PERBANDINGAN BERBAGAI METODE STEGANOGRAFI PADA CITRA DIGITAL

Ratih Kartika Dewi [1], Rinaldi Munir [2]

[1,2] Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
[1]33222014@std.stei.itb.ac.id, [2] rinaldi@informatika.org

Abstrak

Topik keamanan informasi selalu menarik untuk dibahas. Steganografi merupakan salah satu teknik yang digunakan untuk meningkatkan keamanan informasi yang banyak dikembangkan saat ini serta memiliki tujuan menyembunyikan keberadaan pesan untuk menghindari kecurigaan dari pihak lain. Penyisipan pesan ke dalam media pembawa tidak membuat kualitas media pembawa itu berubah dan media yang telah disisipi pesan tidak dapat dibedakan secara kasat mata dengan media aslinya. Media digital yang banyak digunakan dalam steganografi saat ini adalah citra. Oleh karena itu, penelitian ini menyediakan perbandingan metode steganografi pada citra digital, sehingga diharapkan dapat memberikan gambaran mengenai karakteristik citra setelah disisipi pesan menggunakan berbagai algoritma steganografi. Algoritma steganografi yang dibahas dalam penelitian ini meliputi domain spasial (HuGO, WOW, S-Uniward) dan domain transform (J-Uniward). Berdasarkan pengamatan, karakteristik citra yang disisipi dengan HUGO melakukan embedding pada area berkontur atau bertekstur, WOW dan S-Uniward juga melakukan embedding pada area berkontur namun intensitas pada S-Uniward lebih renggang daripada HuGO, sedangkan WOW sedikit menyisipkan pesan pada area dengan tekstur halus. Berbeda dengan domain spasial, karakteristik umum penyisipan pesan dalam domain transform adalah sulit dideteksi secara visual sehingga tidak ada perbedaan mencolok antara citra cover dan stego yang dapat dilihat secara visual, namun beberapa perbedaan yang dapat diamati dalam J-uniward saat memvisualisasikan perubahan nilai koefisien DCT yang berubah akibat adanya penyisipan, yaitu terlihat penyisipan pesan terjadi di area dengan tekstur kasar dan adanya perubahan dalam domain spasial yang terjadi akibat DCT embedding.

Kata kunci : steganografi, citra digital, domain spasial, dan domain transform.

1. Pendahuluan

Masalah keamanan informasi menjadi topik yang selalu menarik untuk dibahas. Keamanan informasi meliputi pencegahan dari pengaksesan informasi oleh pihak yang tidak berwenang, melindungi kerahasiaan informasi yang bersifat privat, pencegahan dari usaha untuk mengubah informasi, dan lain sebagainya (Munir, 2019). Berbicara tentang keamanan informasi tidak bisa lepas dari steganografi dan kriptografi. Steganografi merupakan salah satu teknik yang digunakan untuk meningkatkan keamanan informasi yang banyak dikembangkan saat ini serta memiliki tujuan menyembunyikan keberadaan pesan untuk menghindari kecurigaan dari pihak lain. Penyisipan pesan ke dalam media pembawa tidak membuat kualitas media pembawa itu berubah dan media yang telah disisipi pesan tidak dapat dibedakan secara kasat mata dengan media aslinya. Keunggulan steganografi tersebut merupakan pembeda dari kriptografi. Kriptografi menghasilkan cipherteks yang dapat menimbulkan kecurigaan bagi pengamat, sedangkan steganografi menghasilkan pesan yang tidak dapat dideteksi keberadaannya sehingga tidak menimbulkan kecurigaan. Steganografi dapat diimplementasikan pada berbagai media seperti gambar, audio, video, dan teks.

Kelebihan steganografi dibandingkan kriptografi adalah pesan rahasia yang disembunyikan tidak menarik perhatian lawan. Steganografi juga banyak digunakan karena beberapa negara melarang adanya kriptografi. Prancis, Cina dan Iran secara signifikan membatasi penggunaan kriptografi. Beberapa negara juga memiliki undang-undang yang mengatur penggunaan kriptografi, seperti di Belarus, Kazakhstan, Mongolia, Pakistan, Singapura, Tunisia, dan Vietnam (Shearer & Gutmann, 1996). Meskipun steganografi dapat menjadi cara yang aman untuk komunikasi dengan pemerintah atau bisnis, namun ini akan menimbulkan konsekuensi serius jika digunakan oleh teroris atau penjahat.

Selama ini orang beranggapan steganografi hanya sekedar menyisipkan pesan rahasia, namun steganografi ternyata bila ditelaah lebih jauh, dapat digunakan untuk menghack komputer. Selama ini hacker melakukan hacking dengan mencari celah masuk ke dalam jaringan komputer target, atau menyusupkan kode program melalui attachment file dalam format PDF. Tetapi, ada cara lain yang tidak menimbulkan kecurigaan, yaitu melalui pemasukan kode program pada gambar. Stegosploit

menggunakan teknik steganografi untuk menyembunyikan malicious code (exploit) di dalam sebuah gambar (format JPG atau PNG). Malicious code merupakan kode program jahat yang ditulis dalam bahasa JavaScript. Kode program tersebut berisi beberapa perintah untuk membaca dan mengirimkan data dari komputer target ke penyerang. Jadi, dalam hal ini pesan rahasia yang akan disembunyikan berupa teks kode program yang kemudian disisipkan pada pixel-pixel gambar. Gambar yang sudah disisipi malicious code ini tidak menimbulkan kecurigaan bagi penerimanya.

Setelah sebuah gambar disisipi malicious code, cara penyebarannya bisa menggunakan dua cara. Pertama, gambar (stego image) diletakkan pada sebuah laman web dengan perintah kepada pembaca untuk mengklik gambar tersebut (mungkin agar tampilan dalam ukuran yang lebih besar). Cara kedua, komputer target dikirimi surel (email) yang di dalamnya ada tautan (link) ke laman web yang berisi gambar tersebut. Ketika pengguna meng-klik gambar tersebut kemudian tampil pada laman browser, maka malicious code di dalamnya diekstraksi (decoded) dengan elemen kanvas HTML 5 yang berisi script untuk melakukan rendering citra digital.

Penyisipan pesan ke dalam media pembawa tidak membuat kualitas media pembawa itu berubah dan media yang telah disisipi pesan tidak dapat dibedakan secara kasat mata dengan media aslinya. Berdasarkan ranah operasinya, steganografi dapat diklasifikasikan menjadi dua kategori utama, yaitu domain spasial dan domain transform (Munir, 2019). Teknik steganografi domain spasial memodifikasi langsung nilai byte dari cover-object, sedangkan domain transform memodifikasi hasil transformasi sinyal dalam ranah transform (hasil transformasi dari ranah spasial ke ranah lain (misalnya ranah frekuensi).

Contoh algoritma steganografi dalam ranah domain spasial antara lain Least Significant Bit (Kodar, 2017; Singh & Attri, 2015; Veena & Arivazhagan, 2018), Highly Undetectable Steganography (Filler & Fridrich, 2010), Wavelet Obtained Weights (Holub & Fridrich, 2012), High-Pass Low-pass (Li et al., 2014), Spatial-Universal Wavelet Relative Distortion (S- (Holub et al., 2014), dan Minimizing the Performance of Optimal Detector (Sedighi et al., 2016). Sedangkan contoh algoritma steganografi dalam ranah domain transform antara lain Uniform Embedding Distortion (Guo et al., 2014), Uniform Embedding Revised Distortion (Guo et al., 2015), JPEG-Universal Wavelet Relative Distortion (Holub et al., 2014).

