

Implementasi Metode Watermarking untuk Konten Media Sosial

Andjani Kiranadewi - 13518109
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13518109@std.stei.itb.ac.id

Abstract—Media sosial sudah menjadi hal yang tidak bisa dipisahkan dari kehidupan sehari-hari. Namun, banyak terjadi pembajakan akun dan munculnya akun-akun palsu yang mengambil konten dari akun lain. Makalah ini akan mengimplementasi metode watermarking pada konten media sosial dengan memperhatikan batasan-batasan yang ada pada media sosial populer seperti Facebook, Twitter, dll.

Keywords—dwt; watermarking; wavelet transform

I. PENDAHULUAN

Media sosial sudah menjadi hal yang terintegrasi dalam hidup sehari-hari masyarakat dunia. Apabila dibandingkan dengan tahun 2017, Pengguna media sosial di Indonesia mengalami kenaikan lebih dari 50% dari 120 juta orang menjadi 193 juta orang [1]. Setiap harinya, media sosial rata-rata diakses selama 142 menit [2].

Ramainya media sosial seringkali menginspirasi pengguna untuk membuat konten. Dari hasil fotografi, lukisan digital, video, dan lain lain. Namun, dunia maya yang luas dan bersifat anonim mempersulit para pembuat konten untuk menjaga kepemilikan dari kontennya. Penanda berupa tanda tangan ataupun *watermark* yang terlihat tidak cukup karena adanya teknologi untuk memanipulasi file digital. Dengan itu, makalah ini akan membahas mengenai implementasi metode watermarking pada konten media sosial dengan memperhatikan batasan-batasan yang ada pada media sosial populer seperti Facebook dan Twitter.

II. DASAR TEORI

A. Watermarking

Sebuah konten dapat ditandai kepemilikannya menggunakan *watermark*. *Watermark* sendiri membubuhkan sebuah penanda pada konten seperti gambar dan video. Pada konten digital, *watermark* dapat bersifat dapat dilihat secara kasat mata (*visible*) atau terselubung (*invisible*).

Sebuah *watermark* memiliki beberapa karakteristik dasar [3]:

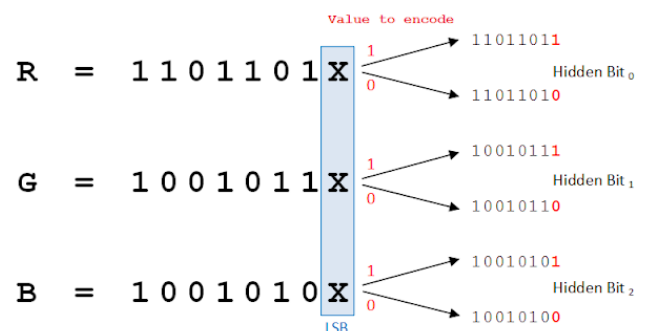
- Imperceptibility: apakah *watermark* dapat terlihat atau tidak

- Capacity: seberapa besar *watermark* yang dapat disisipkan
- Robustness: ketahanan *watermark* terhadap modifikasi pada citra

Untuk *watermark* yang bersifat tidak terlihat, terdapat dua metode untuk mengaplikasikan sebuah *watermark* [4]:

1) Metode domain spasial

Metode ini langsung memodifikasi nilai *byte* dari konten yang ingin diberi *watermark*. Umumnya, konten diubah menjadi bentuk *binary* terlebih dahulu, kemudian diubah berdasarkan *watermark* yang ingin dibubuhi. Metode yang paling umum dari jenis ini adalah metode LSB, dimana setiap *bit* dari *watermark* dimasukkan ke Least Significant Bit dari setiap *byte* pada konten.



GAMBAR I. Visualisasi Metode Least Significant Bit (LSB) (Sumber: <https://www.hacking.land/>)

Metode jenis ini dinilai kurang aman untuk digunakan sebagai *watermark*, karena rentan terhadap transformasi terhadap konten seperti kompresi, *cropping*, rotasi, dll. Apabila konten yang dibubuhi *watermark* dimodifikasi, maka *watermark* juga akan rusak.

2) Metode domain frekuensi

Pada metode ini, domain asal dari konten (spasial atau temporal) ditransformasi menjadi domain frekuensi. Metode yang merupakan jenis ini antara lain Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), dan Discrete Wavelet Transform (DWT).

