

IMPLEMENTASI ALGORITMA ADAPTIVE WATERMARKING PADA PELABELAN IDENTITAS FILE CITRA DIGITAL

Darwis Robinson Manalu

Fakultas Ilmu Komputer Universitas Methodist Indonesia

manaludarwis@gmail.com

Abstract

Watermarking is now well known and popular as one of the solutions to protect files and digital media by entering information such as digital signs into the core of the multimedia objects with an invisible state and reliably in conditions that exploit the contents of the digital media. The information included in the image can be text, logos or audio. Labelling is done with adaptive watermarking techniques to calculate phases of the pixel values of the logo file (watermark), calculate the DCT coefficients (Discrete Cosine Transform) digital images, digital image modifying DCT coefficients according to bit files logo, reconstruct a digital image by performing inverse DCT, extracting files logo of the image that has been watermarked. The extracted watermark data is then compared with the original watermark data by calculating the Bit Error Rate (BER), which is the ratio between the number of bits that one of the bits as a whole. From the results of the testing that has been done on the application that is designed may inserting a message in the form of the image into a digital image file, inserting a message in the form of text into image files, the extraction process penyisip of file watermarking and can do the comparison value BER (Bit Error Rate).

Keyword: Watermaking, Digital Image, Adaptive Watermarking Algorithms

I. PENDAHULUAN

Distribusi data digital berupa citra sangat pesat mengingat perkembangan internet dan dunia digital yang terus maju secara keseluruhan. Tentu saja, hal ini banyak membawa keuntungan bagi para penggunanya. Namun, tidak sedikit pula bahaya dan kerugian yang menyertainya, seperti penggantian citra tanpa izin pemiliknya, klaim kepemilikan atas sebuah karya, dan perubahan keaslian dari sebuah karya digital. Oleh karena itu, diperlukan adanya suatu cara untuk menangani masalah ini.

Watermarking belakangan ini mulai dikenal dan populer sebagai salah satu solusi untuk melindungi media digital. *Watermarking* memasukkan informasi seperti tanda digital ke dalam inti dari obyek multimedia dengan keadaan tak terlihat dan andal dalam kondisi-kondisi yang mengeksploitasi isi dari media digital tersebut. Informasi yang dimasukkan ke dalam citra dapat berupa teks, logo maupun *audio*. Informasi ini dapat digunakan untuk bermacam-macam aplikasi seperti autentikasi, perlindungan hak cipta, pemantauan data, bukti kepemilikan intelektual dan lain-lain. Pada umumnya *watermark* dapat dikelompokkan menjadi dua kelompok utama menurut domain cara kerjanya. Teknik domain spasial menanamkan data langsung dengan cara memodifikasi nilai *pixel* citra, sedangkan teknik domain frekuensi memodifikasi nilai koefisien transformasinya.

Pada penelitian ini digunakan teknik *adaptive watermarking* untuk penyisipan label pada citra digital dalam domain spasial. Teknik *adaptive watermarking* bekerja dengan cara memodifikasi tingkat intensitas dari *pixel-pixel* citra secara

adaptif, sehingga dapat menyisipkan informasi pada bit-bit citra dengan signifikan.

Proses ekstraksi mirip dengan proses penyisipan, tetapi dalam arah sebaliknya. Algoritma *Adaptive Watermarking* merupakan jenis *blind watermarking* dimana proses ekstraksi membutuhkan citra asli. Proses ekstraksi dilakukan dengan menghitung jumlah nilai intensitas piksel blok-blok dari citra asli dan citra *watermark*. Jika nilai intensitas piksel dari blok tersebut lebih besar dari pada blok citra asli, maka bit penyisip bernilai 1, sebaliknya jika nilai intensitas piksel pada blok tersebut lebih kecil daripada blok citra asli, maka bit penyisip bernilai 0. Permasalahan ydalam tulisan ini adalah bagaimana melakukan konversi label yang berformat .TXT menjadi data biner dan melakukan penyisipan label berupa logo ke dalam file citra digital dan sebaliknya bagaimana melakukan ekstraksi label penyisip dari dalam file citra untuk pembuktian kepemilikan file citra. Tujuannya adalah membuat aplikasi yang dapat memberikan identitas kepemilikan dengan cara menyisipkan logo maupun teks yang dapat dilihat ke dalam file citra yang berformat JPG maupun BMP tanpa merusak citra digital tersebut.

II. Watermarking

Ada beberapa pengertian mengenai *watermarking* yang dapat diperoleh dari berbagai referensi, baik itu dari paper-paper ilmiah maupun dari sumber-sumber lain yang penulis dapatkan dari internet. *Watermarking* merupakan sebuah proses penambahan kode secara permanen ke dalam citra digital. Penyisipan kode ini harus memiliki

ketahanan (*robustness*) yang cukup baik dari berbagai manipulasi, seperti pengubahan, transformasi, kompresi, maupun enkripsi. Kode yang disisipkan juga tidak merusak citra digital sehingga citra digital terlihat seperti aslinya. *Watermarking* dapat juga merupakan cara untuk menyisipkan *watermark* ke dalam media yang ingin dilindungi hak ciptanya. *Watermarking* merupakan proses penanaman *watermark*. Digital *watermarking* merupakan cara yang digunakan untuk menyisipkan informasi atau *watermark* pada suatu dokumen digital (Munir, 2007).

Watermark merupakan sebuah pola atau kode atau data tertentu yang membawa informasi tertentu sesuai dengan tujuannya dan sengaja ditanamkan secara permanen ke dalam data media induknya. *Watermark* dalam citra digital tersebut tidak dapat diketahui keberadaannya oleh pihak lain yang tidak mengetahui rahasia skema penyisipan *watermark*. *Watermark* tersebut juga tidak dapat diidentifikasi dan dihilangkan. Penggunaan *watermarking* sangat diperlukan untuk melindungi karya intelektual digital seperti gambar, teks, musik, video, dan termasuk perangkat lunak. Penggandaan atas produk digital yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab semakin merajalela tanpa ada ikatan hukum yang pasti sehingga merugikan pemegang hak cipta akan produk digital tersebut. Oleh karena itu, penyisipan *watermark* memiliki peran yang cukup signifikan untuk mencegah terjadinya penggandaan terhadap produk digital. (Munir, 2007).