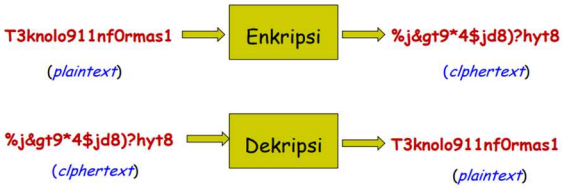
Media digital yang banyak digunakan dalam steganografi saat ini adalah citra (Fridrich, 2012). Oleh karena itu, penelitian ini menyediakan perbandingan metode steganografi pada citra digital, sehingga diharapkan dapat memberikan gambaran mengenai karakteristik citra setelah disisipi pesan menggunakan berbagai algoritma steganografi. Penelitian ini memiliki struktur sebagai berikut: pada bagian I dipaparkan mengenai latar belakang, kemudian bagian II berisi teori dasar steganografi serta bagian III berisi penjelasan mengenai karakteristik citra setelah disisipi pesan menggunakan algoritma steganografi dalam ranah domain spasial dan domain transform. Pada bagian akhir juga akan dibahas mengenai tantangan dan peluang penelitian lanjutan.

2. Dasar Teori Steganografi

Pada bagian kedua ini akan dibahas mengenai teori dasar steganografi yang meliputi pengertian steganografi dan ranah steganografi, yang meliputi domain spasial dan domain transform.

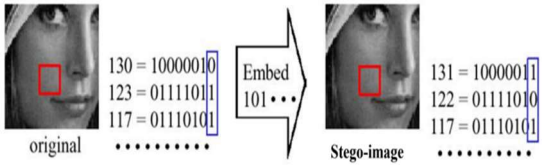
2.1 Steganografi

Apabila kita ingin menyimpan pesan rahasia supaya tidak diketahui orang lain, maka ada dua cara untuk melakukannya. Pertama dengan mengenkripsinya (merubah bentuk dari plain text ke cipher text) dan kedua adalah dengan menyembunyikannya dalam sebuah media. Cara pertama disebut dengan kriptografi seperti pada Gambar 1, sedangkan cara kedua disebut dengan steganografi. Kriptografi adalah cara melindungi isi dari pesan rahasia, sedangkan steganografi berkaitan dengan menyembunyikan keberadaan pesan rahasia yang sedang dikirim (Munir, 2019). Jadi, steganografi didefinisikan sebagai seni dan ilmu menyembunyikan atau menyematkan data rahasia melalui berbagai wadah multimedia seperti video, audio, dan gambar digital.



Gambar 1 Kriptografi
Sumber: (Munir, 2019)

Contoh penyembunyian pesan rahasia dengan Steganografi Least Significant Bit seperti diilustrasikan pada Gambar 2.



Gambar 2 Steganografi
Sumber: (Munir, 2019)

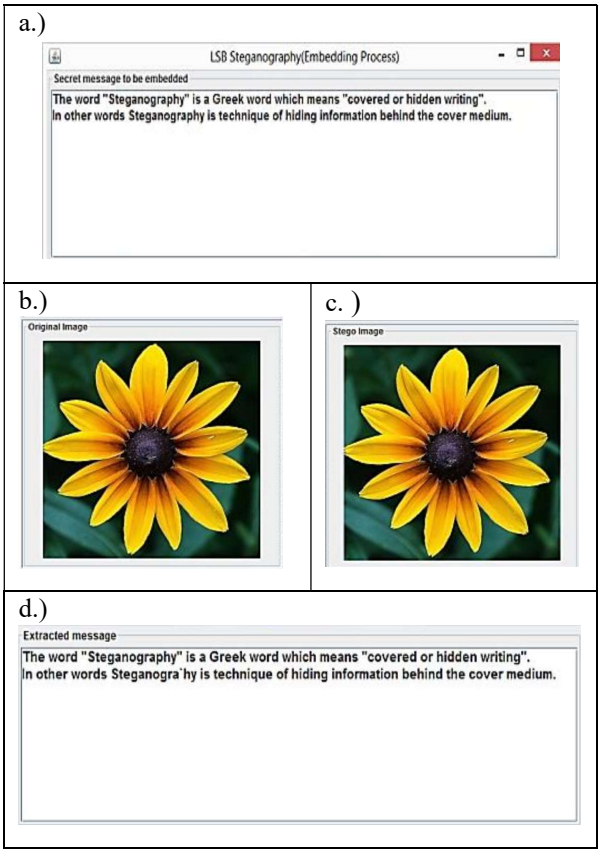
Kita akan menyembunyikan pesan rahasia (101) pada gambar Lena, dengan memilih sebuah piksel bertanda kotak merah dengan nilai Red 130, Green

123 dan Blue 117, kemudian nilai pada masing-masing channel warna diubah menjadi nilai biner, misalnya 130 menjadi 10000010. Least Significant Bit (LSB) adalah nilai paling akhir (dalam hal ini nilai 0 pada digit terakhir) yang apabila nilainya kita ubah, maka tidak akan terjadi perubahan secara visual dalam gambar. Hal ini karena mata manusia tidak begitu sensitif terhadap perubahan warna (chrominance) yang sangat kecil, namun sensitif terhadap brightness (luminance), kekurangan inilah yang dimanfaatkan untuk kompresi citra digital sekaligus juga dimanfaatkan dalam steganografi dengan metode LSB. Selanjutnya, kita juga mengganti bit pada channel warna Green dan Blue, jadi yang mulanya bernilai 011 kita ganti dengan pesan rahasia (101) seperti pada Gambar 2.

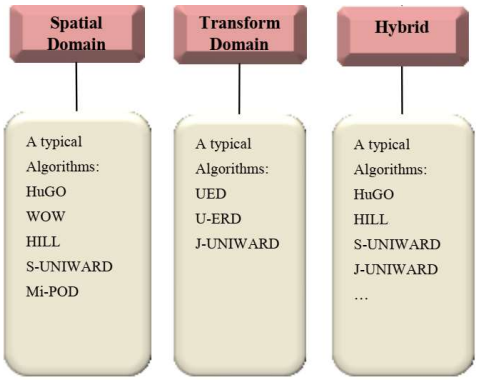
Gambar 3 adalah ilustrasi yang memberikan penjelasan tentang steganografi pada citra digital. Pembahasan diawali dengan penyembunyian pesan rahasia atau *secret message* (M) ke dalam sebuah media kosong (tanpa pesan rahasia), yang dalam steganografi disebut cover object (C), cover object adalah citra asli (*original image*). Apabila cover object disisipi pesan rahasia dengan kunci stego (K), maka ini disebut stego object (S) atau stego image. Jadi, stego object (S) berisi pesan rahasia (M) yang disisipkan dalam cover object (C). Proses ini disebut embedding, yang merupakan rangkaian inti dalam steganografi. Pesan rahasia yang ada dalam stego object dapat diekstraksi, hal ini adalah bagian dari steganalisis.

Cara melakukan embedding terhadap dataset yang tersedia di internet adalah dengan menyiapkan data citra, untuk algoritma WOW misalnya, dalam format .pgm karena domainnya spasial. Kemudian melakukan embedding dengan payload tertentu lewat program matlab yang tersedia pada internet. Penelitian (Płachta et al., 2022) menggunakan dataset BOSSbase (Bas et al., 2011) berupa 10.000 citra tanpa embedding dengan format PGM. Format PGM adalah format yang dipakai untuk domain spasial, sedangkan untuk domain transform, format citra yang biasa dipakai adalah JPEG. BOSSbase memiliki format grayscale. Citra kemudian dilakukan embedding random data dengan payload 0,4 dan 0,1 bpnzac (bit per non zero AC DC coefficient) menggunakan algoritma steganografi tertentu. Penyisipan pesan (embedding) juga dapat dilakukan dengan bantuan tools steganografi (JpegHide dan StegHide) seperti pada (Guttikonda & Sridevi, 2019). Berdasarkan ranah operasinya, steganografi dapat diklasifikasikan menjadi dua kategori utama, yaitu domain spasial dan domain transform seperti Gambar 4.

Pembahasan mengenai ranah steganografi diperlukan dalam steganalisis untuk mengetahui teori mengenai algoritma steganografi seperti pada Gambar 4 yang menunjukkan beberapa algoritma steganografi dalam ranah domain spasial, seperti HUGO, WOW, HILL, S-Uniward, Mi-POD serta algoritma steganografi dalam ranah domain transform seperti UED, UERD dan J-Uniward.