B. Discrete Wavelet Transform (DWT)

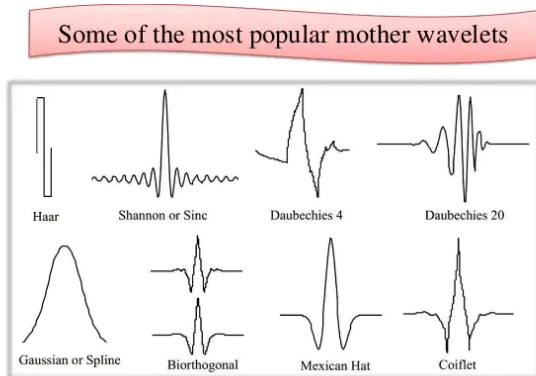
Discrete Wavelet Transform (DWT) merupakan salah satu dari teknik *watermarking* yang memakai domain frekuensi. Teknik ini mendekomposisi sebuah sinyal menjadi dua set koefisien transform, yaitu *low pass filter* yang menganalisis frekuensi rendah dan *high pass filter* yang menganalisis frekuensi tinggi. Set dengan frekuensi tinggi mengandung informasi mengenai *edge* pada citra [5].

Transformasi ini merupakan salah satu jenis dari *wavelet transform*. Sebuah *wavelet transform* dari sinyal kontinu $x(t)$ dapat direpresentasikan dalam bentuk rumus [6]:

$$T(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (1)$$

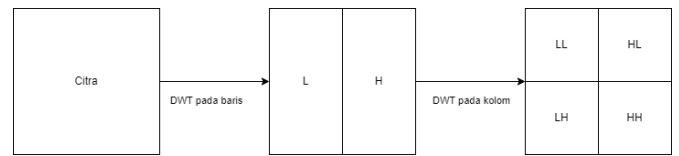
dengan $\psi^*(t)$ adalah konjugat dari *mother wavelet* $\psi(t)$, a adalah parameter skala dari *wavelet*, dan b adalah lokasi dari *wavelet*.

DWT bekerja dengan menggunakan sebuah *mother wavelet* untuk menganalisis sebuah sinyal. *Mother wavelet* merupakan sebuah fungsi ortogonal. Sesuai namanya, posisi serta skala *wavelet* bersifat diskrit, sehingga jumlah koefisien yang didapatkan dibatasi. *Wavelet* yang digunakan dapat didefinisikan sendiri, namun terdapat beberapa *wavelet* yang sering digunakan seperti Haar dan Daubechies.

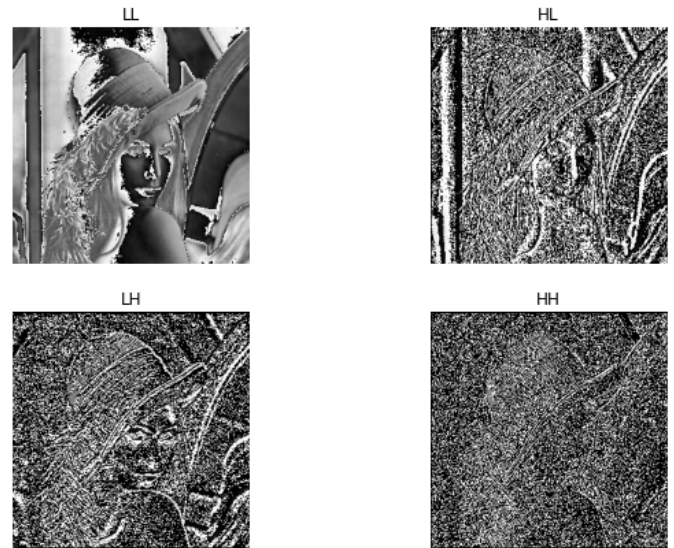


GAMBAR II. *Mother wavelet* yang sering digunakan pada *wavelet transform* (Sumber: <https://www.slideshare.net/>)

Pada citra, DWT diaplikasikan dua kali, pertama pada baris, kemudian pada kolom. Penggunaan DWT sebanyak satu level akan menghasilkan empat buah citra, yaitu LL, LH, HL, dan HH, dengan huruf pertama mengindikasikan jenis filter yang digunakan pada baris, dan huruf kedua jenis filter yang digunakan pada kolom. Citra LL adalah citra aproksimasi, sedangkan tiga citra lainnya merupakan citra detail untuk komponen horizontal, vertikal, dan diagonal. Dekomposisi sebanyak N level akan menghasilkan citra sebanyak $3N+1$. Contoh dari dekomposisi citra oleh DWT dapat dilihat pada Gambar III



GAMBAR III. Dekomposisi DWT terhadap citra



Gambar IV. Hasil dekomposisi DWT pada citra Lena

III. RANCANGAN SOLUSI DAN IMPLEMENTASI

A. Konten Citra

Penyisipan *watermark* dilakukan menggunakan metode alpha blending yang diajukan pada [7] dan [8]. Metode ini menggabungkan *cover object* dan *watermark* menggunakan *blending factor* alpha untuk mengatur persentase warna dari tiap objek.