Label *watermark* adalah sesuatu data atau informasi yang akan dimasukkan ke dalam data digital yang ingin dilakukan proses *watermarking*. Ada 2 jenis label *watermark* yang dapat digunakan:

1. Teks biasa

Label *watermark* dari teks biasanya menggunakan nilai-nilai ASCII dari masing-masing karakter dalam teks yang kemudian dipecahkan atas bit per bit. Kelemahan dari label ini adalah kesalahan pada satu bit saja akan menghasilkan hasil yang berbeda dari teks sebenarnya.

2. logo atau citra atau suara

Berbeda dengan teks, kesalahan pada beberapa bit masih dapat memberikan persepsi yang sama dengan aslinya, baik oleh pendengaran maupun penglihatan.

Oleh karena itu, penyisipan logo sebagai label *watermark* dirasakan lebih efektif dibandingkan teks, citra, ataupun suara karena selain tidak sensitif terhadap kesalahan bit, ukuran *file* juga tidak terlalu besar. Logo yang dipakai berupa logo biner atau hitam putih karena komputasi yang dibutuhkan tidak terlalu rumit namun tetap menjamin visualisasi yang cukup baik.

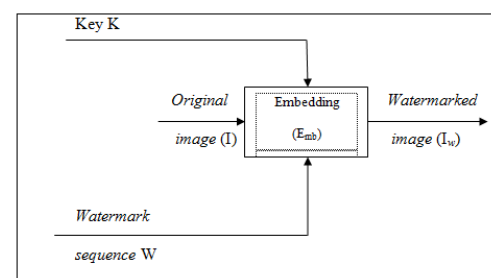
Salah satu metode yang digunakan dalam proses penyisipan label *watermark* ke suatu citra adalah metode *spread spectrum*. Metode *spread spectrum* yang diperkenalkan oleh Ingemar J. Cox didasarkan pada domain frekuensi, yaitu dengan menanamkan sejumlah urutan bilangan real

sepanjang n pada citra berukuran $N \times N$ dengan menghitung atau mentransformasikan terlebih dahulu menjadi koefisien DCT (*Discrete Cosine Transform*) $N \times N$. Untuk label *watermark* yang berupa logo atau citra biner (tidak berwarna/hitam putih), maka terlebih dahulu logo tersebut harus dikonversikan menjadi deretan bilangan biner $\{0,1\}$ untuk kemudian diubah menjadi deretan bilangan yang berisi -1 dan 1 . Selanjutnya, deretan bilangan ini nantinya akan dikali dengan bilangan *real* acak antara 0 dan 1 . Bilangan-bilangan tersebut kemudian ditanamkan pada n koefisien

Digital Watermarking

Teknik digital *watermarking* pada prinsipnya memiliki prinsip yang sama dengan *watermarking* pada media selain citra. Secara umum, *watermarking* terdiri dari dua tahapan, yaitu penyisipan *watermark* dan ekstraksi/verifikasi atau pendeteksian *watermark*. Pengekstraksian dan pendeteksian sebuah *watermark* sebenarnya tergantung pada algoritma yang digunakan untuk *watermarking*. Pada beberapa algoritma *watermarking*, *watermark* dapat diekstraksi dalam bentuk yang eksak, sedangkan pada algoritma yang lain, hanya dapat dilakukan pendeteksian *watermark* pada citra (Munir, 2004).

Secara umum proses *watermarking* pada citra digital ditunjukkan pada Gambar 2.1 dimana citra digital disisipi dengan *watermark* menggunakan kunci sebagai sarana kepemilikan untuk dapat membuka *watermark* yang disisipkan melalui *encoder* yang berisi algoritma penyisipan *watermark* kedalam citra digital.

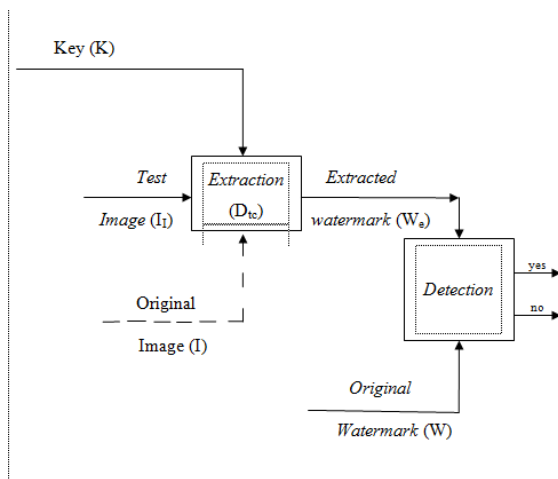


Gambar 1 Penyisipan *watermark* (*Steganografi dan Watermarking*. Makalah Departemen Teknik Informatika Institut Teknologi Bandung hal. 21-22, Munir 2004)

Citra ber-*watermark* yang dihasilkan dari proses *watermarking* tidak berbeda jauh secara visual dengan citra digital asalnya. Hal ini disebabkan karena pengubahan dari citra digital asal ke citra ber-*watermark* hanya berpengaruh sedikit terhadap perubahan warna dari citra digital, sehingga sistem penglihatan manusia tidak dapat memberi persepsi terhadap perubahan tersebut.

Proses *watermarking* perlu didukung dengan proses ekstraksi *watermark* dari citra ber-*watermark*. Proses ekstraksi/verifikasi ini bertujuan untuk mendapatkan kembali citra digital asal dan

watermark yang disisipkan dalam citra digital tersebut. Umumnya proses ekstraksi/verifikasi melibatkan proses perbandingan citra digital asal dengan citra ber-*watermark* untuk mendapatkan *watermark* yang disisipkan, seperti yang digambarkan pada Gambar 2.



Gambar 2 Ekstraksi *Watermark* (Steganografi dan *Watermarking*. Makalah Departemen Teknik Informatika Institut Teknologi Bandung hal. 21-22, Munir 2004).