Gambar 3 Proses embedding pesan rahasia (a) ke dalam cover image (b) sehingga menjadi stego image (c), ekstraksi pada stego image menghasilkan pesan yang telah diekstraksi (d)
Sumber: (Singh & Attri, 2015)



Gambar 4 **Algoritma Steganografi berdasarkan Ranah Steganografi**
Sumber: (Bashir & Selwal, 2021)

2.2 Steganografi Domain Spasial

Teknik steganografi domain spasial juga dikenal sebagai teknik substitusi, yaitu sekelompok teknik yang relatif sederhana dengan membuat penyisipan pesan rahasia di bagian-bagian citra cover yang mana perubahannya cenderung sedikit jika dilihat dengan mata manusia. Teknik steganografi domain spasial (waktu) memodifikasi langsung nilai byte

dari cover-object (nilai byte dapat merepresentasikan intensitas, warna pixel atau amplitudo). Salah satu cara untuk melakukan steganografi dalam domain spasial adalah dengan menyembunyikan informasi dalam Least Significant Bit (LSB) dari data citra. Metode penyisipan ini pada dasarnya didasarkan pada fakta bahwa bit paling tidak signifikan dalam suatu citra dapat dianggap sebagai noise acak, dan akibatnya, bit tersebut menjadi tidak responsif terhadap perubahan apapun pada gambar. Dengan kata lain, apabila kita mengganti LSB dengan pesan rahasia, maka citra hasil penyisipan tidak begitu berbeda dengan citra asli atau citra tanpa penyisipan. Contoh algoritma steganografi dalam ranah domain spasial antara lain Highly Undetectable Steganography (HuGO) yang meminimalisir distorsi dalam embedding untuk steganografi dengan dimensi fitur yang tinggi (Filler & Fridrich, 2010), Wavelet Obtained Weights (WOW) yang menggunakan reciprocal Holder norm untuk mendefinisikan individual pixel cost untuk meminimalisir distorsi (Holub & Fridrich, 2012), High-Pass Low-pass (HILL) yang menggunakan high pass filter untuk mengalokasikan less predictable part dan dua low pass filter untuk mengcluster low cost value (Li et al., 2014), Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) menggunakan universal wavelet relative distortion yang dihitung sebagai penjumlahan dari koefisien yang berubah pada dekomposisi directional filter bank pada citra (Holub et al., 2014), Minimizing the Performance of Optimal Detector (Mi-POD) yang menggunakan estimasi multivariate Gaussian secara lokal pada citra cover(Sedighi et al., 2016).

a) Highly Undetectable Steganography (HuGO)

Pada algoritma dengan ranah domain spasial, penyisipan pesan dilakukan dengan memodifikasi piksel dari cover image, dengan tahapan umum sebagai berikut:

1) Memilih cover image (*x*), dalam hal ini bertipe grayscale dengan nilai piksel antara 0 sampai 255.
2) Menentukan least significant bit (LSB) dari piksel di *x* untuk kemudian di-generate menjadi cover vektor dengan notasi $LSB(x) = v_c = (v_{c1}, \dots, v_{cn}) \in \{0,1\}^n$
3) Penghitungan cost map (ρ) untuk mengurangi distorsi (perubahan pada citra yang disebabkan oleh proses embedding) dengan penghitungan berikut pada Persamaan 1:

$$\rho_i = \sum_{j=1}^{d} w[j] \mid f_x[j] - f_{x\sim x_i}[j] \quad (1)$$

Dimana *f* adalah fitur vector yang degenerate dari occurrence dari nilai triplet (d_1,d_2,d_3) pada citra *x* dengan dimensi *d*, sedangkan $f_{x\sim x_i}$ mengacu pada fitur vector pada citra *x* yang piksel *i* nya sudah dimodifikasi. Weight atau bobot dari triplet (d_1,d_2,d_3) dihitung dengan Persamaan 2 dimana σ, γ merupakan 2 parameter yang dapat dirubah untuk meminimalisir deteksi.

$$w[j] = \frac{1}{\left[\sqrt{d_1^2+d_2^2+d_3^2}+\sigma\right]^\gamma} \quad (2)$$

4) Menggunakan simulator Syndrom Trellis Code untuk meng-generate stego vector $v_s = (v_{s1}, \dots, v_{sn})$ yang menerima masukan berupa cover vector v_c , cost map dan pesan atau message *m*.
5) Melakukan embedding v_s pada cover image *x*, untuk mendapatkan stego image *y*. Proses embedding dilakukan dengan memodifikasi beberapa piksel tertentu, modifikasi ini dapat berupa LSB replacement seperti telah dijelaskan pada bagian 2.1 atau dapat juga dilakukan dengan menambah / mengurangi ”1” dari nilai piksel saat elemen stego vector berbeda dengan cover vector (LSB matching). Algoritma steganografi modern seperti HUGO, WOW dan UNIWARD melakukan embedding dengan LSB matching. Teknik embedding ini juga disebut sebagai ± 1 embedding.

b) Spatial-Universal Wavelet Relative Distortion (S-UNIWARD)

Pada S-Uniward, penyisipan pesan dilakukan dengan langkah yang sama seperti algoritma sebelumnya, yang membedakan adalah pada Langkah 3, untuk S-Uniward distortion function yang digunakan didefinisikan pada wavelet domain sehingga penghitungan cost map (ρ) seperti pada Persamaan 3:

$$\rho_i = \sum_{k=n}^{d} \sum_{u=1}^{n1} \sum_{v=1}^{n2} \frac{|W_{(uv)}^k(x) - W_{(uv)}^k(x\sim x_i)|}{\sigma + |W_{(uv)}^k(x)|} \quad (3)$$

Dimana $W_{(uv)}^k(x)$ adalah koefisien wavelet pada (u,v) $\in$ {1,…, n_1} x {1,…,n_2} posisi untuk *k* sub band pada citra *x* dengan dimensi *d*, sedangkan $W_{(uv)}^k(x\sim x_i)$ mengacu pada koefisien wavelet pada citra *x* yang piksel *i* nya sudah dimodifikasi. Konstanta σ adalah konstante numerical stabilization yang bernilai 1.

c) Wavelet Obtained Weights (WOW)

Pada algoritma WOW, penyisipan pesan dilakukan dengan cara yang hampir sama dengan S-Uniward, yaitu memilih cost yang kecil dari perhitungan 3 arah, yaitu horizontal, vertical dan diagonal. Perbedaan antara wavelet koefisien pada cover dan stego image dihitung kemudian di agregasi sehingga membentuk cost map yang berbeda dari S-Uniward, yaitu sebagai berikut:

$$\rho_i = \sum_{k=1}^{d} \frac{1}{\varepsilon_i^{(k)}} \quad (4)$$

Dimana embedding suitability $\varepsilon_i^{(k)}$ dihitung dengan Persamaan 5:

$$\varepsilon_i^{(k)} = \sum_{u=1}^{n1}\sum_{v=1}^{n2} |W_{(uv)}^k(x)| * |W_{(uv)}^k(x) - W_{(uv)}^k(x \sim x_i)| \quad (5)$$

Pembahasan mengenai algoritma High-Pass Low-pass (HILL) (Li et al., 2014) dan Minimizing the Performance of Optimal Detector (Mi-POD) (Sedighi et al., 2016) berkaitan dengan WOW, karena kedua algoritma tersebut merupakan varian dari algoritma WOW.

d) High-Pass Low-pass (HILL)

Varian lain dari algoritma WOW adalah High-Pass Low-pass (HILL) (Li et al., 2014) dan Minimizing the Performance of Optimal Detector (Mi-POD) (Sedighi et al., 2016). Perbedaannya adalah pada cost function, HILL mengganti 3 arah untuk wavelet decomposition, yaitu horizontal, vertical dan diagonal menjadi 1 high pass filter (*H*) yang bertujuan memberi nilai piksel yang bertekstur dengan nilai cost yang kecil dan 2 low pass filter (*L*) yang bertujuan membuat nilai dengan cost rendah makin terkumpul (clustered) seperti pada Persamaan 6, sehingga karakteristik algoritma HILL membuat penyisipan pesan terkonsentasi (tercluster) pada area bertekstur kasar.

