Penggunaan *Watermarking* untuk Menandai Hasil Karya *AI* dan Hasil Karya *Artist* agar Tidak digunakan untuk *Training AI*

David Owen Adiwiguna - 13519169
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519169@std.stei.itb.ac.id

***Abstract*—Baru-baru ini dunia dihebohkan dengan kemunculan AI Art Generator yang bisa menghasilkan hasil karya yang indah walaupun tidak sempurna. Seiring dengan berkembangkan AI Art Generator menjadi lebih baik lagi, mulai banyak muncul pertanyaan terutama terkait para *digital artist* yang merasa posisinya akan tergantikan dengan AI. Makalah ini bertujuan untuk mencoba menggunakan *watermark* dengan memanfaatkan *robust watermarking* menggunakan DCT dan DWT secara bersamaan untuk menanggulangi masalah yang timbul akibat AI Art Generator.**

Keywords—watermark; AI Art Generator; robust watermarking; dct-dwt;

I. Introduction (*Heading 1*)

Baru-baru ini dunia sempat heboh dengan munculnya berbagai terobosan teknologi berbasis *Artificial Intelligence* (AI) yang dapat membuat sebuah gambar atau ilustrasi hanya berdasarkan deskripsi yang diberikan pengguna atau bisa juga disebut AI Art Generator. Beberapa dari AI Art Generator yang tersedia juga menyediakan fitur untuk membuat sebuah ilustrasi berdasarkan referensi gambar yang diberikan oleh pengguna, hal ini bisa dipergunakan untuk menghasilkan ilustrasi yang spesifik, misalnya seorang karakter dalam sebuah permainan, atau bahkan sebuah karakter yang baru. Orang-orang sangat tertarik akan adanya teknologi ini yang seakan-akan tiba-tiba muncul ke permukaan, terutama karena kebanyakan dari aplikasi AI Art Generator tersebut gratis untuk digunakan atau setidaknya memiliki versi gratisnya.

Saat awal-awal kemunculan AI Art Generator, gambar yang dihasilkan masih bisa dikatakan sangat buruk, dimana terjadi kecacatan di berbagai macam tempat. Namun, seiring berkembangnya teknologi AI Art Generator semakin maju, masalah adanya cacat ini semakin terselesaikan, apalagi dengan adanya fitur untuk seseorang melakukan edit atau koreksi manual terhadap gambar yang dihasilkan. Bahkan, ada seseorang yang melakukan submisi gambar AI ke dalam sebuah kompetisi fotografi dan berhasil memenangkannya, orang tersebut pun lantas menolak dijadikan juara dan mengakui perbuatannya. Kasus ini terjadi di tengah-tengah perdebatan yang sangat panas antara penggunaan AI.

Untuk membuat sebuah AI yang baik, pasti akan diperlukan data yang sangat banyak untuk melatihnya, dan data tersebut tentu saja diambil dari berbagai macam sumber yang tersedia di internet hasil karya dari para *artist* seperti seniman, fotografer, desainer, ilustrator dan berbagai profesi yang terkait dengan seni. Ironisnya, eksistensi AI Art Generator ini justru menjadi hal yang merugikan bagi orang-orang tersebut, karena keahliannya seakan-akan terancam untuk digantikan oleh mesin, padahal merekalah yang menyediakan data-data untuk melakukan *training* AI tersebut. Selain itu, dari sisi lainnya ada oknum-oknum nakal yang menjual gambar yang dihasilkan oleh AI sebagai hasil karyanya sendiri, hal ini tentu juga menimbulkan pertanyaan tersendiri akan legalitas perbuatannya tersebut, dan hak cipta gambar yang dihasilkan oleh AI.

Makalah ini dibuat sebagai sebuah *proof of concept* dimana *watermarking* bisa dipergunakan untuk menyelesaikan masalah-masalah yang timbul dari munculnya AI Art Generator dengan menggunakan *watermark* pada media digital seseorang yang menandai bahwa media ini tidak boleh dipergunakan untuk *training* AI dan juga untuk memberikan *watermark* kepada media yang dihasilkan oleh AI. Namun, agar hal ini bisa dilakukan maka semua orang harus menggunakan algoritma *watermarking* yang sama agar bisa mendeteksi *watermark* yang ada.