Metode Adaptive Watermarking

Pada kenyataannya, menyisipkan *watermark* dengan metode *Least Significant Bit* (LSB) dari sebuah *pixel* tidak menyebabkan perbedaan untuk mata manusia. Tetapi *watermark* tidak memiliki ketahanan yang cukup dan mudah hancur menghadapi operasi pemrosesan citra yang dilakukan seperti pemfilteran *low-pass*. Oleh karena itu, untuk membuat *watermark* lebih tahan terhadap serangan, *watermark* harus ditanamkan pada bit-bit yang lebih signifikan. Hal ini akan menghasilkan lebih banyak distorsi pada citra digital dan bertentangan dengan kebutuhan akan perubahan takterlihat mata manusia. Untuk memenuhi syarat *robustness* dan *transparency*, maka secara adaptif *watermark* disisipkan dengan memodifikasi intensitas pada *pixel* tertentu sebesar-besarnya dan tidak terlihat oleh mata manusia. *Host image* atau *cover image* adalah citra yang akan disisipi *watermark* melalui proses penyisipan *watermark* (Setiadikarunia, 2008).

Penyisipan Watermark

Setelah label *watermark* diacak, setiap bit *pixel-pixel*nya disisipkan ke dalam blok-blok *pixel* pada *host image*. Ukuran dan banyaknya blok-blok *pixel* tergantung pada ukuran *host image* dan *watermark*. *Host image* dibagi menjadi blok dengan setiap bloknya disisipkan satu bit *watermark*. Misalkan *host image* X , berukuran $N \times M$ *pixel* akan disisipkan *watermark* W dengan ukuran $N \times M$, maka *host image* akan dibagi menjadi blok dengan jumlah *pixel* setiap bloknya.

Misalkan dengan ukuran *host image pixel* dan *watermark* 200×200 *pixel*, maka *host image*

dibagi menjadi 200×200 blok dengan ukuran *pixel*. Lalu setiap bit *watermark*-nya disisipkan pada 40.000 blok yang berukuran 32×32 . Setelah diacak, lalu bit-bit *watermark* bw_{ij} disisipkan pada blok B_{ij} dengan langkah-langkah sebagai berikut:

1. Bit-bit *watermark* bw_{ij} akan disisipkan berurutan sesuai dengan bloknya B_{ij} .
2. Hitung intensitas rata-rata (y_{mean}), maksimum (y_{max}), dan minimum (y_{min}) dari blok B_{ij} .
3. Hitung intensitas rata-rata m_H dan m_L .

$$m_H = \frac{(y_{max} + y_{mean})}{2}, \quad m_L = \frac{(y_{mean} + y_{min})}{2} \quad \text{nilai}$$

intensitas *pixel-pixel* pada blok B_{ij} dengan aturan berikut:

- a. 1) $bw_{ij} = 1$:

$$y' = y_{max} \quad \text{jika } y > m_H$$

$$y' = y_{mean} \quad \text{jika } m_L \leq y < y_{mean}$$

- b. $y' = y + \delta$ lainnya.

$$y' = y_{min} \quad \text{jika } y < m_L$$

$$y' = y_{mean} \quad \text{jika } y_{mean} \leq y < m_H$$

- Dim $y' = y - \delta$ lainnya.

cil

Nilai δ suatu nilai *offset* yang diatur untuk mengubah sedikit nilai intensitas citra *host*. Batasan nilai delta (δ) ini merupakan parameter yang nantinya akan menentukan tingkat ketahanan dan transparansi citra.

Ekstraksi Watermark

Proses ekstraksi *watermark* mirip dengan proses penyisipan *watermark* tetapi dalam arah sebaliknya. Algoritma ekstraksi *watermark* membutuhkan *original host image*. Proses ekstraksi dilakukan dengan menghitung jumlah nilai intensitas *pixel* blok-blok dari *host image* dan *watermarked image*. Bit-bit *watermark* bw didapat kembali dengan aturan berikut:

$$bw_{ij} = 1 \quad \text{jika } Sw_{ij} > So_{ij}$$

$$bw_{ij} = 0 \quad \text{jika } Sw_{ij} \leq So_{ij}$$

Dimana: Sw_{ij} : jumlah intensitas *pixel* blok B_{ij} dari *watermarked image*.

So_{ij} : jumlah intensitas *pixel* blok B_{ij} dari *original host image*.

Setelah bit-bit bw diperoleh, lalu disusun kembali menjadi *watermark* awal menggunakan inversi dari proses pengacakan.

Aplikasi Watermark

Watermark telah diterapkan secara luas untuk mengatasi berbagai tindak kejahatan yang berkaitan dengan dokumen digital (Munir, 2006).

Fungsi penggunaan *watermark* tersebut antara lain adalah sebagai:

1. Identifikasi kepemilikan

Sebagai identitas dari pemilik dokumen digital, identitas ini disisipkan dalam dokumen digital dalam bentuk *watermark*. Biasanya identitas kepemilikan seperti ini diterapkan melalui *visible*

- watermarking*. Contohnya *url* halaman *web* tempat suatu gambar di-*download*.
2. Bukti kepemilikan
Watermark merupakan suatu bukti yang sah yang dapat dipergunakan di pengadilan. Banyak kasus pemalsuan foto yang akhirnya terungkap karena penggunaan *watermark* ini.
 3. Memeriksa keaslian isi karya digital
Watermark juga dapat digunakan sebagai teknik untuk mendeteksi keaslian dari suatu karya. Suatu *image* yang telah disisipi *watermark* dapat dideteksi perubahan yang dilakukan terhadapnya dengan memeriksa apakah *watermark* yang disisipkan dalam *image* tersebut rusak atau tidak.
 4. *User authentication* atau *fingerprinting*
Seperti halnya bukti kepemilikan, *watermark* juga dapat digunakan sebagai pemeriksaan hak akses atau penanda (sidik jari) dari suatu *image*.
 5. *Transaction tracking*
Fungsi *transaction tracking* ini dapat dilakukan pada *image* yang mengandung *watermark*. Pengimplementasiannya dilakukan dengan memberikan *watermark* yang berbeda pada sejumlah domain/kelompok pengguna. Sehingga bila *image* tersebar diluar domain tersebut, dapat di ketahui domain mana yang menyebarkannya.
 6. *Piracy protection/copy*
Untuk dapat melakukan ini, perancang *watermark* harus bekerjasama tidak hanya pada masalah *software*, tetapi juga dengan vendor yang membuat *hardware*. Sehingga sebelum dilakukan *peng-copy-an*, terlebih dahulu dilakukan pemeriksaan apakah *image* tersebut boleh di-*copy* atau tidak.
 7. *Broadcast monitoring*
Dalam dunia *broadcasting/television news channel*, *watermark* biasanya disisipkan sebagai logo dari perusahaan *broadcasting* yang bersangkutan. Hal ini dilakukan untuk menandai berita yang mereka siarkan. Sehingga bila pihak lain merekam berita tersebut, maka *watermark*-nya akan otomatis terbawa.