$$W^{(k)} = x * H^{(k)} \; ; \; \varepsilon^{(k)} = \left|W^{(k)}\right| * L_1 \; ; \; \rho = \sum_{k=1}^{d} \frac{1}{\varepsilon^{(k)}} * L_2 \quad (6)$$

e.) Minimizing the Performance of Optimal Detector (Mi-POD)

Algoritma selanjutnya adalah MiPOD, yang memiliki karakteristik dalam penghitungan cost function tidak menggunakan perhitungan nilai piksel, namun menggantinya dengan nilai rata-rata perubahan yang terjadi akibat embedding (embedding change rate, *p*) dengan perhitungan cost ρ seperti persamaan 7.

$$\rho_i = \ln\left({}^{1}/_{p_i} - 2\right) \quad (7)$$

2.3 Steganografi Domain Transform

Transform domain embedding dapat didefinisikan sebagai teknik embedding yang proses penyematan pesan rahasianya ada dalam domain frekuensi sinyal. Domain transform memodifikasi hasil transformasi sinyal dalam ranah transform (hasil transformasi dari ranah spasial ke ranah lain (misalnya ranah frekuensi). Algoritma steganografi mencoba menambahkan pseudo-noise ke dalam citra cover. Salah satu cara untuk melakukan steganografi dalam domain transform antara lain menggunakan koefisien Discrete Cosine Transformation (DCT) dan Discrete Wavelet Transformation (DWT).

Sebelum membahas mengenai algoritma steganografi dalam domain transform, diperlukan penjelasan mengenai kompresi untuk memahami koefisien Discrete Cosine Transformation (DCT) dan Discrete Wavelet Transformation (DWT) yang ada dalam algoritma steganografi domain transform. DCT (Discrete Cosine Transform) digunakan pada kompresi citra yang bersifat lossy compression, seperti JPEG.

Proses kompresi pada JPEG dimulai dengan mengkonversi ruang warna dari citra asli dari ruang warna RGB ke YCbCr kemudian memecah cover image menjadi 8*8 blok piksel. DCT didesain untuk bekerja pada rentang nilai -128 sampai 127, oleh karena itu dilakukan shift blok dari 0 sampai dengan 255 ke shift blok -128 sampai dengan 127, baru dihitung persamaan DCT seperi pada Persamaan 8.

$$F(x,y) = \frac{2}{M} C(x)C(y) \sum_{x=0}^{M-1}\sum_{y=0}^{M-1} f(i,j) \cos\left[\frac{(2i+1)\pi x}{2M}\right] cos\left[\frac{(2j+1)\pi y}{2M}\right] \quad (8)$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & untuk\ x=0 \\ 0 & x \neq 0 \end{cases} \quad (9)$$

Dimana F(x,y) adalah koefisien frekuensi dari M*M blok. *C(x), C(y)* adalah constant scale factor dengan perhitungan pada Persamaan 9, *f(i,j)* adalah elemen citra grayscale yang direpresentasikan dalam matriks f. M adalah ukuran blok DCT, yaitu 8, sehingga *x,y* berada dalam range 0 sampai 7. Pada tahapan selajutnya, masing-masing blok DCT dilakukan pemecahan menjadi matriks kuantisasi luminance dan matriks kuantisasi chrominance yang kemudian diberikan kuantisasi dengan quantization table (QT) yang sesuai dengan standar Annex, untuk selanjutnya dilakukan round atau pembulatan nilai. Pada tahap ini, mayoritas data yang redundan dan mengandung noise dihilangkan, itulah mengapa kompresi ini bersifat lossy. Meskipun begitu, beberapa kamera memiliki built-in QT yang berbeda dengan standar Annex, hal ini bertujuan menyeimbangkan trade-off antara kompresi dan quality factor (QF).

Kompresi JPEG kemudian dilanjutkan dengan zig-zag reordering matrix, RunLength Encoding (untuk koefisien AC), DPCM untuk koefisien DC dan kemudian Huffman encoding. Langkah selanjutnya adalah dekompresi dengan IDCT (Inverse Discrete Cosine Transform) atau transformasi pembalikan, kemudian shift blok akan kembalikan dari -128 sampai dengan 127 ke shift blok 0 sampai dengan 255 dan susun blok pada citra. Persamaan 10 menunjukkan perhitungan IDCT:

$$f(i,j) = \frac{2}{M} \sum_{x=0}^{M-1}\sum_{y=0}^{M-1} C(x)C(y)F(x,y) \cos\left[\frac{(2i+1)\pi x}{2M}\right] \quad (10)$$

Contoh algoritma steganografi dalam ranah domain transform antara lain Uniform Embedding Distortion (UED) yang memberikan fungsi distorsi seragam untuk side informed dan non-side informed JPEG (Guo et al., 2014), Uniform Embedding

Revised Distortion (U-ERD) yang memperbaiki algoritma UED dengan menggunakan semua koefisien DCT seperti DC dan zero, non zero AC sebagai cover element (Guo et al., 2015), JPEG-Universal Wavelet Relative Distortion (J-UNIWARD) menggunakan universal wavelet relative distortion yang dihitung sebagai penjumlahan dari koefisien yang berubah pada dekomposisi directional filter bank pada citra(Holub et al., 2014).

a) JPEG-Universal Wavelet Relative Distortion (J-UNIWARD)

Prinsip embedding J-Uniward sama dengan S-Uniward, yaitu menggunakan menggunakan universal wavelet relative distortion, perbedaannya adalah S-Uniward untuk domain spasial (tipe citra pgm) sedangkan J-Uniward untuk domain transform (tipe citra jpeg). Penyisipan pesan melalui koefisien DCT memiliki tahapan sebagai berikut:

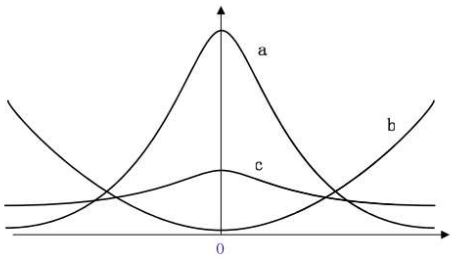
a) Pembacaan nilai piksel dalam cover image
b) Konversi biner untuk embedded message yang akan disisipkan
c) Cover image dipecah menjadi 8*8 blok piksel
d) DCT (Discrete Cosine Transform) dilakukan pada masing-masing blok
e) Kuantisasi dengan QT dilakukan untuk mengkompresi masing-masing blok
f) LSB seperti yang dilakukan pada S-Uniward pada bagian 2.1.2 dihitung untuk masing-masing koefisien DCT
g) Stego image telah berisi embedded message

b) Uniform Embedding Distortion (UED) dan Uniform Embedding Revised Distortion (U-ERD)

Citra digital pada domain transform yang disisipi pesan dengan algoritma UED dan UERD masih menggunakan prinsip penyisipan dengan DCT. Pada Uniform Embedding Distortion (UED) menggunakan syndrome trellis code (STC) dan fungsi distorsi seragam (uniform embedding), dimana modifikasi embedding disebar secara seragam pada koefisien DCT terkuantisasi seperti perhitungan fungsi distorsi ρ pada Persamaan 11, dimana x adalah koefisien DCT yang terdistribusi berdasarkan distribusi Laplacian seperti Gambar 5 kurva a, kemudian probabilitas dari sebuah koefisien x naik secara monotonik berdasarkan magnitude $|x|$ seperti kurva b, sedangkan kurva c adalah uniform distribution dari koefisien DCT setelah disisipi pesan dengan UED.

$$\rho(x) = {}^{1}/_{|x|} \qquad (11)$$

Tujuan dari UED adalah meminimalisir deteksi 1st dan 2nd order statistik (histogram dan co-occurrence). Ciri dari UED adalah hanya menggunakan non zero AC coefficient, yang kemudian diperbaiki dengan menggunakan semua koefisien DCT seperti DC dan zero, non zero AC pada Uniform Embedding Revised Distortion (U-ERD) karena DC dan zero AC coefficient juga berperan penting dalam Uniform Embedding Distortion.