Berikut tahap dari pembubuhan *watermark*:

- 1) Ubah ukuran *cover image* dan *watermark* sesuai dengan kebutuhan. Tinggi dan lebar citra harus dapat dibagi dengan 2^N , dengan N adalah jumlah dekomposisi yang akan dilakukan
- 2) Kedua citra dipisah setiap *channel*-nya. Karena citra yang digunakan memiliki format RGB, maka citra akan terbagi menjadi citra komponen R, G, dan B
- 3) Lakukan dekomposisi terhadap semua komponen warna dengan menggunakan DWT
- 4) Komponen LL level N dari *watermark* digabungkan terhadap komponen level N pada sub band lain pada *cover image* menggunakan teknik alpha blend. Hasil *cover image* C yang sudah dibubuhi *watermark* W dengan faktor *blending* k dihitung dengan rumus:

$$W_m(x, y) = (k * C(x, y)) + ((1 - k) * W(x, y)) \quad (2)$$

- 5) Rekonstruksi kembali setiap *channel* dengan melakukan DWT invers pada komponen dekomposisi
- 6) *Channel-channel* dari *cover image* yang sudah dibubuhi *watermark* digabungkan untuk membentuk citra RGB

Berikut adalah kode dalam bahasa Python untuk proses *watermarking*.

```
def embed(cover, wm, level=3, sub_band='LH',
wavelet='haar', k=0.95):
    channels = cover.getbands()
    if sub_band.upper() not in ('LL', 'LH', 'HL', 'HH'):
        raise ValueError

    size = cover.size
    mode = cover.mode
    n = determine_bytes(mode)

    wm = wm.resize(size)

    res = {}
    for channel in channels:
        cv_data = list(cover.getchannel(channel).getdata())
        cv_arr = np.array(cv_data, dtype='uint8')
        cv_arr = dwt_reshape(cv_arr, size[0], size[1], 1)

        wm_data = list(wm.getchannel(channel).getdata())
        wm_arr = np.array(wm_data, dtype='uint8')
        wm_arr = dwt_reshape(wm_arr, size[0], size[1], 1)

        cv_dwt = dwtn(cv_arr, level, wavelet)

        wm_dwt = dwtn(wm_arr, level, wavelet)
        wm_ll = fetch_sub_band(wm_dwt, level, 'LL')

        blend_embed(cv_dwt, wm_ll, level, sub_band, k, 1-k)

    recon = idwtm(cv_dwt, level)

    channel_recon = Image.fromarray(dwt_reshape(recon,
size[0], size[1], 1).astype(np.uint8), mode='L')
    res[channel] = channel_recon
    return Image.merge("RGB", (res['R'], res['G'], res['B']))
```

Untuk ekstraksi *watermark* dari citra dapat dilakukan dengan:

- 1) Pisahkan citra menjadi *channel-channel* warna

- 2) Lakukan dekomposisi DWT terhadap citra *watermarked* dan citra asli
- 3) Ekstrak *watermark* dari setiap *channel* menggunakan teknik ekstraksi alpha blend. *Watermark* dari citra Wm yang berasal dari pembubuhan *watermark* pada citra C dengan faktor *blending* k dapat dicari dengan rumus:

$$W(x, y) = Wm(x, y) - (k * C(x, y)) \quad (3)$$

- 4) Hasil ekstraksi dari tahap 3 digabungkan untuk membentuk citra *watermark* dengan format RGB