III. METODOLOGI

- Dalam penulisan ini metode yang dilakukan adalah:
- a. Mempelajari tentang citra digital, algoritma *Adaptive Watermarking* dan *flow chart*
 - b. Analisis dan Perancangan, Melakukan analisis terhadap sistem yang akan dibangun.
 - c. Merancang sistem yang telah dianalisis pada perangkat lunak.
 - d. Pengujian dan Implementasi pengujian sistem. Dilakukan pengujian sistem apakah sudah sesuai dengan perancangan dan bebas dari kesalahan.
 - e. Dokumentasi, Pembuatan dokumen sistem dan analisis yang telah diperoleh.

IV. ANALISIS DAN PEMBAHASAN

Pemberian label dilakukan dengan teknik *adaptive watermarking*. Tahap-tahap yang dilakukan adalah:

1. Menghitung nilai piksel file logo (*watermark*).
2. Menghitung koefisien DCT citra digital.
3. Memodifikasi koefisien DCT citra digital sesuai dengan bit file logo
4. Merekonstruksi citra digital dengan melakukan invers DCT.
5. Mengekstraksi file logo dari citra yang telah ter-*watermark*.

Pada penelitian ini digunakan citra digital dengan mode warna 24 bit sebagai citra digital yang hendak dilindungi dan citra logo "UMI" dengan dimensi 32 x 32 piksel sebagai file *watermark*. Citra digital dan citra logo dapat dilihat pada Gambar 3.



Gambar 3 Citra Digital dan Logo Penyisip

Menghitung Koefisien DCT Citra

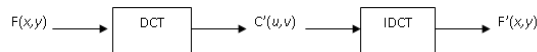
Non blind watermarking adalah verifikasi *watermark* yang membutuhkan citra yang asli sebagai pembanding keberadaan penyisip. Metode *Adaptive Watermarking* dilakukan dengan menanamkan sejumlah urutan bilangan *real* sepanjang n pada citra berukuran 8×8 piksel per blok dengan menghitung atau mentransformasikan terlebih dahulu menjadi koefisien DCT (*Discrete Cosine Transform*) $N \times N$. Untuk implementasi pada sebuah citra dengan ukuran tertentu, citra asal dibagi menjadi blok-blok berukuran 8×8 piksel. Pembagian matriks koefisien citra menjadi blok-blok ini bertujuan untuk mengurangi jumlah total operasi pengalihan dan penambahan yang dibutuhkan dan mengeliminasi kebutuhan untuk menghitung nilai *cosine* dari banyak argumen yang berbeda sehingga proses transformasi yang dilakukan dapat lebih cepat dan efisien. Nilai α yang digunakan pada persamaan dapat memiliki beberapa nilai atau terdapat satu parameter skala untuk setiap *bit watermark* yang akan disisipkan. Pada penelitian ini digunakan nilai parameter skala tunggal dengan rentang antara 0 sampai 1. Nilai α merupakan bilangan *real* positif. Untuk sebuah matriks $n \times m$, 2-D DCT dapat dihitung dengan cara 1-D DCT diterapkan pada setiap baris dari C dan kemudian hasilnya dihitung DCT untuk setiap kolomnya. 2-D DCT dapat dihitung dengan menerapkan transformasi 1-D secara terpisah pada baris dan kolomnya, sehingga dapat dikatakan bahwa 2-D DCT *separable* dalam dua dimensi.

DCT dua dimensi dapat dipandang sebagai komposisi dari DCT pada masing-masing dimensi. Sebagai contoh, jika himpunan bilangan *real* disajikan dalam *array* dua dimensi terhadap masing-masing baris dan kemudian melakukan DCT satu dimensi terhadap masing-masing kolom dari hasil

DCT tersebut. DCT dua dimensi dapat dinyatakan sebagai berikut dengan persamaan (3.3).

$$C(u,v) = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} f(i,j) \cos \left[\frac{\pi(2i+1)\mu}{2M} \right] \cos \left[\frac{\pi(2j+1)v}{2N} \right] \quad (1)$$

Penyisipan *watermark* pada citra di domain DCT dilakukan dengan cara terlebih dahulu melakukan transformasi DCT terhadap citra yang akan disisipi *watermark*. Setelah dilakukan transformasi, kemudian dilakukan modifikasi terhadap koefisien-koefisien DCT sesuai dengan *bit watermark* yang akan disisipkan. Setelah dilakukan modifikasi, dilakukan *inverse DCT* untuk mengembalikan data citra ke domain spasial. Skema proses *watermarking* di domain DCT dapat dilihat pada Gambar 4.



Gambar 4 Skema Proses *Watermarking*

Dimana:

- f = matriks nilai piksel citra
- C = matriks koefisien DCT citra asal
- W = data *watermark* yang disisipkan
- C' = matriks koefisien DCT yang sudah dimodifikasi
- F' = matriks nilai piksel sesudah penyisipan *watermark*.

A. Menghitung Matriks Transform

Menghitung matriks *transform* (A) dengan menggunakan bagian persamaan (3.3) untuk matriks citra per kolom.

$$A(i,j) = \frac{1}{\sqrt{N}} \sum_{i=0}^{M-1} \cos \left[\frac{\pi(2j+1)i}{2N} \right] C(u,v)$$

$$= \sum_{j=0}^{N-1} f(i,j) \cos \left[\frac{\pi(2j+1)i}{2N} \right]$$

$$C_i = \alpha(u)$$

$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N} \text{ dimana } C_i = \frac{1}{\sqrt{N}} (i=0), C_i =$$

