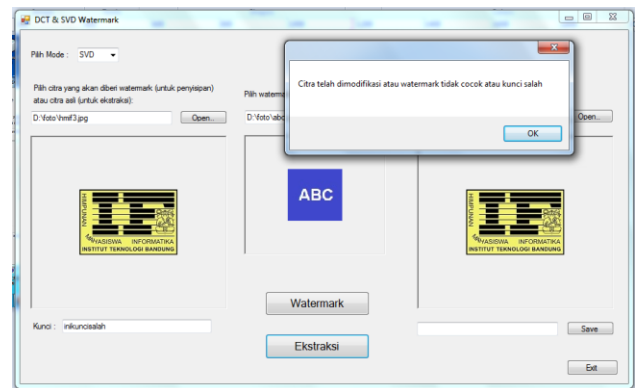
Pada uji coba ini program dapat membuktikan bahwa *watermark* yang dimasukkan oleh pengguna cocok dengan *watermark* pada citra uji.

b. Kasus salah, dimana *watermark* belum disisipkan atau *watermark* berbeda dengan watermark inputan atau citra telah dimodifikasi



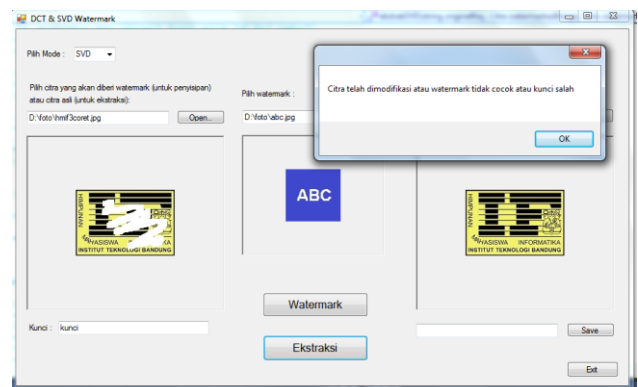
Gambar 5. Proses ekstraksi *watermark* dengan metode SVD dengan *watermark* yang salah

Program mengeluarkan pesan kesalahan karena *watermark* yang dimasukkan oleh pengguna berbeda dengan *watermark* yang disisipkan pada citra uji.



Gambar 6. Proses ekstraksi *watermark* dengan metode SVD dengan kunci yang salah.

Program mengeluarkan pesan kesalahan karena kunci yang dimasukkan oleh pengguna tidak cocok dengan kunci yang dimasukkan saat proses penyisipan dilakukan.



Gambar 4. Proses ekstraksi *watermark* dengan metode SVD pada citra yang telah diubah.

Program mengeluarkan pesan kesalahan karena citra uji yang dicoba telah dimodifikasi, yaitu dihapus sebagian.

Program memerlukan waktu yang lebih sedikit dibandingkan saat menjalankan *watermarking* DCT, baik pada proses penyisipan maupun pada proses ekstraksi.

C. PEMBAHASAN UJI COBA

Pada saat pengujian, waktu yang diperlukan oleh DCT dalam melakukan watermarking sangatlah lama. Hal ini disebabkan banyaknya operasi yang dilakukan dalam metode DCT, seperti menghitung koefisien DCT pada citra, menghitung nilai cos, dan lain-lain. Pada metode ini pun dilakukan perulangan yang sangat banyak sehingga untuk menyelesaikan proses penyisipan maupun proses ekstraksi memerlukan usaha yang lebih besar dan dibuktikan dengan waktu proses yang sangat lama. Selain itu, metode yang digunakan dalam DCT ini juga dinilai kurang efektif karena banyaknya operasi yang harus dilakukan tersebut.

Pada pembuatan program metode DCT, berbeda dengan SVD yang fungsinya telah dimiliki oleh Microsoft Visual Studio 2010, pembuat membuat fungsi DCT sendiri. Dan dari fungsi yang dibuat ini masih terdapat kekurangan, yaitu saat melakukan proses ekstraksi, perhitungan nilai perbedaan antara *watermark* yang dimasukkan oleh pengguna dan *watermark* yang tersimpan pada citra belum tepat. Hal ini dapat menyebabkan kesalahan pengambilan kesimpulan saat program dieksekusi.

Untuk metode SVD, program dapat melakukan proses penyisipan dan proses ekstraksi jauh lebih cepat dibandingkan dengan saat program melakukan proses dengan metode DCT. Hal ini dikarenakan dalam metode SVD, perhitungan yang dilakukan tidak terlalu banyak seperti halnya metode DCT, sehingga usaha yang perlu dilakukan dalam melakukan penyisipan ataupun ekstraksi tidak terlalu besar, dan waktu yang diperlukan untuk eksekusi tidak lama.

Salah satu alasan lainnya mengapa metode SVD ini lebih efektif dibandingkan dengan metode DCT adalah karena telah tersedianya fungsi SVD dari Microsoft Visual Studio 2010.

IV. KESIMPULAN

Baik dari studi literatur maupun hasil uji coba kedua metode, yaitu metode Discrete Cosine Transform (DCT) dan Singular Value Decomposition (SVD), dapat disimpulkan bahwa metode yang lebih baik digunakan adalah metode SVD. Hal ini dilihat mulai dari efisiensi rumus-rumus yang digunakan dalam melakukan penyisipan, lamanya waktu eksekusi, serta kemudahan

dalam pembuatan maupun pemakaian fungsi.

REFERENCES

- [1] Sampena, Samsu. "Autentikasi Citra menggunakan Digital Fragile Watermarking dengan Skema SVD (Singular Value Decomposition)", 2010.
- [2] Persada, Bayu Adi. "Studi dan Implementasi Non Blind Watermarking dengan Metode Spread Spectrum".
- [3] Cahyana, dkk. "Teknik Watermarking Citra Berbasis SVD".2007.
- [4] http://iatt.kememperin.go.id/tik/fullpaper/fullpaper243_wibi_n_alazhar.pdf. Tanggal akses : 22 Maret 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Maret 2011



Shofi Nur Fathiya
13508084