II. Dasar Teori

A. *Watermark*

Watermark adalah sebuah penanda kepemilikan sebuah konten atau sebagai penanda keaslian sebuah konten. *Watermark* bisa diletakan di dalam berbagai media, seperti

audio, gambar, video, dan lain-lain. *Watermark* yang baik tidak akan merusak atau mengubah media dimana *watermark* tersebut disisipkan secara drastis.

Watermark bisa dibagi menjadi 2 jenis berdasarkan kekuatannya, yaitu :

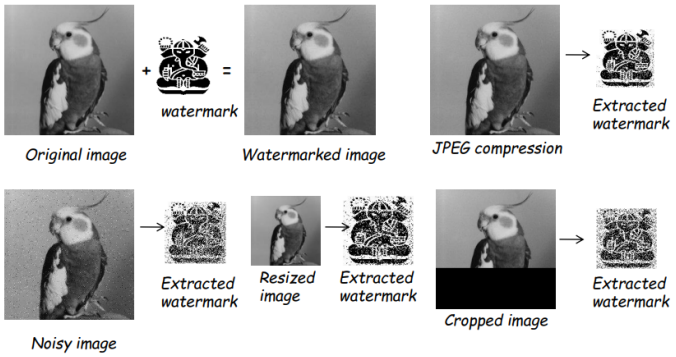
1) *Fragile Watermarking*

Fragile Watermarking adalah sebuah *watermark* yang relatif mudah untuk dirusak ketika terjadi sebuah perubahan kepada media yang dipakai. Jenis ini biasanya dipakai untuk mendeteksi apabila media sudah diubah.

Gambar 1. Contoh *Fragile Watermarking* dan *Insertion Attack*

2) *Robust Watermarking*

Robust Watermarking adalah sebuah *watermark* yang kuat terhadap manipulasi atau perubahan atas media. Jenis ini biasa dipakai untuk membuktikan hak cipta atau kepemilikan dari media tersebut.



Gambar 2. Contoh *Robust Watermarking* dan beberapa jenis perubahan citra yang dilakukan

B. *Discrete Wavelet Transform*

Discrete Wavelet Transform (DWT) adalah teknik watermarking yang menggunakan frekuensi. Sebuah sinyal akan dipecah menjadi 2 jenis, low-pass dan high-pass, dimana low-pass memuat fitur dari gambar yang kasar dan high-pass memuat fitur dari gambar yang halus, biasanya merupakan *edge* pada gambar. Pada kasus ini, DWT akan diaplikasikan 2 kali, sehingga akhirnya akan ada 4 buah sub-band yaitu LL, LH, HL, HH dan setiap sub-band bisa dipecah lagi untuk menghasilkan koefisien yang lebih detail.

C. *Discrete Cosine Transform*

Discrete Cosine Transform (DCT) adalah teknik untuk mengubah sebuah citra dari ranah spasial (dimana setiap pixel merepresentasikan lokasi pada citra) ke ranah transform (frekuensi) dengan menggunakan rumus berikut

$$C(u,v) = \alpha_u \alpha_v \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x,y) \cos\frac{\pi(2x+1)u}{2M} \cos\frac{\pi(2y+1)v}{2N}$$

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}} & ,u = 0 \\ \sqrt{\frac{2}{M}} & ,1 \le u \le M-1 \end{cases} \qquad \alpha_v = \begin{cases} \frac{1}{\sqrt{N}} & ,v = 0 \\ \sqrt{\frac{2}{N}} & ,1 \le v \le N-1 \end{cases}$$

Dan juga menggunakan rumus berikut untuk mengembalikannya ke ranah spasial dari frekuensi

$$I(x,y) = \alpha_u \alpha_v \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u,v) \cos\frac{\pi(2x+1)u}{2M} \cos\frac{\pi(2y+1)v}{2N}$$

III. Rancangan dan Implementasi

Algoritma melakukan *encode watermarking* dan *decode* akan menggunakan kombinasi dari DCT dan DWT dengan tujuan menutupi kekurangan masing-masing teknik dan menghasilkan *watermarking* yang baik. DWT yang digunakan disini adalah Haar wavelet.