$$\sqrt{\frac{2}{N}} (i > 0) \dots \dots \dots (2)$$

Karena $i=0$ maka digunakan rumus $\frac{1}{\sqrt{N}}$

$$A(0,0) = \frac{1}{\sqrt{4}} = 0.5$$

$$A(0,1) = \frac{1}{\sqrt{4}} = 0.5$$

$$A(0,2) = \frac{1}{\sqrt{4}} = 0.5$$

$$A(0,3) = \frac{1}{\sqrt{4}} = 0.5$$

$$A(0,3) = \frac{1}{\sqrt{4}} = 0.5$$

Untuk $i > 0$ maka digunakan rumus

$$A(1,0) = \sqrt{\frac{2}{4}} \cos \frac{(2*0+1)1*\pi}{2*4} = 0.7063$$

$$A(1,1) = \sqrt{\frac{2}{4}} \cos \frac{(2*1+1)1*\pi}{2*4} = 0.7069$$

$$A(1,2) = \sqrt{\frac{2}{4}} \cos \frac{(2*2+1)1*\pi}{2*4} = 0.7066$$

$$A(1,3) = \sqrt{\frac{2}{4}} \cos \frac{(2*3+1)1*\pi}{2*4} = 0.7062$$

$$A(2,0) = \sqrt{\frac{2}{4}} \cos \frac{(2*0+1)2*\pi}{2*4} = 0.7070$$

$$A(2,1) = \sqrt{\frac{2}{4}} \cos \frac{(2*1+1)2*\pi}{2*4} = 0.7065$$

$$A(2,2) = \sqrt{\frac{2}{4}} \cos \frac{(2*2+1)2*\pi}{2*4} = 0.7054$$

$$A(2,3) = \sqrt{\frac{2}{4}} \cos \frac{(2*3+1)2*\pi}{2*4} = 0.6997$$

$$A(3,0) = \sqrt{\frac{2}{4}} \cos \frac{(2*0+1)3*\pi}{2*4} = 0.7069$$

$$A(3,1) = \sqrt{\frac{2}{4}} \cos \frac{(2*1+1)3*\pi}{2*4} = 0.7057$$

$$A(3,2) = \sqrt{\frac{2}{4}} \cos \frac{(2*2+1)3*\pi}{2*4} = 0.7033$$

$$A(3,3) = \sqrt{\frac{2}{4}} \cos \frac{(2*3+1)3*\pi}{2*4} = 0.6997$$

Maka perhitungan diatas didapatkan nilai untuk matriks transform ditunjukkan pada Gambar5.

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.7063 & 0.7069 & 0.7066 & 0.7062 \\ 0.7070 & 0.7065 & 0.7054 & 0.6997 \\ 0.7069 & 0.7057 & 0.7033 & 0.6997 \end{bmatrix}$$

Gambar 5 Matriks Transform (A)

B. Menghitung Matriks Transpose

A' (matriks *transpose*) yaitu perubahan kolom menjadi baris dari matriks A seperti ditunjukkan pada Gambar 6

$$\begin{bmatrix} 0.5 & 0.7063 & 0.7070 & 0.7069 \\ 0.5 & 0.7069 & 0.7065 & 0.7057 \\ 0.5 & 0.7066 & 0.7054 & 0.7033 \\ 0.5 & 0.7062 & 0.6997 & 0.6997 \end{bmatrix}$$

Gambar 6 Matriks *Transpose* (A')

Kemudian cari matriks D dengan $D = A * M * A'$

Dimana A = Matriks Transform

Dimana M = Matriks Citra asli

Dimana A' = Matriks *Transpose*

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.7063 & 0.7069 & 0.7066 & 0.7062 \\ 0.7070 & 0.7065 & 0.7054 & 0.6997 \\ 0.7069 & 0.7057 & 0.7033 & 0.6997 \end{bmatrix}$$

X

$$\begin{bmatrix} 47 & 68 & 71 & 88 \\ 148 & 211 & 135 & 139 \\ 189 & 177 & 158 & 190 \\ 151 & 88 & 84 & 192 \end{bmatrix}$$

X

$$\begin{bmatrix} 0.5 & 0.7063 & 0.7070 & 0.7069 \\ 0.5 & 0.7069 & 0.7065 & 0.7057 \\ 0.5 & 0.7066 & 0.7054 & 0.7033 \\ 0.5 & 0.7062 & 0.6997 & 0.6997 \end{bmatrix} =$$

Perkalian matriks *transform* dengan matriks citra asli (A^*M) adalah sebagai berikut:

$$\begin{aligned} Y'(0,0) &= (0.5*47) + (0.5*148) + (0.5*189) + (0.5*151) \\ Y'(0,0) &= 267.5 \\ Y'(0,1) &= (0.5*68) + (0.5*211) + (0.5*177) + (0.5*88) \\ Y'(0,1) &= 272 \\ Y'(0,2) &= (0.5*71) + (0.5*135) + (0.5*158) + (0.5*84) \\ Y'(0,2) &= 224 \\ Y'(0,3) &= (0.5*88) + (0.5*139) + (0.5*190) + (0.5*192) \\ Y'(0,3) &= 304.5 \end{aligned}$$

$$\begin{aligned} Y'(1,0) &= (0.7063*47) + (0.7069*148) + (0.7066*189) + (0.7062*151) \\ Y'(1,0) &= 378.0009 \\ Y'(1,1) &= (0.7063*68) + (0.7069*211) + (0.7066*177) + (0.7062*88) \\ Y'(1,1) &= 384.3981 \\ Y'(1,2) &= (0.7063*71) + (0.7069*135) + (0.7066*158) + (0.7062*84) \\ Y'(1,2) &= 316.5424 \\ Y'(1,3) &= (0.7063*88) + (0.7069*139) + (0.7066*190) + (0.7062*192) \\ Y'(1,3) &= 430.2579 \end{aligned}$$

$$\begin{aligned} Y'(2,0) &= (0.7070*47) + (0.7065*148) + (0.7054*189) + (0.6997*151) \\ Y'(2,0) &= 376.7663 \\ Y'(2,1) &= (0.7070*68) + (0.7065*211) + (0.7054*177) + (0.6997*88) \\ Y'(2,1) &= 383.5769 \\ Y'(2,2) &= (0.7070*71) + (0.7065*135) + (0.7054*158) + (0.6997*84) \\ Y'(2,2) &= 315.8025 \\ Y'(2,3) &= (0.7070*88) + (0.7065*139) + (0.7054*190) + (0.6997*192) \\ Y'(2,3) &= 428.7879 \end{aligned}$$

$$\begin{aligned} Y'(3,0) &= (0.7069*47) + (0.7057*148) + (0.7033*189) + (0.6997*151) \\ Y'(3,0) &= 376.2463 \\ Y'(3,1) &= (0.7069*68) + (0.7057*211) + (0.7033*177) + (0.6997*88) \\ Y'(3,1) &= 383.0296 \\ Y'(3,2) &= (0.7069*71) + (0.7057*135) + (0.7033*158) + (0.6997*84) \\ Y'(3,2) &= 315.3556 \\ Y'(3,3) &= (0.7069*88) + (0.7057*139) + (0.7033*190) + (0.6997*192) \\ Y'(3,3) &= 428.2689 \end{aligned}$$

Dari perhitungan untuk matriks *transpose* $Y'(1,0)$ sampai $Y'(7,7)$ diperoleh matriks koefisien A^*M seperti pada Gambar 7

267.5	272	224	304.5
378	384.39	316.54	430.25
376.76	383.57	315.80	428.78
376.24	383.02	315.35	428.26

Gambar 7 Matriks Koefisien A^*M
Selanjutnya Matriks Koefisien A^*M dikalikan dengan matriks koefisien A' Gambar 8.

