

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Ada beberapa penelitian sebelumnya yang telah dilakukan oleh beberapa peneliti yaitu sebagai berikut:

1. Pada makalah ini telah dirancang algoritma implementasi steganografi untuk keamanan pengiriman citra digital menggunakan metoda DCT (*Discrete Cosine Transform*). Steganografi yang dirancang terdiri dari dua bagian. Yaitu disisi pengirim dilakukan penyisipan data digital berupa citra rahasia .bmp *grayscale* ke dalam citra media .bmp RGB., sedangkan disisi penerima dilakukan ekstraksi citra rahasia yang telah disisipkan ke dalam citra media. Pengiriman Citra digital dengan proses *steganografi* terbukti aman karena citra rahasia yang disisipkan ke cover tidak dapat dilihat secara persepsional. Resolusi gambar akan mempengaruhi penyisipan, waktu pengiriman, dan waktu ekstraksi. Waktu pengiriman paling lama terjadi pada pengiriman citra 48x48 piksel sebesar 13.668 detik Nilai PSNR untuk ukuran citra 32x32 adalah 44.292 dB, ukuran 40x40 sebesar 43.643 dB dan ukuran 48x48 sebesar 43.401 dB. Sedangkan nilai MSE ukuran citra 32x32 adalah 2.419, ukuran 40x40

sebesar 2.810 dan ukuran 48x48 sebesar 2.970. Dari hasil terlihat bahwa kualitas hasil proses ekstraksi *steganografi* dilihat dari tingkat kemiripan citra asli dengan citra hasil ekstraksi masih dapat diterima dengan kondisi baik (Hamdani & Samosir, 2018).

2. Penerapan metode DCT (*discrete cosine transform*) pada aplikasi penyembunyian pesan *teks* berbasis matlab. Dari penelitian ini aplikasi *Steganografi* berbasis desktop yang dapat digunakan sebagai media untuk menyisipkan pesan kedalam citra dengan menerapkan metode DCT (*Discrete Cosine Transform*) berjalan dengan baik. Pada pengujian aplikasi ini dengan file pesan berupa file dengan ekstensi *.txt. isi pesan 100 karakter, 250 karakter dan lebih dari 500 karakter berhasil dilakukan dengan baik. File citra input : dengan ekstensi *.bmp lebih baik digunakan untuk media penyisipan pesan karena ukuran file input dengan output sama besar. Metode DCT (*Discrete Cosine Transform*) dapat menyisipkan pesan kedalam gambar baik itu berupa huruf, angka dan *symbol* (Ardiansyah et al., 2017).
3. Dari uji coba yang mereka lakukan yaitu *mewatermarking* citra digital menggunakan metode *discrete cosine transform* dapat disimpulkan bahwa semakin besar nilai K yang merupakan bilangan yang menjadikan blok-blok koefisien DCT yang telah ditukar antara dua blok yang telah ditentukan sebagai area penanaman memiliki selisih tertentu, akan menghasilkan kualitas citra *terwatermark* yang semakin menurun sehingga secara kasat mata

tampak adanya perubahan pada *cover host*. Metode yang diusulkan juga menghasilkan citra ter*watermark* yang tahan terhadap serangan (Fanani & Ulinuha, 2016).

4. Jurnalnya ini berfokus pada tema desain dan implementasi *watermark* digital citra berbasis DCT. Meningkatkan algoritma *watermarking* gambar digital yang didasarkan pada transformasi DCT dan pengacakan *Arnold*, Ini memastikan keamanan *watermarking* oleh *Arnold* mengacak informasi *watermark* asli. Sistem *watermarking* dapat membuat citra ber-*watermark* terbaik pada bit ke-16, 17 dan 18 frekuensi tengah koefisien DCT. Sistem *watermarking* dengan nilai PSNR rata-rata 63 dB pada penyisipan *watermark* bit ke-16, 17 dan;18 frekuensi tengah koefisien DCT. Sistem *watermarking* yang dibangun dapat melakukan ekstraksi citra ber-*watermark* terbaik dengan rata- rata nilai PSNR 43 dB untuk ekstraksi citra dan rata-rata nilai PSNR 22 dB untuk ekstraksi *watermark* pada penyisipan *watermark* bit ke-16, 17 dan, 18 frekuensi tengah koefisien DCT. Sistem yang dibangun tidak *reversibility* karena pada saat proses ekstraksi, *watermark* rusak sebelum dilakukan deteksi. Sistem dapat melakukan perbaikan dengan hasil yang buruk karena citra ciri mengalami kerusakan di seluruh bagian citra (Azzahra & Triantoro, 2015).
5. Penelitiannya ini menggunakan algoritma *Huffman* untuk mengkodekan koefisien hasil proses DCT yang akan mengeliminasi nilai bernilai nol menghilangkan kelebihan dari keluaran kuantiser secara zig-zag dan akhirnya