A. *Encode*

Berikut ini adalah proses yang dilalui untuk melakukan *encoding watermark* ke dalam sebuah gambar

1. Membagi gambar menjadi 4 buah sub-band dengan DWT
2. Mengubah masing-masing sub-band dengan menggunakan DCT
3. Memasukkan bit-bit *watermark* ke dalam maksimum *non-trivial* koefisien
4. Mengembalikan setiap sub-band dengan reverse DCT
5. Mengembalikan sub-band menjadi gambar yang utuh kembali menggunakan DWT

B. *Decode*

Berikut ini adalah proses yang dilalui untuk melakukan *decode* untuk mengekstraksi *watermark* dari gambar

1. Membagi gambar menjadi 4 buah sub-band dengan DWT
2. Mengubah masing-masing sub-band dengan menggunakan DCT
3. Melakukan *decode* dari maksimal *non-trivial* koefisien
4. Hasil *decode* pada bagian 3 digunakan untuk menentukan bit *watermark* final

IV. Pengujian dan Analisis

Pengujian dilakukan dengan mencoba berbagai jenis gambar untuk dilakukan *encoding* dan *decoding watermark*, selain itu juga akan dilakukan berbagai jenis serangan untuk mengubah gambar untuk salah satu gambar saja.

A. Pengujian *encoding* dan *decoding*

Pengujian bagian ini akan berfokus kepada tingkat keberhasilan algoritma untuk melakukan *watermarking* dan juga melakukan *decoding* untuk watermarking tersebut dan juga diukur kualitas dari gambarnya sebelum dan setelah disisipkan *watermark* dengan menggunakan Peak Signal to Noise Ratio (PSNR). Ada 5 gambar yang akan diuji pada bagian ini, terurut dari nomor 1 sampai 1 dari kiri ke kanan.

$$\text{PSNR}_{\text{dB}} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_{\text{I}}^2}{\text{MSE}}\right)$$

$$= 20 \cdot \log_{10}\left(\frac{\text{MAX}_{\text{I}}}{\sqrt{\text{MSE}}}\right)$$

Gambar 3. Contoh Gambar yang akan diuji

Pada pengujian gambar 1, algoritma berhasil untuk melakukan *watermarking*, namun gagal melakukan *decoding* karena alasan yang tidak diketahui. Gambar sebelum dan setelah diberi *watermark* memiliki PSNR sebesar 41,129 dB.

Gambar 4. Perbandingan gambar asal (kiri), dan gambar yang telah disisipkan *watermark* (kanan) pada pengujian gambar 1.

Pada pengujian gambar 2, algoritma berhasil untuk melakukan *watermarking*, namun ketika melakukan *decoding*, *watermark* yang dihasilkan tidak sesuai dengan yang pertama disisipkan, dari *watermark* yang berupa kata ‘tester’ menjadi ‘tgstes’, sama seperti kasus sebelumnya penyebab dari kegagalan ini juga tidak diketahui. Gambar sebelum dan setelah diberi *watermark* memiliki PSNR sebesar 44,802 dB.

Gambar 5. Perbandingan gambar asal (kiri), dan gambar yang telah disisipkan *watermark* (kanan) pada pengujian gambar 2.

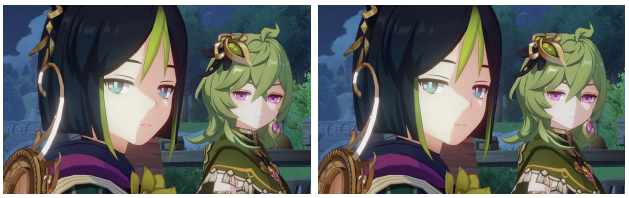
Pada pengujian gambar 3, algoritma berhasil untuk melakukan *watermarking*, namun gagal melakukan *decoding*, alasannya diduga karena gambar asal pada pengujian ini berupa PNG dan bukan JPG seperti gambar lainnya. Gambar sebelum dan setelah diberi *watermark* memiliki PSNR sebesar 45,209 dB.

Gambar 6. Perbandingan gambar asal (kiri), dan gambar yang telah disisipkan *watermark* (kanan) pada pengujian gambar 3.

Pada pengujian gambar 4, algoritma berhasil untuk melakukan *watermarking*, algoritma juga berhasil melakukan *decoding* secara sempurna, karena itulah gambar ini yang akan dipakai di pengujian bagian B nanti. Gambar sebelum dan setelah diberi *watermark* memiliki PSNR sebesar 44,710 dB.

Gambar 7. Perbandingan gambar asal (kiri), dan gambar yang telah disisipkan *watermark* (kanan) pada pengujian gambar 4.

Pada pengujian gambar 5, algoritma berhasil untuk melakukan *watermarking*, namun gagal melakukan *decoding* karena alasan yang tidak diketahui. Gambar sebelum dan setelah diberi *watermark* memiliki PSNR sebesar 43,393 dB.