0.5	0.7063	0.7070	0.7069
0.5	0.7069	0.7065	0.7057
0.5	0.7066	0.7054	0.7033
0.5	0.7062	0.6997	0.6997

Gambar 8 Matriks Koefisien A'

Perkalian matriks A^*M dengan matriks A' (*Transpose*) adalah sebagai berikut:

$$\begin{aligned} X(0,0) &= (267.5*0.5) + (272*0.5) + (224*0.5) + (304.5*0.5) \\ X(0,0) &= 534 \\ X(0,1) &= (267.5*0.7063) + (272*0.7069) + (224*0.7066) + (304.5*0.7062) \\ X(0,1) &= 755.23455 \\ X(0,2) &= (267.5*0.7070) + (272*0.7065) + (224*0.7054) + (304.5*0.6997) \\ X(0,2) &= 753.058 \\ X(0,3) &= (267.5*0.7069) + (272*0.7057) + (224*0.7033) + (304.5*0.6997) \\ X(0,3) &= 752.3437 \end{aligned}$$

Untuk menghitung nilai $X(1,0)$ sampai $X(7,7)$ dilakukan dengan cara yang sama seperti di atas dan selanjutnya hasil perhitungan di atas dimasukkan ke dalam matriks koefisien DCT Gambar 9.

534	755.23	753.05	752.23
-26.29	8.30	45.83	16.08
-2.58	29.29	65	17.87
5.65	1.21	-0.88	-38.02

Gambar 9 Matriks Koefisien DCT

Matriks D sekarang berisi dengan koefisien DCT, dimana data yang akan digantikan dengan nilai piksel logo penyisip adalah frekuensi-frekuensi tertinggi dengan mengabaikan tanda minus (nilai mutlak) yaitu:

1. Piksel $f(0,1)$ dengan nilai 755.23
2. Piksel $f(1,2)$ dengan nilai 45.83
3. Piksel $f(2,3)$ dengan nilai 65
4. Piksel $f(3,3)$ dengan nilai -38.02

Modifikasi Koefisien DCT

Dari matriks Gambar 3.13 Koefisien DCT dimodifikasi menurut persamaan:

$$Ci' = Ci (1 - \alpha Wi)$$

Ci' = Koefisien DCT setelah dimodifikasi

Ci = Koefisien DCT sebelum dimodifikasi

α = Nilai modifikasi sebesar 0.1

Wi = Nilai bit *watermark* yang disisipkan.

Nilai wi adalah -1 jika bit yang akan disisipkan adalah 0, dan 1 jika bit yang disisipkan adalah 1.

W adalah nilai *watermark* (logo modifikasi) :

-1	-1	1	1
----	----	---	---

Gambar 10 Matriks Citra Logo

Untuk setiap nilai koefisien pada matriks koefisien DCT yang terbesar lakukan modifikasi dengan nilai $\alpha=0.1$:

Perhitungan hasil modifikasi adalah sebagai berikut:

$$C'(0,1) = 755.23 (1 - 0.1(-1)) = 755.23 (1.1) = 830.753$$

$$C'(1,2) = 45.83 (1 - 0.1(-1)) = 45.83 (1.1) = 50.413$$

$$C'(2,3) = 65 (1 - 0.1(1)) = 65 (0.9) = 58.5$$

$$C'(3,3) = -38.02 (1 - 0.1(1)) = -38.02 (0.9) = -34.218$$

Selanjutnya hasil perhitungan modifikasi koefisien DCT di atas dimasukkan pada matriks Y seperti pada Gambar 11.

534	830.753	753.05	752.23
-26.29	8.30	50.413	16.08
-2.58	29.29	58.5	17.87
5.65	1.21	-0.88	-34.218

Gambar 11 Matriks Modifikasi Koefisien DCT (Y)

Tahap selanjutnya adalah mengembalikan citra digital dalam bentuk visual dengan melakukan proses *Invers* DCT dengan mengalikan matriks hasil modifikasi DCT dengan matriks *transpose* ($X=A^*Y$).

$$\begin{bmatrix} 0.5 & 0.7063 & 0.7070 & 0.7069 \\ 0.5 & 0.7069 & 0.7065 & 0.7057 \\ 0.5 & 0.7066 & 0.7054 & 0.7033 \\ 0.5 & 0.7062 & 0.6997 & 0.6997 \end{bmatrix}$$

Gambar 12 Matriks *Transpose* (A')

$$\begin{aligned} X(0,0) &= (534 \cdot 0.5) + (830.753 \cdot 0.5) + (753.05 \cdot 0.5) + (852.23 \cdot 0.5) \\ X(0,0) &= 1485.01 \\ X(0,1) &= (534 \cdot 0.7063) + (830.753 \cdot 0.7069) + (753.05 \cdot 0.7066) + (852.23 \cdot 0.7062) \\ X(0,1) &= 2098.37 \\ X(0,2) &= (534 \cdot 0.7070) + (830.753 \cdot 0.7065) + (753.05 \cdot 0.7054) + (852.23 \cdot 0.6997) \\ X(0,2) &= 2091.97 \\ X(0,3) &= (534 \cdot 0.7069) + (830.753 \cdot 0.7057) + (753.05 \cdot 0.7033) + (852.23 \cdot 0.6997) \\ X(0,3) &= 2089.67 \end{aligned}$$

Perhitungan di atas dilanjutkan sampai $X(7,7)$ dan selanjutnya hasil perhitungan di atas dimasukkan ke dalam matriks X seperti pada Gambar 13.

$$\begin{bmatrix} 1485.01 & 2098.37 & 2091.97 & 2089.67 \\ 250.56 & 9.08 & 46.15 & 17.24 \\ 456.59 & 62.92 & 8.26 & 18.08 \\ 342.15 & 46.38 & -34.36 & -22.80 \end{bmatrix}$$

Gambar 13 Matriks X

Dari matriks Gambar 13 di atas kalikan dengan matriks transform seperti pada Gambar 14.