akan memperoleh image yang telah direkonstruksi / kompres. Pengujian dilakukan dengan 2 tahap yaitu pengujian *white box* dilakukan dengan sintak dan angka sedangkan pengujian *black box* dilakukan dengan menguji apakah setiap fungsi didalam program dapat dijalankan. Menggunakan program java dibuat NetBeans IDE. Dengan teknik *Lossy Compression* kompresi citra yang dihasilkan sedikit mengurangi warna namun tampak tidak terlihat perbedaannya dengan citra asli sebelum dikompresi. Hasil kompresi tergantung pada pemilihan kualitas kompresi yang diinginkan. Jika kita memilih kompresi dengan kualitas standar, maka citra hasil kompresi dengan citra yang asli tidak akan terlihat perbedaannya namun pengurangan ukuran *bytes* tidak terlalu drastis (Krasnala et al., 2017).

6. Pada jurnal ini proses penyisipan citra RGB akan dikonversikan menjadi citra YCbCr kemudian diambil komponen *luminance* (Y) saja untuk proses transformasi DCT. Kemudian komponen Y menjadi sub blok berukuran 8 x 8 piksel, lalu komponen Y digabung kembali dengan komponen Cb dan Cr sehingga membentuk citra YCbCr *terwatermarking*. Proses terakhir konversikan kembali citra YCbCr *terwatermark* menjadi citra RGB sehingga menjadi citra RGB *terwatermark*, bahasa pemrograman Visual C# .NET. Untuk mendapatkan hasil ekstraksi yang baik, perbandingan ukuran citra *watermark* dengan citra asli adalah 1:32 piksel. Pengujian kompresi JPEG pada citra *terwatermark* menghasilkan citra *terwatermark* dapat tahan terhadap kompresi hingga 50% dan menghasilkan nilai PSNR yang cukup

tinggi, yaitu 30 dB. Pengujian berupa *screen capture* menghasilkan citra *watermark* dengan kualitas yang rendah dengan nilai PSNR rata-rata 25 dB. Pada pengujian citra berupa *cropping*, citra *watermark* akan hilang pada bagian yang terkena *cropping*. Pengujian berupa perubahan warna dan penambahan teks menghasilkan hasil ekstraksi citra *watermark* dengan kualitas yang baik secara *imperceptibility*, namun citra *watermark* memiliki nilai PSNR yang rendah karena bernilai di bawah 30 dB, namun pada penggantian *background* dan penambahan filter berupa *emboss* dan penambahan filter blur menghasilkan ekstraksi citra *watermark* yang buruk, karena nilai PSNR yang sangat rendah atau bernilai di bawah 15 Db (Agustina & Asmara, 2004).