Berikut adalah kode dalam bahasa Python untuk proses ekstraksi *watermark*.

```
def extract(stego: Image, orig: Image, level=3,
sub_band="LH", wavelet='haar', k=0.95):
    channels = stego.getbands()

    if sub_band.upper() not in ('LL', 'LH', 'HL', 'HH'):
        raise ValueError

    size = stego.size
    mode = stego.mode
    n = determine_bytes(mode)

    res = {}
    for channel in channels:
        st_data = list(stego.getchannel(channel).getdata())
        st_arr = np.array(st_data, dtype='uint8')
        st_arr = dwt_reshape(st_arr, size[0], size[1], 1)

        or_data = list(orig.getchannel(channel).getdata())
        or_arr = np.array(or_data, dtype='uint8')
        or_arr = dwt_reshape(or_arr, size[0], size[1], 1)

        st_dwt = dwtn(st_arr, level, wavelet)
        or_dwt = dwtn(or_arr, level, wavelet)

        extracted = blend_extract(st_dwt, or_dwt, level,
sub_band, k)

        channel_recon =
Image.fromarray(dwt_reshape(extracted,
int(size[0]/(2**level)), int(size[1]/(2**level)),
1).astype(np.uint8), mode='L')
        res[channel] = channel_recon
    return Image.merge("RGB", (res['R'], res['G'], res['B']))
```

IV. EKSPERIMEN DAN ANALISIS

A. Konten citra

Untuk eksperimen, citra yang digunakan sebagai *cover image* adalah citra lena.bmp berukuran 512x512 dan city.bmp yang berukuran 500x500. *Watermark* yang digunakan adalah citra *grayscale* berukuran 32x32



Gambar V. Citra lena.bmp (kiri) dan city.bmp (kanan) yang akan digunakan sebagai *cover image*



Gambar VI. Citra letters.png

a. Metode pengukuran

Kualitas dari gambar hasil *watermarking* dapat diukur menggunakan Peak Signal to Noise Ratio (PSNR). Metrik ini merupakan rasio dari nilai maksimal dari sebuah sinyal terhadap kekuatan dari noise pada sebuah citra [9]. PSNR dapat dihitung dengan rumus:

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right) \quad (4)$$

Pada sebuah citra, nilai MAX_f adalah nilai maksimum dari sebuah komponen warna, yaitu 255. MSE merupakan akronim dari *Mean Squared Error*, dimana dihitung rata-rata dari selisih kuadrat kedua buah citra. Rumus dari MSE:

$$MSE = \frac{1}{MN} \sum_0^{m-1} \sum_0^{n-1} ||f(i,j) - g(i,j)||^2 \quad (5)$$

b. Hasil *watermarking*

Citra letters.png disisipkan pada citra Lena pada sub band HL dan nilai k sebesar 0.95. Hasil dari pembubuhan pada level dekomposisi yang berbeda dapat dilihat pada Gambar VII dan VIII.



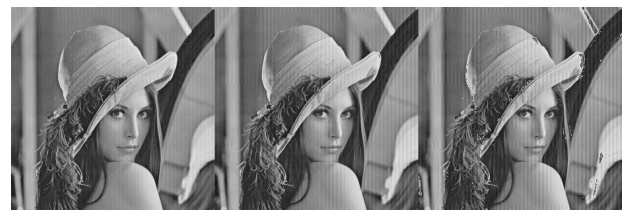
Gambar VII. Citra lena.bmp asli (kiri) dan citra lena.bmp dengan *watermark* pada sub band LH1 (kanan)



Gambar VIII. Citra lena.bmp dengan *watermark* pada sub band LH2 (kiri) dan citra lena.bmp dengan *watermark* pada sub band LH3 (kanan)

Secara sekilas, dapat terlihat bahwa semakin rendah frekuensi tempat *watermark* dibubuhi, semakin jelas perbedaannya pada citra hasil akhir. Hal ini disebabkan frekuensi rendah pada citra menyimpan lebih banyak detail dibandingkan frekuensi yang lebih tinggi.

Selain level dekomposisi, nilai k juga mempengaruhi kualitas dari gambar hasil *watermarking* dan *watermark* yang diekstrak. Gambar berikut merupakan perbedaan citra hasil *watermarking* pada sub band HL1 dengan nilai k yang berbeda.



Gambar IX. Citra lena.bmp dengan k=0.95 (kiri), k=0.9 (tengah), dan k=0.85 (kanan)



Gambar X. Hasil ekstraksi *watermark* dari lena.bmp dengan $k=0.95$ (kiri), $k=0.9$ (tengah), dan $k=0.85$ (kanan)

Pada Tabel I dapat dilihat PSNR dari hasil pemberian *watermark* terhadap citra Lena dan City dengan parameter yang berbeda.