Gambar 8. Perbandingan gambar asal (kiri), dan gambar yang telah disisipkan *watermark* (kanan) pada pengujian gambar 5.

Dari pengujian pertama ini, bisa dilihat banyak gambar yang tidak dapat di-*decode* kembali tanpa alasan yang jelas dan akan diperlukan penelitian lebih lanjut. Semua penambahan *watermark* menghasilkan PSNR sebesar 40-45 dB. Penambahan watermark itu sendiri tidak dapat terlihat langsung oleh mata.

B. Pengujian terhadap serangan

Pada pengujian bagian ini, gambar yang telah disisipkan *watermark* akan berusaha untuk diubah dengan berbagai macam teknik dan dilihat apakah algoritma masih bisa melakukan *decoding* dengan baik.

1) *Compression*

Pada pengujian ini, gambar yang sudah disisipkan *watermark* dimasukkan ke dalam website image *compression* yang tersedia di internet. Hasil dari pengujian ini adalah algoritma bisa melakukan *decoding* dengan benar meskipun dilakukan *compression* sebesar 60%.

2) *Noise*

Pada pengujian ini, gambar yang sudah disisipkan *watermark* dimasukkan ke dalam website penambah *noise* ke gambar yang tersedia di internet. Hasil dari pengujian ini adalah algoritma hanya bisa melakukan *decoding* dengan benar pada *noise* yang tingkatnya sangat rendah, pengujian dilakukan di 2 tingkatan, yaitu pada 5% dan 20%, dimana hanya gambar dengan *noise* 5% saja lah yang dapat di-*decode*.

Gambar 8. Perbandingan 3 tingkat *noise*, gambar asal (kiri), 5% *noise*(tengah), 20% *noise* (kanan)

3) *Brightness*

Pada pengujian ini, gambar yang sudah disisipkan *watermark* dimasukkan ke dalam website pengatur *brightness* gambar yang tersedia di internet. Hasil dari pengujian ini adalah algoritma bisa melakukan *decoding* pada gambar yang tingkat *brightness*-nya tidak jauh berubah, pengujian dilakukan dengan menambah *brightness* sebanyak 20% dan 50%, dimana hanya gambar dengan penambahan *brightness* 20% lah yang dapat di-*decode*.

Gambar 9. Perbandingan 3 tingkat *brightness*, gambar asal (kiri), penambahan 20% *brightness* (tengah), penambahan 50% *brightness* (kanan)

4) Penambahan gambar

Pada pengujian ini, gambar yang sudah disisipkan *watermark* ditambah gambar lain di atasnya berbentuk block berwarna dengan menggunakan aplikasi pengeditan gambar. Hasil dari pengujian yang dilakukan adalah algoritma cukup tahan dengan penambahan gambar lain, dimana pada contoh dibawah algoritma baru gagal melakukan *decode* pada contoh ketiga.

Gambar 10. Penambahan gambar

5) Penghapusan gambar

Pada pengujian ini, gambar yang sudah disisipkan *watermark* dihapus sebagian menggunakan aplikasi pengeditan gambar. Hasil dari pengujian yang dilakukan adalah algoritma cukup tahan dengan penambahan gambar lain, dimana pada contoh dibawah algoritma baru gagal melakukan *decode* pada contoh keempat.

Gambar 11. Penghapusan gambar

6) Pemotongan gambar

Pada pengujian ini, gambar yang sudah disisipkan *watermark* dipotong sebagian menggunakan aplikasi pengeditan gambar. Hasil dari pengujian yang dilakukan adalah algoritma hanya bisa melakukan *decoding* pada gambar yang dipotong sedikit dan sangat rentan ketika dipotong lebih dari satu sisi. Pada gambar dibawah, gambar pertama adalah gambar aslinya, gambar kedua adalah gambar yang dipotong bagian bawahnya namun masih berhasil dilakukan *decode*, gambar ketiga adalah gambar yang dipotong bagian bawahnya dan sudah tidak dapat dilakukan *decode*, dan gambar keempat adalah gambar yang bagian bawah dan kanannya dipotong sangat sedikit namun sudah tidak dapat dilakukan *decode*

Gambar 12. Pemotongan gambar

7) Pengubahan ukuran gambar

Pada pengujian ini, gambar yang sudah disisipkan *watermark* diubah ukurannya menggunakan aplikasi pengeditan gambar. Hasil dari pengujian yang dilakukan adalah algoritma sama sekali tidak bisa melakukan *decode* setelah adanya perubahan ukuran gambar.