7. Dalam jurnal *A DCT Domain Image Watermarking Method Based on Matlab* berfokus pada tema desain dan implementasi *watermark* digital citra berbasis DCT. Meningkatkan algoritma *watermarking* gambar digital yang didasarkan pada transformasi DCT dan pengacakan *Arnold*. Ini memastikan keamanan *watermarking* oleh *Arnold* mengacak informasi *watermark* asli. Hasil percobaan menunjukkan bahwa ia menyembunyikan informasi citra abu-abu, dan melakukan simulasi eksperimental pada beberapa serangan citra umum. Melakukan berbagai percobaan untuk menyerang algoritma tersebut, disimpulkan bahwa algoritma tersebut dapat diterapkan dengan baik untuk perlindungan hak cipta dan otentikasi. Hasil percobaan membuktikan bahwa algoritma *watermarking* digital yang kami gunakan memiliki ketahanan yang

baik, *transparansi*, algoritma dengan kompleksitas rendah. Singkatnya, penelitian tentang teknik *watermarking* digital dikembangkan dalam beberapa tahun terakhir dan lebih aktif. Penelitian di bidang ini telah mencapai beberapa hasil, tetapi beban kerjanya tidak cukup. Diperlukan lebih banyak penelitian untuk mendapatkan kemajuan yang lebih besar di bidang ini (He-Jing, 2017).

8. Dalam jurnal *Discrete Cosine Transform Residual Feature based Filtering Forgery and Splicing Detection in JPEG Images*, mereka telah mengusulkan pendekatan berbasis fitur DCTR untuk memfilter deteksi dan klasifikasi pemalsuan serta deteksi penyambungan berbasis kompresi ulang dengan pengklasifikasi SVM. Efektivitas fitur DCTR secara teoritis dibenarkan untuk deteksi penyaringan *linier* dan *non-linier* (median) dan masalah klasifikasi, bersama dengan deteksi penyambungan berbasis kompresi ulang. Secara *eksperimental*, kemanjuran teknik yang diusulkan dan keunggulannya atas teknik mutakhir juga diverifikasi untuk beberapa jenis penyaringan *linier*, pemfilteran median, dan deteksi penyambungan untuk kumpulan data patokan standar yang berbeda. Upaya penelitian kami di masa mendatang akan diarahkan untuk mengeksplorasi penerapan teknik yang diusulkan untuk format gambar lain dan juga teknik deteksi pemalsuan terpadu, efektif untuk beberapa pemalsuan gambar (Roy et al., 2018).

B. Landasan Teori

1. Pemrograman Matlab

Matlab adalah sebuah bahasa dengan (*high-performance*) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk dipakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Matlab merupakan akronim dari kata *Matrix Laboratory*. Versi pertama Matlab ditulis pada tahun 1970. Saat itu, Matlab digunakan untuk pelatihan dalam teorimatrik, aljabar *linier* dan analisis *numerik*. Fungsi-fungsi Matlab ini digunakan untuk menyelesaikan masalah bagian khusus, yang disebut *tool boxes*. *Tool boxes* dapat digunakan untuk bidang pengolahan sinyal, *system* pengaturan, *fuzzy logic*, *numeral network*, optimasi, pengolahan citra, dan simulasi yang lain(Ardiansyah et al., 2017).

2. Informasi Multimedia

Multimedia adalah penggunaan komputer untuk menampilkan informasi yang merupakan gabungan dari *teks*, grafik, *audio* dan *video*. Selain itu, konten digital multimedia dapat diakses dan didistribusikan melalui jaringan. Akibatnya, proses duplikasi menjadi mudah dan tidak mengurangi isinya dimana hasil duplikasi menjadi identik dengan yang asli dan dapat diproduksi dengan mudah(Sari, 2018).

3. Definisi Citra

Citra digital dapat dinyatakan sebagai fungsi kontinu dari intensitas cahaya dari dua dimensi dalam fungsi $f(x,y)$, dimana (x,y) menyatakan koordinat spasial pada bidang dua dimensi dan nilai dari f pada titik (x,y) menyatakan intensitas cahaya (*brightness*) pada titik (x,y) tersebut. Agar dapat diolah dengan komputer digital, suatu citra harus direpresentasikan dengan nilai-nilai *diskrit* (citra digital). Pada umumnya citra digital berbentuk empat persegi panjang dengan berukuran dua dimensi (panjang x lebar). Citra digital yang berukuran $N \times M$ dinyatakan dengan matriks yang berukuran N baris dan M kolom (Hamdani & Samosir, 2018).

4. Grayscale

Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu (Sen, 2018). Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra keabuan hanya perlu menyatakan nilai intensitas untuk tiap *pixel* sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap *pixel*nya (Wijaya, 2019).