Gambar 5 **Kurva yang menggambarkan Uniform Embedding Distortion**
Sumber: (Guo et al., 2015)

Selain Discrete Cosine Transform, perlu dipahami juga mengenai Discrete Wavelet Transform (DWT). DWT menjadi konsep yang mendasari beberapa algoritma steganografi domain spasial seperti S-Uniward dan WOW yang telah kita bahas pada bagian sebelumnya. DWT membagi sebuah dimensi sinyal menjadi dua bagian, yaitu frekuensi tinggi dan frekuensi rendah, atau disebut dekomposisi. Sebuah sinyal dilewatkan melalui highpass filter untuk menganalisis frekuensi tinggi (*H*), dan dilewatkan melalui lowpass filter untuk menganalisis frekuensi rendah(*L*). Keluaran dari highpass filter dan lowpass filter ini menghasilkan koefisien DWT, kemudian citra asli dapat direkonstruksi dengan Inverse Discrete Wavelet Transform (IDWT).

Contoh DWT yang paling sederhana adalah menggunakan Haar-DWT dengan dua operasi yaitu vertikal dan horizontal. Pada operasi horizontal, dilakukan penjumlahan dan pengurangan nilai piksel yang bersebelahan. Hasil penjumlahan diletakkan di sebelah kiri piksel citra, sedangkan pengurangan diletakkan di sebelah kanan. Penghitungan koefisien DWT seperti pada persamaan 12 dan 13. $y_{tinggi}[k]$ merupakan hasil dari *highpass filter* dan $y_{rendah}[k]$ merupakan hasil dari *lowpassfilter* dimana $k = 1,2,3,\ldots,\frac{1}{2}K$, $x[n]$ merupakan sinyal asal, $h[n]$ adalah *highpass filter*, dan $g[n]$ adalah *lowpass filter* .

$$y_{tinggi}[k] = \sum_n x[n]h[2k-n] = \frac{X_{2n-1}-X_{2n}}{2} \quad (12)$$

$$y_{renda}\ [k] = \sum_n x[n]g[2k-n] = \frac{X_{2n-1}+X_{2n}}{2} \quad (13)$$

Algoritma steganografi dalam domain transform menggunakan koefisien DWT ini untuk melakukan penyisipan (embedding), metode penyisipan pesan dengan koefisien DWT yang paling sederhana adalah menggunakan substitusi LSB. Langkah terakhir untuk mendapatkan stego image adalah

menggunakan inverse DWT (IDWT). Proses rekontruksi merupakan kebalikan dari proses dekomposisi seperti pada Persamaan 14.

$$x[n] = \sum_k (y_{tinggi[k]}\, h[-n+2k]\, y_{rendah[k]} g[-n+2k]) \quad (14)$$

3. Hasil dan Pembahasan

Pada bagian ini akan dibahas mengenai karakteristik algoritma steganografi dalam domain spasial dan domain transform. Tahapan penelitian dimulai dengan citra cover bertipe .pgm yang diberikan kode untuk masing-masing algoritma (http://dde.binghamton.edu/download/stego_algorithms/). Kemudian kita akan membandingkan citra cover dan citra hasil penyisipan pesan. Perlu dipahami dulu mengenai pesan rahasia yang disisipkan, pesan rahasia yang disisipkan dapat berupa text atau image. Namun, pada penelitian mengenai steganografi, umumnya penyisipan pesan berupa pesan random yang dihitung secara kuantitatif, dalam hal ini kita kenal dengan istilah payload.

Sebuah citra disisipkan pesan rahasia dengan payload sebesar 0,4 bpnzac (bits per non-zero AC DCT coefficient) yang artinya ada 2638 data tersembunyi dalam setiap 53 kB file citra. Bpnzac biasanya digunakan untuk domain transform, sedangkan untuk domain spasial menggunakan bpp (bit per pixel). Misalnya, citra disisipkan pesan rahasia dengan payload sebesar 0,4 bpp artinya 40% bit dalam sebuah piksel adalah pesan rahasia, sedangkan 60% lainnya adalah cover object (tidak mengandung pesan rahasia).

Langkah terakhir adalah menganalisis citra cover dan citra hasil penyisipan pesan sehingga diketahui karakteristik dari masing-masing algoritma steganografi yang digunakan dalam penyisipan pesan. Citra yang digunakan adalah citra 2D atau grayscale bertipe .pgm untuk domain spasial, sedangkan pada domain transform citra diubah ke format jpeg. Semuanya dengan ukuran 512 x 512.

Gambar 6 **Citra cover (a) Domain Spasial (b) Domain Transform**
Sumber: http://dde.binghamton.edu/download/stego_algorithms/

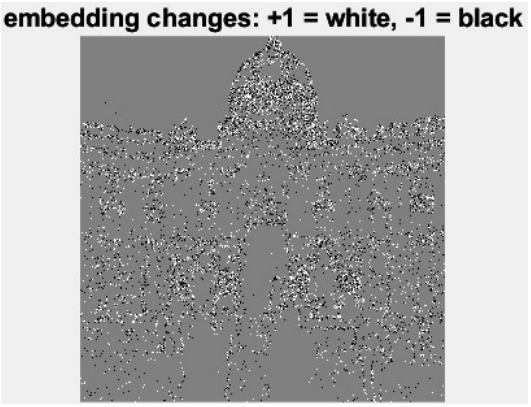
Perangkat yang digunakan untuk kompilasi adalah Matlab R2022b. Citra cover yang akan dianalisis seperti terdapat dalam Gambar 5a untuk domain spasial dan Gambar 5b untuk domain transform, gambar ini merupakan gambar standar yang sudah ada dalam source code sebelumnya.

Secara umum, algoritma steganografi melakukan penyisipan pesan (embedding) sekaligus mengurangi distorsi pada cover image. Distorsi adalah perubahan pada citra yang disebabkan oleh proses embedding. Dengan kata lain, embedding dilakukan pada area di cover image yang sulit untuk dideteksi dan distorsi diminimalisir lewat cost map, cost map inilah yang membedakan antara 1 algoritma steganografi dengan yang lain, seperti terdapat dalam pembahasan berikutnya. Berikut akan dibahas satu persatu mengenai algoritma steganografi dalam domain spasial dan transform.

Pada bagian ini akan diambil contoh 3 algoritma steganografi yaitu Highly Undetectable Steganography (HuGO) (Filler & Fridrich, 2010), Wavelet Obtained Weights (WOW) (Holub & Fridrich, 2012), Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) (Holub et al., 2014).

a) Highly Undetectable Steganography (HuGO)

Hasil implementasi HuGO pada citra hasil embedding menyebabkan perbedaan yang disebut dengan stego noise seperti terdapat pada Gambar 7.