TABEL I. PSNR DARI CITRA LENA YANG SUDAH DIBERI *WATERMARK*

Parameter		PSNR
Level	k	
1	0.95	28.213
2	0.95	28.226
3	0.95	28.244
1	0.9	22.083
1	0.85	17.719

TABEL II. PSNR DARI CITRA CITY YANG SUDAH DIBERI *WATERMARK*

Parameter		PSNR
Level	k	
1	0.95	28.241
2	0.95	28.244
3	0.95	28.294
1	0.9	20.668
1	0.85	16.742

c. Pengujian terhadap kompresi

Sebuah foto yang diunggah ke media sosial umumnya dikompresi terlebih dahulu untuk mengurangi data yang perlu di *load*. Dengan mengunggah citra Lena yang sudah diberi *watermark*, dilakukan pengujian terhadap ketahanan *watermark* terhadap kompresi.

Citra PNG, setelah diunggah ke Twitter, mengalami kompresi menjadi 90% ukurannya. Sedangkan citra yang diunggah ke Facebook dikonversi menjadi JPEG, yang mana kompresinya bersifat *lossy*.



Gambar XI. Hasil ekstraksi *watermark* setelah citra diunggah ke Twitter



Gambar XII. Hasil ekstraksi *watermark* setelah citra diunggah ke Facebook

V. KESIMPULAN DAN SARAN

Digital image watermarking dengan menggunakan metode domain frekuensi terbukti lebih tahan terhadap manipulasi citra. *Watermark* yang disisipkan ke dalam citra tetap dapat dikenali meskipun citra sudah dikompresi atau diberi *noise*.

Dari ketiga karakteristik, *watermark* yang dihasilkan oleh DWT dengan *alpha blending* bersifat robust, memiliki kapasitas sesuai dengan level dekomposisi, namun masih dapat terlihat secara kasat mata. Kedepannya, solusi perlu dikembangkan lebih jauh agar kualitas citra dapat dijaga setelah dilakukan *watermarking*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih terhadap Tuhan Yang Maha Esa, karena dengan segala karunia-Nya, penulis dapat menyelesaikan makalah ini.

Selain itu, saya juga mengucapkan terima kasih kepada Bapak Dr. Rinaldi Munir selaku dosen Mata Kuliah IF 4020 Kriptografi. Berkat bimbingannya, saya memiliki ilmu pengetahuan mengenai ilmu kriptografi.

REFERENSI

- [1] Social media users in Indonesia 2025 | Statista (n.d). <https://www.statista.com/forecasts/1144743/social-media-users-in-indonesia>. Diakses pada tanggal 20 Desember 2021
- [2] Armstrong, Martin. Is Peak Social Media Already Behind Us? <https://www.statista.com/chart/26272/global-average-daily-time-spent-on-social-media-per-internet-user/>. Diakses pada tanggal 20 Desember 2021
- [3] Hai Tao, et al. "Robust Image Watermarking Theories and Techniques: A Review". Journal of Applied Research and Technology 12. 1(2014): 122-138.
- [4] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi: Steganografi (Bagian 2). Diakses pada tanggal 20 Desember 2021
- [5] N. Kashyap and G. R. Sinha, "Image Watermarking Using 3-Level Discrete Wavelet Transform (DWT) Image denoising View project Breast Cancer View project Image Watermarking Using 3-Level Discrete Wavelet Transform (DWT)," Article in International Journal of Modern Education and Computer Science, vol. 3, pp. 50–56, 2012, doi: 10.5815/ijmecs.2012.03.07.

- [6] Addison, P.S.. (2005). Wavelet transforms and the ECG: A review. Physiological measurement. 26. R155-99. 10.1088/0967-3334/26/5/R01.
- [7] N. Dey, A. Professor, A. Bardhan, R. B. Tech, and B. E. Student, "A Novel Approach of Color Image Hiding using RGB Color planes and DWT Sayantan Dey," International Journal of Computer Applications, vol. 36, no. 5, pp. 975–8887, 2011.
- [8] Akhil Pratap Singh, A. WAVELET BASED WATERMARKING ON DIGITAL IMAGE. Indian Journal of Computer Science and Engineering, 1, 86-91.
- [9] Peak Signal-to-Noise Ratio as an Image Quality Metric - NI (n.d.). <https://www.ni.com/en-id/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>. Diakses pada tanggal 20 Desember 2021

Bandung, 20 Desember 2021



Andjani Kiranadewi - 13518109

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.