Konversi dari citra RGB ke *grayscale* pada persamaan (1) :

$$I_y = 0,2989F_r + 0,5870F_g + 0,1140F_b \quad (1)$$

F_r = intensitas R

F_g = intensitas G

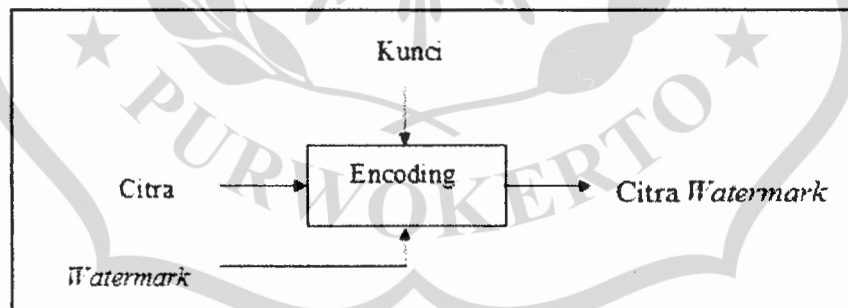
F_b = intensitas B

I_y = intensitas abu-abu yang setara gambar level RGB

5. Watermarking

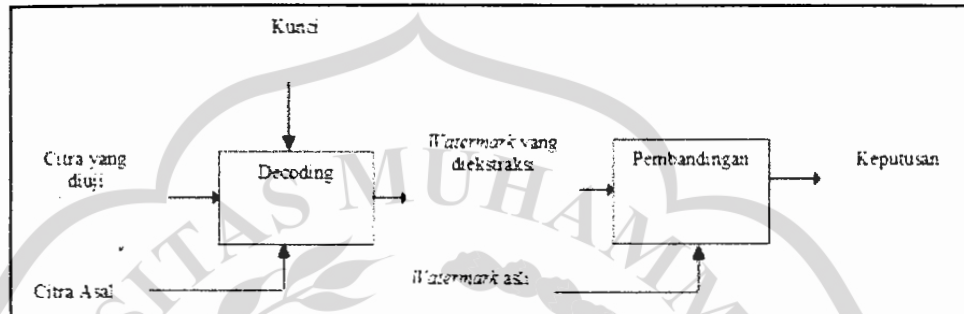
Watermarking merupakan suatu bentuk dari *steganography*, yaitu ilmu yang mempelajari bagaimana menyembunyikan suatu data pada data yang lain. *Watermarking* (tanda air) ini agak berbeda dengan tanda air pada uang kertas. Tanda air pada uang kertas masih terlihat oleh indera manusia (dalam posisi kertas tertentu), tetapi *watermarking* pada media digital tak akan dirasakan kehadirannya oleh manusia tanpa alat bantu mesin pengolah digital seperti computer (Irfan, 2013).

Pada Gambar 2. 1. Berikut ini adalah diagram yang digunakan untuk menggambarkan proses penyisipan *watermark* pada dokumen citra digital :



Gambar 2. 1. Proses *Watermark* pada dokumen Citra Digital

Sedangkan Gambar 2.2. Berikut ini merupakan gambar diagram yang di gunakan untuk menggambarkan proses verifikasi *watermark* pada citra digital:



Gambar 2. 2. Proses verifikasi *watermark* pada dokumen Citra Digital.

Didalam *watermarking* terdapat berbagai macam teknik yang dapat digunakan. Salah satunya adalah sebagai berikut:

a. *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform (DCT) adalah sebuah teknik untuk mengubah sinyal kedalam komponen frekuensi dasar. Kelebihan kompresi data menggunakan *Discrete Cosine Transform* adalah:

- 1) DCT menghitung kuantitas bit-bit data gambar dimana pesan tersebut
- 2) Disembunyikan didalamnya. Walaupun *image* di kompresi dengan *lossy compression* tidak akan menimbulkan kecurigaan karena metode ini terjadi di domain frekuensi didalam *image*, bukan pada domain spasial sehingga tidak akan ada perubahan dilihat pada *image*.
- 3) Kokoh terhadap manipulasi pada *stego-object*.