8) Pemutaran gambar

Pada pengujian ini, gambar yang sudah disisipkan *watermark* diubah putarannya menggunakan aplikasi pengeditan gambar. Hasil dari pengujian yang dilakukan adalah algoritma sama sekali tidak bisa melakukan *decode* setelah adanya perubahan putaran gambar.

Dari pengujian kedua, bisa dilihat bahwa algoritma sangat sensitif terhadap perubahan ukuran atau putaran gambar, sementara itu algoritma bisa mentoleransi sedikit pemotongan gambar, perubahan *brightness* dan penambahan *noise*, sementara itu algoritma bisa dibilang cukup kuat untuk menangani serangan dengan *compression*, penghapusan gambar dan juga penambahan gambar.

V. Kesimpulan dan Saran

Watermarking adalah sebuah teknik yang sudah sejak lama terbukti dapat bertahan mengalami berbagai macam perubahan gambar dan dapat dipergunakan untuk mendeteksi keaslian gambar. Dalam eksperimen ini, teknik yang digunakan adalah *robust watermarking* dengan menggunakan gabungan dari DWT dan DCT. DWT yang digunakan adalah Haar wavelet, yang disebut sebagai wavelet paling mudah.

Dari hasil eksperimen yang dilakukan, algoritma berhasil bertahan atas beberapa serangan seperti *image compression*, penambahan sedikit *noise*, penambahan gambar lain di atas gambar yang memiliki *watermark*, dan juga perubahan brightness yang tidak ekstrim, penghapusan sebagian gambar, pemotongan sedikit bagian dari gambar. Sementara itu, algoritma gagal total terhadap serangan yang mengubah ukuran gambar atau perputaran gambar. Maka dari itu, bisa dibilang bahwa algoritma yang ada sejauh ini masih cukup buruk.

Kedepannya, jika hal ini ingin diterapkan untuk menyelesaikan masalah tentang AI Art Generator, maka *robust watermarking* yang dipakai harus benar-benar bagus dan tidak boleh terdapat celah untuk merusak atau bahkan menghapus *watermark* yang ada. Hal ini mungkin bisa dilakukan dengan menggunakan DWT yang lebih kompleks atau lebih mendalam.

Barulah kemudian hal ini dapat membantu untuk menyelesaikan masalah atas penggunaan AI Art Generator, yaitu dengan memastikan bahwa semua gambar buatan AI memiliki *watermark* yang bisa dipastikan keasliannya dan juga memberikan *watermark* kepada gambar buatan para *artist*, agar gambarnya tidak bisa dijadikan bahan pelatihan AI tanpa mendapat ijin. Namun, hal ini juga memerlukan semua penyedia AI Art Generator untuk mematuhi penggunaan *robust watermarking* tersebut dan juga harus menggunakan algoritma *robust watermarking* yang sama agar proses pengecekan apakah gambar tersebut merupakan buatan AI atau bukan menjadi mudah.

Acknowledgment

Saya berterimakasih kepada Tuhan Yang Maha Esa, karena hanya berkatNya lah penulis dapat menyelesaikan makalah ini.

Penulis juga berterimakasih kepada Bapak Rinaldi Munir selaku Dosen mata kuliah kriptografi atas segala ilmunya yang telah diajarkan di kelas beserta seluruh asisten yang telah membantu.

References

[1] Al-Haj. Ali, Combined DWT-DCT Digital Image Watermarking, Department of Computer Engineering, School of Electrical Engineering, Princess Sumaya University for Technology, PO Box 1928, Al-Jubeiha, 11941 Amman, Jordan, 2007.

[2] Kasiran. Zolidah, A. T. Rabi'atul, Zainol. Zarina, et al, Analysis On Digital Image Watermarking Using Dct-Dwt Techniques Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia, 2022/

[3] Munir, Rinaldi, Slide Kuliah IF4020 Kriptografi: Digital Watermarking, 2023. Diakses 20 Mei 2023

[4] J. Grierson, Photographer Admits prize-winning image was AI-generated.https://www.theguardian.com/technology/2023/apr/17/photographer-admits-prize-winning-image-was-ai-generated. Diakses pada 19 Mei 2023