Gambar 7 Perbedaan citra cover dan citra hasil penyisipan dengan HUGO

Analisis yang dapat disampaikan adalah karakteristik citra yang disisipi dengan HuGO melakukan embedding pada area berkontur atau bertekstur, hal ini disebabkan weight pada persamaan 2 yang bernilai tinggi pada area berkontur atau bertekstur. Berikut adalah fungsi utama untuk algoritma HuGO (http://dde.binghamton.edu/download/stego_algorithms/):

```
function [stego, distortion] =
HUGO_like(cover, payload, params)

cover = double(cover);
wetCost = 10^8;
responseP1 = [0; 0; -1; +1; 0; 0];

% create mirror padded cover image
padSize = 3;
```

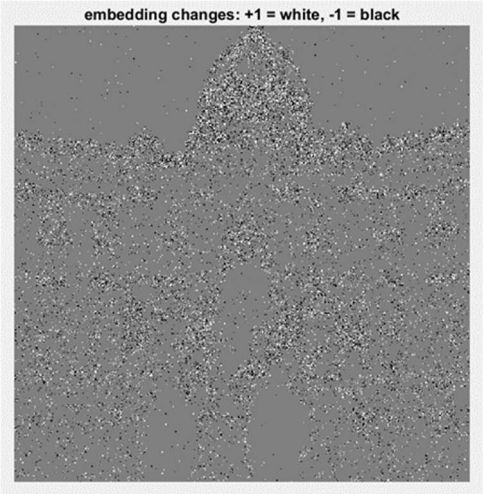
```
coverPadded = padarray(cover, [padSize
padSize], 'symmetric');

% create residuals
C_Rez_H = coverPadded(:, 1:end-1) -
coverPadded(:, 2:end);
C_Rez_V = coverPadded(1:end-1, :) -
coverPadded(2:end, :);
C_Rez_Diag = coverPadded(1:end-1, 1:end-1) -
coverPadded(2:end, 2:end);
C_Rez_MDiag = coverPadded(1:end-1, 2:end) -
coverPadded(2:end, 1:end-1);

stego = cover;
% initialize stego image
stegoPadded = coverPadded;

% create residuals
S_Rez_H = stegoPadded(:, 1:end-1) -
stegoPadded(:, 2:end);
S_Rez_V = stegoPadded(1:end-1, :) -
stegoPadded(2:end, :);
S_Rez_Diag = stegoPadded(1:end-1, 1:end-1) -
stegoPadded(2:end, 2:end);
S_Rez_MDiag = stegoPadded(1:end-1, 2:end) -
stegoPadded(2:end, 1:end-1);

rhoM1 = zeros(size(cover));
% declare cost of -1 change
rhoP1 = zeros(size(cover));
% declare cost of +1 change

%% Iterate over elements in the sublattice
for row=1:size(cover, 1)
    for col=1:size(cover, 2)
        D_P1 = 0;
        D_M1 = 0;

        % Horizontal
        cover_sub = C_Rez_H(row+3,
col:col+5)';
        stego_sub = S_Rez_H(row+3,
col:col+5)';

        stego_sub_P1 = stego_sub +
responseP1;
        stego_sub_M1 = stego_sub -
responseP1;

        D_M1 = D_M1 +
GetLocalDistortion(cover_sub, stego_sub_M1,
params);
        D_P1 = D_P1 +
GetLocalDistortion(cover_sub, stego_sub_P1,
params);

        % Vertical
        cover_sub = C_Rez_V(row:row+5,
col+3);
        stego_sub = S_Rez_V(row:row+5,
col+3);

        stego_sub_P1 = stego_sub +
responseP1;
        stego_sub_M1 = stego_sub -
responseP1;

        D_M1 = D_M1 +
GetLocalDistortion(cover_sub, stego_sub_M1,
params);
        D_P1 = D_P1 +
GetLocalDistortion(cover_sub, stego_sub_P1,
params);

        % Diagonal
        cover_sub = [C_Rez_Diag(row, col);
C_Rez_Diag(row+1, col+1); C_Rez_Diag(row+2,
col+2); C_Rez_Diag(row+3, col+3);
C_Rez_Diag(row+4, col+4); C_Rez_Diag(row+5,
col+5)];
        stego_sub = [S_Rez_Diag(row, col);
S_Rez_Diag(row+1, col+1); S_Rez_Diag(row+2,
col+2); S_Rez_Diag(row+3, col+3);
S_Rez_Diag(row+4, col+4); S_Rez_Diag(row+5,
col+5)];

        stego_sub_P1 = stego_sub +
responseP1;
        stego_sub_M1 = stego_sub -
responseP1;

        D_M1 = D_M1 +
GetLocalDistortion(cover_sub, stego_sub_M1,
params);
        D_P1 = D_P1 +
GetLocalDistortion(cover_sub, stego_sub_P1,
params);

        % Minor Diagonal
        cover_sub = [C_Rez_MDiag(row,
col+5); C_Rez_MDiag(row+1, col+4);
C_Rez_MDiag(row+2, col+3);
C_Rez_MDiag(row+3, col+2);
C_Rez_MDiag(row+4, col+1);
C_Rez_MDiag(row+5, col)];
        stego_sub = [S_Rez_MDiag(row,
col+5); S_Rez_MDiag(row+1, col+4);
S_Rez_MDiag(row+2, col+3);
S_Rez_MDiag(row+3, col+2);
S_Rez_MDiag(row+4, col+1);
S_Rez_MDiag(row+5, col)];

        stego_sub_P1 = stego_sub +
responseP1;
        stego_sub_M1 = stego_sub -
responseP1;

        D_M1 = D_M1 +
GetLocalDistortion(cover_sub, stego_sub_M1,
params);
        D_P1 = D_P1 +
GetLocalDistortion(cover_sub, stego_sub_P1,
params);

        rhoM1(row, col) = D_M1;
        rhoP1(row, col) = D_P1;
    end
end

% truncation of the costs
rhoM1(rhoM1>wetCost) = wetCost;
rhoP1(rhoP1>wetCost) = wetCost;

rhoP1(cover == 255) = wetCost;
rhoM1(cover == 0) = wetCost;

%% Embedding
% embedding simulator - params.qarity \in
{2,3}
stego = EmbeddingSimulator(cover, rhoP1,
rhoM1, round(numel(cover)*payload), false);

% compute distortion
distM1 = rhoM1(stego-cover==-1);
distP1 = rhoP1(stego-cover==1);
distortion = sum(distM1) + sum(distP1);

end
```

Kelebihan algoritma HuGO adalah menggunakan distribusi Gibbs untuk meminimalisir distorsi akibat embedding, yang dapat

mensimulasikan impact dari embedding optimal (Filler & Fridrich, 2010). Sedangkan kekurangannya adalah lebih mudah terdeteksi daripada S-Uniward dengan steganalyzer GBRAS-Net pada payload 0,2 bpp. Akurasi HuGO 74,6% dan S-Uniward 73,6% (Reinel et al., 2021).

b.) Spatial-Universal Wavelet Relative Distortion (S-UNIWARD)

Hasil implementasi S-Uniward pada citra hasil embedding menyebabkan perbedaan yang disebut dengan stego noise seperti terdapat pada Gambar 8. Analisis yang dapat disampaikan adalah karakteristik citra yang disisipi dengan S-Uniward melakukan embedding pada area berkontur atau bertekstur, hal ini disebabkan terdapat 3 arah untuk wavelet decomposition, yaitu horizontal, vertical dan diagonal.