Sedangkan kekurangan kompresi data menggunakan *Discrete Cosine Transform* adalah:

- 1) Tidak tahan terhadap perubahan suatu objek dikarenakan pesan mudah dihapus karena lokasi penyisipan data dan pembuatan data dengan metode DCT diketahui
- 2) *Implementasi* algoritma yang panjang dan membutuhkan banyak perhitungan.

Penggunaan DCT pada pengolahan citra dilakukan dengan membagi citra kedalam sub blok berukuran standar 8x8 piksel. Dimana sub blok 8x8 piksel tersebut akan menghasilkan 64 koefisien yang terdiri dari 1 koefisien DC (*zero frequencies*) yang terletak pada pojok kiri atas dan 63 koefisien AC yang terdiri dari 3 frekuensi, yaitu frekuensi rendah, frekuensi menengah, dan frekuensi tinggi (Agustina & Asmara, 2004).

DCT dapat digunakan untuk mengubah informasi spasial kebentuk frekuensi atau *spektral*. Terdapat fungsi *invers* DCT (IDCT) yang dapat membalikkan representasi *spectral* kebentuk spasial mula-mula. Rumus DCT dua dimensi dapat ditunjukkan pada persamaan (2) dan rumus untuk *Invers* DCT ditunjukkan pada persamaan (3) sebagai berikut :

i = posisi baris untuk koefisien DCT y = posisi kolom untuk piksel
 j = posisi kolom untuk koefisien DCT x = posisi baris untuk piksel
 C = citra komponen nilai piksel aslinya N = banyaknya baris/kolom

π = Perbandingan Keliling lingkaran dengan diameternya

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

$$C(i)C(j) = \begin{cases} \frac{1}{\sqrt{N}} & \text{untuk } i, j = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & \text{untuk } i, j \neq 0 \end{cases} \quad (2)$$

DCT dikenakan pada sebuah matriks bujur sangkar $N \times N$ dari nilai-nilai piksel, dan akan menghasilkan sebuah matrik bujur sangkar $N \times N$ dari koefisien frekuensi.

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

$$C(i)C(j) = \begin{cases} \frac{1}{\sqrt{N}} & \text{untuk } i, j = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & \text{untuk } i, j \neq 0 \end{cases} \quad (3)$$

Transformasi ini berbasiskan *vektor*. DCT membagi citra ke dalam 8×8 blok citra. Setiap blok akan dihitung koefisiennya masing-masing. DCT 2 dimensi menghasilkan citra hasil transformasi ke dalam matriks 2 dimensi.

b. Proses Perbandingan *Watermark*

PSNR digunakan untuk menentukan kualitas citra *watermarked* setelah disisipi *watermark*. Citra *watermarked* dibandingkan dengan citra asli (citra *host*) untuk menentukan kualitas citra *watermarked*. Semakin besar nilai PSNR berarti penyisipan pesan (*watermark*) kedalam citra asli tidak menyebabkan penurunan kualitas citra *watermarked*. Sebaliknya jika nilai

PSNR semakin kecil maka pada citra *watermarked* akan terjadi penurunan kualitas citra. Nilai PSNR biasanya mempunyai rentang nilai antara 20 dB sampai dengan 60 dB (Firdausy et al., 2006).

Secara matematis dapat dirumuskan pada persamaan (4):

M = Panjang citra dalam *pixel* yang telah di*watermarking*

N = Lebar citra dalam *pixel* yang telah di*watermarking*

I_1 = Nilai citra hasil perhitungan *inverse*

I_2 = Nilai citra *grayscale*

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (4)$$

Sementara nilai PSNR dihitung dari kuadrat nilai maksimum sinyal dibagi dengan MSE. Apabila diinginkan PSNR dalam *desibel*, maka nilai PSNR akan menjadi persamaan berikut (5):

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \quad (5)$$

Semakin rendah nilai MSE maka akan semakin baik, dan semakin besar nilai PSNR maka semakin Baik kualitas citra *watermark*.