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.7063 & 0.7069 & 0.7066 & 0.7062 \\ 0.7070 & 0.7065 & 0.7054 & 0.6997 \\ 0.7069 & 0.7057 & 0.7033 & 0.6997 \end{bmatrix}$$

Gambar 14 Matriks Transform

$$\begin{aligned} X(0,0) &= (1485.01 \cdot 0.5) + (2098.37 \cdot 0.7063) + (2091.97 \cdot 0.7070) + (2089.67 \cdot 0.7069) \\ X(0,0) &= 5180.79 \\ X(0,1) &= (1485.01 \cdot 0.5) + (2098.37 \cdot 0.7069) + (2091.97 \cdot 0.7065) + (2089.67 \cdot 0.7057) \\ X(0,1) &= 5178.49 \\ X(0,2) &= (1485.01 \cdot 0.5) + (2098.37 \cdot 0.7066) + (2091.97 \cdot 0.7054) + (2089.67 \cdot 0.7033) \\ X(0,2) &= 5170.55 \\ X(0,3) &= (1485.01 \cdot 0.5) + (2098.37 \cdot 0.7062) + (2091.97 \cdot 0.6997) + (2089.67 \cdot 0.6997) \\ X(0,3) &= 5150.26 \end{aligned}$$

Perhitungan di atas dilanjutkan sampai $X(7,7)$ dan dari hasil perhitungan IDCT di atas maka dapat diperoleh matriks citra IDCT seperti pada Gambar 15.

$$\begin{bmatrix} 5180.79 & 5178.49 & 5170.55 & 5150.26 \\ 148 & 211 & 135 & 139 \\ 189 & 177 & 158 & 190 \\ 151 & 88 & 84 & 192 \end{bmatrix}$$

Gambar 15 Matriks Citra IDCT Citra

Proses Ekstraksi

Untuk melakukan operasi ekstraksi *watermark*, diperlukan citra uji (ber-*watermark*) dan citra asli. Dengan melakukan transformasi *DCT* pada

kedua citra, maka dapat diketahui koefisien *DCT* per blok untuk setiap citra. Perbedaan atau selisih nilai pasti akan terdapat pada sejumlah koefisien jika memang telah terjadi proses penyisipan *watermark* pada citra tersebut. Nilai koefisien citra yang diekstraksi seperti ditunjuk pada Gambar 16.

$$\begin{bmatrix} 5180.79 & 5178.49 & 5170.55 & 5150.26 \\ 148 & 211 & 135 & 139 \\ 189 & 177 & 158 & 190 \\ 151 & 88 & 84 & 192 \end{bmatrix}$$

Gambar 16 Nilai Koefisien *DCT* Citra *Watermarking*

Selanjutnya dilakukan transformasi citra asli sebagai pembandingan untuk mengetahui keberadaan piksel *watermark* yang ditunjuk pada Gambar 17

$$\begin{bmatrix} 534 & 830.753 & 753.05 & 752.23 \\ -26.29 & 8.30 & 50.413 & 16.08 \\ -2.58 & 29.29 & 58.5 & 17.87 \\ 5.65 & 1.21 & -0.88 & -34.218 \end{bmatrix}$$

Gambar 17 Nilai Koefisien *DCT* Citra Asli

Untuk melihat keberadaan *watermark* pada citra Gambar 16 dan 17, maka dilakukan perhitungan sebagai berikut:

Jika sebuah koefisien *DCT* pada citra uji atau ber-*watermark* lebih besar daripada koefisien *DCT* pada citra asli berarti pada koefisien tersebut disisipkan *bit* 1. Hal ini dikarenakan penyisipan *bit* positif 1 menyebabkan koefisien termodifikasi bernilai lebih besar daripada koefisien awal.

Jika suatu nilai koefisien *DCT* citra uji lebih kecil dari koefisien *DCT* citra asal berarti pada koefisien tersebut disisipkan *bit* 0 karena penyisipan *bit* 0 akan mengurangi nilai koefisien citra asal.

Dari hasil perbandingan matriks koefisien Gambar 16 dan 17 di atas, maka posisi penyisipan terdapat pada piksel (2,2). Sehingga hasil ekstraksi penyisip berupa nilai biner piksel penyisip dapat diketahui seperti pada Gambar 18.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Gambar 18 Hasil Ekstraksi Penyisip

Dari hasil ekstraksi penyisip hanya diperoleh 1 nilai piksel dan untuk nilai piksel lainnya dapat dilakukan pada blok citra 4x4 piksel selanjutnya dengan melakukan perhitungan koefisien citra seperti di atas.

Data *watermark* yang diekstraksi tersebut kemudian dibandingkan dengan data *watermark* asli dengan menghitung *Bit Error Rate (BER)*, yaitu perbandingan antara *bit* yang salah dengan banyaknya *bit* secara keseluruhan. Persamaan *BER* dijabarkan sebagai berikut:

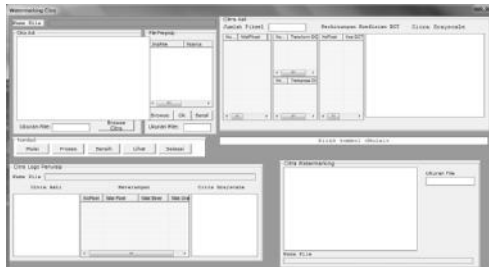
$$BER(W, W') = \frac{\sum P_i}{N} \dots (3)$$

w adalah *watermark* asli dan w' adalah *watermark* yang diekstraksi. N adalah banyaknya bit dan nilai p_i didefinisikan sebagai berikut:

$p_i = 1'$ untuk $w_i \neq w_i'$, $p_i = 0$ untuk $w_i = w_i'$. (3.6)
 Dari matriks Gambar 24 di atas diperoleh:
 $N= 4$ dan $\sum P_i= 1$
 $BER(W,W') = 2/4= 0.5$

V. HASIL DAN PROTOTYPE

Tampilan aplikasi Watermarking Citra merupakan tampilan yang berguna untuk melakukan watermarking Citra dan citra logo. Tampilan Watermarking Citra dapat dilihat pada Gambar 19.