Gambar 8 Perbedaan citra cover dan citra hasil penyisipan dengan S-Uniward

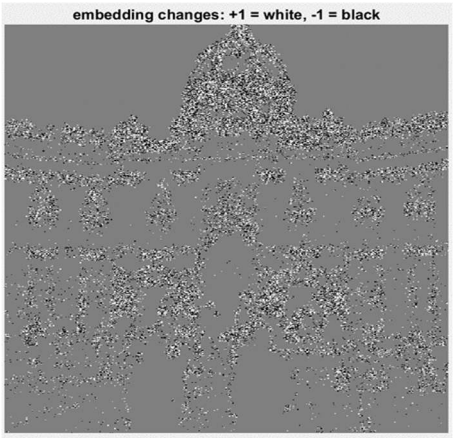
Berdasarkan persamaan 3, cost ρ_i bernilai kecil pada area yang bertekstur, sehingga piksel inilah yang diberikan penyisipan pesan. Perbedaan dengan HUGO, intensitas pada S-Uniward lebih renggang, terlihat dari Gambar 8 lebih sedikit titik daripada Gambar 7. Berikut adalah fungsi utama untuk algoritma S-Uniward (http://dde.binghamton.edu/download/stego_algorithms/):

```
function stego = S_UNIWARD(coverPath,
payload)
sgm = 1;

%% Get 2D wavelet filters - Daubechies 8
% 1D high pass decomposition filter
hpdf = [-0.0544158422, 0.3128715909, -
0.6756307363, 0.5853546837, 0.0158291053, -
0.2840155430, -0.0004724846, 0.1287474266,
0.0173693010, -0.0440882539, ...
        -0.0139810279, 0.0087460940,
0.0048703530, -0.0003917404, -0.0006754494,
-0.0001174768];
% 1D low pass decomposition filter
lpdf = (-1).^(0:numel(hpdf)-
1).*fliplr(hpdf);

% construction of 2D wavelet filters
F{1} = lpdf'*hpdf;
F{2} = hpdf'*lpdf;
F{3} = hpdf'*hpdf;

%% Get embedding costs
% inicialization
cover = double(imread(coverPath));
wetCost = 10^8;
[k,l] = size(cover);

% add padding
padSize = max([size(F{1})'; size(F{2})';
size(F{3})']);
coverPadded = padarray(cover, [padSize
padSize], 'symmetric');

xi = cell(3, 1);
for fIndex = 1:3
    % compute residual
    R = conv2(coverPadded, F{fIndex},
'same');
    % compute suitability
    xi{fIndex} = conv2(1./(abs(R)+sgm),
rot90(abs(F{fIndex}), 2), 'same');
    % correct the suitability shift if
filter size is even
    if mod(size(F{fIndex}, 1), 2) == 0,
xi{fIndex} = circshift(xi{fIndex}, [1, 0]);
end;
    if mod(size(F{fIndex}, 2), 2) == 0,
xi{fIndex} = circshift(xi{fIndex}, [0, 1]);
end;
    % remove padding
    xi{fIndex} =
xi{fIndex}(((size(xi{fIndex}, 1)-
k)/2)+1:end-((size(xi{fIndex}, 1)-k)/2),
((size(xi{fIndex}, 2)-l)/2)+1:end-
((size(xi{fIndex}, 2)-l)/2));
end

% compute embedding costs \rho
rho = xi{1} + xi{2} + xi{3};

% adjust embedding costs
rho(rho > wetCost) = wetCost; % threshold on
the costs
rho(isnan(rho)) = wetCost; % if all xi{} are
zero threshold the cost
rhoP1 = rho;
rhoM1 = rho;
rhoP1(cover==255) = wetCost; % do not embed
+1 if the pixel has max value
rhoM1(cover==0) = wetCost; % do not embed -1
if the pixel has min value

%% Embedding simulator
stego = EmbeddingSimulator(cover, rhoP1,
rhoM1, payload*numel(cover), false);
```

Kelebihan S-Uniward adalah memiliki akurasi lebih rendah daripada HuGO saat dilakukan steganalisis dengan GBRAS-Net pada payload 0,2 bpp, yaitu 73,6% (Reinel et al., 2021). Sedangkan kekurangan S-Uniward adalah output dari directional high pass filter tidak diberi bobot, sehingga ini memberikan peluang penelitian baru pada algoritma WOW untuk memperbaiki hal ini. Tujuannya adalah melakukan embedding pada area yang memiliki tekstur sangat kasar dan menghindari area sudut yang bersih (Holub & Fridrich, 2012).

c.) Wavelet Obtained Weights (WOW)

Hasil implementasi WOW pada citra hasil embedding menyebabkan perbedaan yang disebut dengan stego noise seperti terdapat pada Gambar 9.



Gambar 9 Perbedaan citra cover dan citra hasil penyisipan dengan WOW

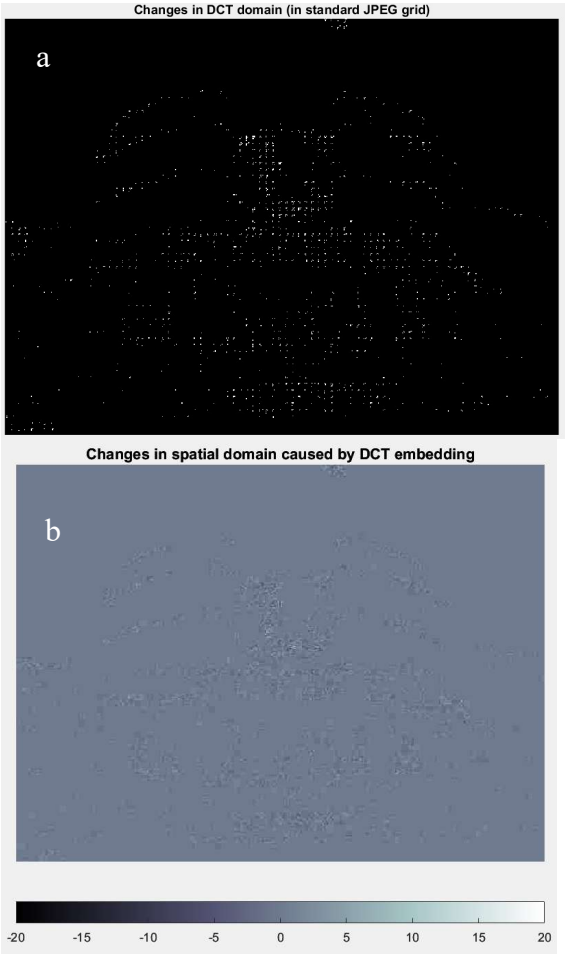
Analisis yang dapat disampaikan adalah karakteristik citra yang disisipi dengan melakukan embedding pada area berkontur atau bertekstur dan menghindari area dengan tekstur halus, karena area denga tekstur halus atau clean edge diberikan embedding cost yang besar. Perbedaan WOW dengan S-Uniward dan HuGO adalah WOW sedikit menyisipkan pesan pada area dengan tekstur halus seperti pada Gambar 9. Berikut adalah fungsi utama untuk algoritma WOW (http://dde.binghamton.edu/download/stego_algorithms/):

```
function [stego, distortion] = WOW(cover,
payload, params)
%% Get 2D wavelet filters - Daubechies 8
% 1D high pass decomposition filter
hpdf = [-0.0544158422, 0.3128715909, -
0.6756307363, 0.5853546837, 0.0158291053, -
0.2840155430, -0.0004724846, 0.1287474266,
0.0173693010, -0.0440882539, ...
        -0.0139810279, 0.0087460940,
0.0048703530, -0.0003917404, -0.0006754494,
-0.0001174768];
% 1D low pass decomposition filter
lpdf = (-1).^(0:numel(hpdf)-
1).*fliplr(hpdf);
% construction of 2D wavelet filters
F{1} = lpdf'*hpdf;
F{2} = hpdf'*lpdf;
F{3} = hpdf'*hpdf;

%% Get embedding costs
% inicialization
cover = double(cover);
p = params.p;
wetCost = 10^10;
sizeCover = size(cover);

% add padding
padSize = max([size(F{1})'; size(F{2})';
size(F{3})']);
coverPadded = padarray(cover, [padSize
padSize], 'symmetric');

% compute directional residual and
suitability \xi for each filter
xi = cell(3, 1);
for fIndex = 1:3
    % compute residual
    R = conv2(coverPadded, F{fIndex},
'same');

    % compute suitability
    xi{fIndex} = conv2(abs(R),
rot90(abs(F{fIndex}), 2), 'same');
    % correct the suitability shift if
filter size is even
    if mod(size(F{fIndex}, 1), 2) == 0,
xi{fIndex} = circshift(xi{fIndex}, [1, 0]);
end;
    if mod(size(F{fIndex}, 2), 2) == 0,
xi{fIndex} = circshift(xi{fIndex}, [0, 1]);
end;
    % remove padding
    xi{fIndex} =
xi{fIndex}(((size(xi{fIndex}, 1)-
sizeCover(1))/2)+1:end-((size(xi{fIndex},
1)-sizeCover(1))/2), ((size(xi{fIndex}, 2)-
sizeCover(2))/2)+1:end-((size(xi{fIndex},
2)-sizeCover(2))/2));
end

% compute embedding costs \rho
rho = ( (xi{1}.^p) + (xi{2}.^p) + (xi{3}.^p)
) .^ (-1/p);

% adjust embedding costs
rho(rho > wetCost) = wetCost; % threshold on
the costs
rho(isnan(rho)) = wetCost; % if all xi{} are
zero threshold the cost
rhoP1 = rho;
rhoM1 = rho;
rhoP1(cover==255) = wetCost; % do not embed
+1 if the pixel has max value
rhoM1(cover==0) = wetCost; % do not embed -1
if the pixel has min value

%% Embedding simulator
stego = EmbeddingSimulator(cover, rhoP1,
rhoM1, payload*numel(cover), false);
distortion_local = rho(cover~=stego);
distortion = sum(distortion_local);
```

Kelebihan algoritma WOW adalah memperbaiki algoritma S-Uniward dalam melakukan embedding pada area yang memiliki tekstur sangat kasar dan menghindari area sudut yang bersih (Holub & Fridrich, 2012). Caranya adalah memberikan bobot kepada output dari directional high pass filter setelah perubahan 1 piksel, kemudian melakukan agregasi dengan reciprocal Holder norm. Kekurangan algoritma WOW adalah memiliki akurasi lebih tinggi daripada S-Uniward saat dilakukan steganalisis dengan GBRAS-Net pada payload 0,2 bpp, yaitu 80,3% (Reinel et al., 2021).