Gambar 19 Tampilan Watermarking Citra

Tampilan Hasil Pengujian adalah tampilan hasil proses penyisipan pesan ke dalam file citra asli. Tampilan Hasil Pengujian dapat dilihat pada Gambar 20.

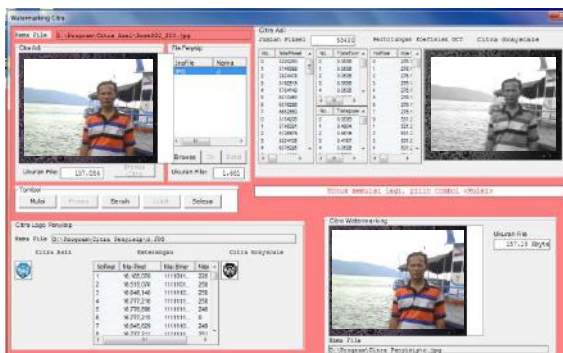
ID	Nama File Asli	Ukuran	Nama File Watermarking	Ukuran	File Hasil Penyisip	Ukuran	File Hasil Ekstraksi	Ukuran	BER
1	Bendera100_200.jpg	175.83 KByte	Bendera100.jpg	175.94 KByte	Klyte Merah&Laga.jpg	1.66 KByte	HasilEkst.JPG	1.03 KByte	0.933
2	Coba100_200.jpg		Coba100.bmp		tanu.jpg		Coba100.bmp		
3	April.jpg		Coba100_1.bmp		Mohob&Laga.jpg		Coba100_1.bmp		
4	Bendera_1.bmp		Bendera_12.bmp				SirToko		
5	Rasa100_200.jpg		namaKanan.bmp		Mohob&Laga.jpg		namaKanan.bmp		
6	Bendera100_200.jpg	175.83 KByte	Bendera100.jpg	175.94 KByte	Klyte Merah&Laga.jpg	1.66 KByte			
7	Bendera100_200.jpg	175.83 KByte	HasilTanBendera.jpg	175.94 KByte	Mohob&Laga.jpg	1.66 KByte	LogoHasil.JPG	1.03 KByte	0.933
8	Bendera100_200.bmp		Bendera100_200-A.bmp				SirToko		
9	Bendera100_200.jpg	175.83 KByte	HasilTanWm.jpg	175.94 KByte	Mohob&Laga.jpg	1.66 KByte			
10	Rasa100_200.jpg		HasilWaVisu.bmp		Mohob&Laga.jpg		HasilWaVisu.bmp		

Gambar 20 Tampilan Hasil Pengujian

Pengujian Program

Pengujian program adalah tampilan program untuk melakukan watermarking dan ekstraksi dengan data yang bervariasi.

Watermarking Citra Invisible

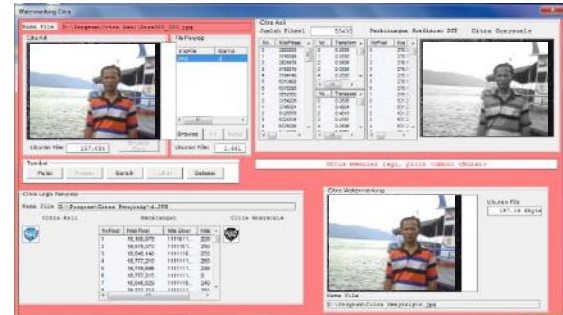


Gambar 21 Tampilan Watermarking Citra Invisible

Keterangan:

- Nama file citra Asli : BungaMawar.jpg
- Ukuran file citra asli : 157.084 Kb
- Nama file penyisip : Sisip.jpg
- Ukuran file penyisip : 8.492 Kb
- Nama file Hasil penyisipan : Hasilsip.jpg
- Ukuran file Hasil penyisipan : 157.19 Kb

Ekstraksi Citra Invisible



Gambar 22 Tampilan Help

- Nama File Watermark : Bunga.jpg
- Ukuran file Watermark : 157.19 Kb
- Nama File Penyisip : BungaPlusLogo.jpg
- Ukuran file : 2.865 Kb.

VI. PENUTUP

Kesimpulan

Dari hasil pengujian yang telah dilakukan pada aplikasi yang dirancang dapat diberikan beberapa kesimpulan antar lain: 1) Dapat melakukan penyisipan pesan berupa citra ke dalam file citra digital; 2) Dapat melakukan Penyisipan pesan berupa teks ke dalam file citra; 3) Proses Ekstraksi penyisip dari file watermarking dan 4) dapat melakukan perbandingan nilai BER (Bit Error Rate)

Saran

Adapun saran yang dapat dilakukan untuk memodifikasi sistem selanjutnya adalah agar dapat melakukan perbandingan hasil watermarking dengan algoritma yang lain.

DAFTAR PUSTAKA

Jogiyanto. 2005. *Analisis dan desain Sistem Informasi*. Penerbit: ANDI. Yogyakarta.

Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Penerbit Informatika.

Munir, Rinaldi. 2004. *Steganografi dan Watermarking*. Makalah Departemen Teknik Informatika Institut Teknologi Bandung 2004

Munir, Rinaldi. 2007. *Skema Asymmetric Watermarking Berbasis Uji Korelasi*.

- Munir, Rinaldi. 2006. *Sekilas Image Watermarking untuk Memproteksi Citra Digital dan Aplikasinya pada Citra Medis*. Makalah Sekolah Tinggi Elektro dan Informatika. Bandung.
- Setiadarunia, Daniel dan Danandy, Yohanes. 2008. Teknik Adaptive Watermarking Pada Domain Spasial Untuk Penyisipan Label Pada Citra Digital. Seminar Nasional Sains dan Teknologi-II 2008 Universitas Lampung, 17-18 November 2008.
- Persada, Bayu Adi. 2004. Studi dan implementasi *non blind watermarking* Dengan metode *spread spectrum*, Bandung: Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
- Yudi Dwiandiyanta, 2015, Perbandingan Watermarking Citra dengan Alihragam Wavelet dan Discrete Cosine Transform, *Jurnal Buana Informatika*, Universitas Atma Jaya Yogyakarta
- Wardana, 2005. *Pembuatan Kontrol ActiveX di Visual Basic 6*. Jakarta. Penerbit Elex Media Komputindo.