Pembahasan berikutnya adalah algoritma steganografi domain transform. Algoritma dengan ranah domain transform memodifikasi hasil transformasi sinyal dalam ranah transform (hasil transformasi dari ranah spasial ke ranah lain (misalnya ranah frekuensi). Pada subbab ini akan diambil contoh algoritma steganografi yaitu JPEG-

Universal Wavelet Relative Distortion (J-UNIWARD) (Holub et al., 2014).



Gambar 10 Visualisasi citra hasil penyisipan dengan J-Uniward

Karakteristik umum penyisipan pesan dalam domain transform adalah sulit dideteksi dengan visual attack sehingga tidak ada perbedaan mencolok antara citra cover dan stego yang dapat dilihat secara visual seperti yang telah dipaparkan dalam domain spasial. Ada beberapa perbedaan yang dapat diamati dalam J-uniward seperti Gambar 10 (a) yang memvisualisasikan perubahan nilai koefisien DCT yang berubah akibat adanya penyisipan dengan J-Uniward, terlihat di area dengan tekstur kasar diberikan penyisipan pesan, dan Gambar 10 (b) memvisualisasikan perubahan pada domain spasial yang menyiratkan bahwa ada perubahan yang juga terjadi akibat DCT embedding seperti Gambar 10(b).

Dari karakteristik algoritma steganografi yang telah dibahas, kelebihan J-Uniward dibandingkan adalah dapat diimplementasikan pada citra bertipe JPEG yang banyak digunakan dalam pertukaran data saat ini dan kekurangan J-Uniward adalah adanya variable lain (JPEG quality factor) yang mempengaruhi kemampuan embedding, hal ini yang membedakannya dengan embedding pada domain spasial.

4. Kesimpulan

Steganografi pada citra digital dalam domain spasial maupun domain transform menyebabkan perubahan pada citra yang disisipi. Pada bagian hasil dan pembahasan telah dibahas mengenai karakteristik citra hasil penyisipan dengan berbagai algoritma steganografi yang secara umum menyisipkan pesan pada area bertekstur. Analisis mengenai karakteristik ini dapat membuka peluang penelitian baru.

Peluang penelitian untuk steganografi secara umum adalah citra digital yang telah disisipi pesan rahasia dengan berbagai algoritma steganografi menyebabkan perubahan karakteristik citra hasil penyisipan,, perubahan inilah yang kemudian digunakan untuk melakukan steganalisis seperti pada (Reinel et al., 2021). Steganalisis adalah aktivitas analisis citra yang memiliki tujuan utama mendeteksi apakah sebuah citra suspect memiliki pesan rahasia di dalamnya.

Peluang penelitian kedua, setelah sebuah citra dideteksi mengandung pesan rahasia, penelitian lanjutan akan dilakukan, misalnya menentukan panjang pesan serta algoritma steganografi yang digunakan. Metode yang dapat digunakan untuk memprediksi algoritma steganografi yang digunakan adalah klasifikasi multi kelas, seperti multi class SVM pada (Pevny & Fridrich, 2007).

Daftar Pustaka

Bas, P., Filler, T., & Pevný, T. (2011). "Break our steganographic system": The ins and outs of organizing BOSS. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6958 LNCS*. https://doi.org/10.1007/978-3-642-24178-9_5

Bashir, B., & Selwal, A. (2021). Towards Deep Learning-Based Image Steganalysis: Practices and Open Research Issues. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3883330

Filler, T., & Fridrich, J. (2010). Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, *5*(4). https://doi.org/10.1109/TIFS.2010.2077629

Fridrich, J. (2012). Steganography in digital media: Principles, algorithms, and applications. In *Steganography in Digital Media* (Vol. 9780521190190). https://doi.org/10.1017/CBO9781139192903

Guo, L., Ni, J., & Shi, Y. Q. (2014). Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, *9*(5). https://doi.org/10.1109/TIFS.2014.2312817

Guo, L., Ni, J., Su, W., Tang, C., & Shi, Y. Q. (2015). Using Statistical Image Model for JPEG Steganography: Uniform Embedding

Revisited. *IEEE Transactions on Information Forensics and Security*, *10*(12). https://doi.org/10.1109/TIFS.2015.2473815

Guttikonda, J. B., & Sridevi, R. (2019). A new steganalysis approach with an efficient feature selection and classification algorithms for identifying the stego images. *Multimedia Tools and Applications*, *78*(15), 21113–21131. https://doi.org/10.1007/s11042-019-7168-5

Holub, V., & Fridrich, J. (2012). Designing steganographic distortion using directional filters. *WIFS 2012 - Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security*. https://doi.org/10.1109/WIFS.2012.6412655

Holub, V., Fridrich, J., & Denemark, T. (2014). Universal distortion function for steganography in an arbitrary domain. *Eurasip Journal on Information Security*, *2014*. https://doi.org/10.1186/1687-417X-2014-1

Kodar, A. (2017). Implementation of Steganography in Image Media Using Algorithm LSB (Least Significant Bit). *International Research Journal of Computer Science*, *4*(8). https://doi.org/10.26562/irjcs.2017.aucs10081

Li, B., Wang, M., Huang, J., & Li, X. (2014). A new cost function for spatial image steganography. *2014 IEEE International Conference on Image Processing, ICIP 2014*. https://doi.org/10.1109/ICIP.2014.7025854

Munir, R. (2019). Kriptografi. In *2*.

Pevny, T., & Fridrich, J. (2007). Merging Markov and DCT features for multi-class JPEG steganalysis. *Security, Steganography, and Watermarking of Multimedia Contents IX*, *6505*. https://doi.org/10.1117/12.696774

Płachta, M., Krzemień, M., Szczypiorski, K., & Janicki, A. (2022). Detection of Image Steganography Using Deep Learning and Ensemble Classifiers. *Electronics (Switzerland)*, *11*(10). https://doi.org/10.3390/electronics11101565

Reinel, T. S., Brayan, A. A. H., Alejandro, B. O. M., Alejandro, M. R., Daniel, A. G., Alejandro, A. G. J., Buenaventura, B. J. A., Simon, O. A., Gustavo, I., & Raul, R. P. (2021). GBRAS-Net: A Convolutional Neural Network Architecture for Spatial Image Steganalysis. *IEEE Access*, *9*. https://doi.org/10.1109/ACCESS.2021.3052494

Sedighi, V., Cogranne, R., & Fridrich, J. (2016). Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, *11*(2). https://doi.org/10.1109/TIFS.2015.2486744

Singh, S., & Attri, V. K. (2015). Dual Layer Security of data using LSB Image Steganography Method and AES Encryption Algorithm. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(5). https://doi.org/10.14257/ijsip.2015.8.5.27

Veena, S. T., & Arivazhagan, S. (2018). Quantitative steganalysis of spatial LSB based stego images using reduced instances and features. *Pattern Recognition Letters*, *105*. https://doi.org/10.1016/j.patrec.2017.08.016