Jurnal
ILMU KEPOLISIAN
ISSN: 2620-5025
e-ISSN: 2621-8410

Volume 17 | Nomor 1| April 2023

PENGAMANAN DATA DIGITAL MENGGUNAKAN TEKNIK *INTEGER WAVELET TRANSFORM* DAN *LEAST SIGNIFICANT BIT*

Suryadi MT[1,2,3 *], Orchini Liviana[1], Yudi Satria[1]
[1]Departemen Matematika, Universitas Indonesia, Depok, Indonesia, 16424
[2]Program Studi Kajian Ilmu Kepolisian, SKSG, Universitas Indonesia, Jakarta, Indonesia, 10430
[3]Sekolah Tinggi Ilmu Kepolisian PTIK, Jakarta, Indonesia, 12160
[*]e-maill: yadi.mt@sci.ui.ac.id

Abstract

Data is an asset for all parties. So that it must always be guarded or secured so that it is not misused by irresponsible parties. One of the methods used for data security is steganography. In this paper, efforts to secure data are discussed through the development of steganographic algorithms using the haar integer wavelet transform (IWT-Haar) and least significant bit (LSB) techniques to hide secret messages in the form of digital text into digital images (cover images) to produce stego images. The results show that the extracting process is much faster than the embedding process. The PSNR values obtained ranged from 55-71,5 dB (far greater than 40 dB). This shows that the resulting stego image is of very good quality. In plain view, the difference between the cover image and the stego image has not changed significantly.

Key words: digital image, embedding, extraction, haar integer wavelet transform, steganography

Abstrak

Data merupakan asset bagi semua pihak. Sehingga harus selalu dijaga atau diamankan agar tidak disalahgunakan oleh pihak yang tidak bertanggung jawab. Salah satu metode yang digunakan untuk pengamanan data adalah steganografi. Pada paper ini, dibahas upaya pengamanan data melalui pembangunan algoritma steganografi menggunakan teknik Haar *integer wavelet transform* (IWT-Haar) dan *least significant bit* (LSB) untuk menyembunyikan pesan rahasia berupa teks digital ke dalam citra digital (*cover image*) sehingga menghasilkan *stego image*. Hasilnya menunjukkan bahwa proses *extracting* jauh lebih cepat dibandingkan proses *embedding*. Nilai PSNR yang diperoleh berkisar pada nilai 55-71,5 dB (jauh lebih besar dari 40 dB). Ini menunjukkan *stego image* yang dihasilkan memiliki kualitas yang sangat baik. Secara kasat mata perbedaan *cover image* dengan *stego image* tidak mengalami perubahan yang signifikan.

Kata Kunci: citra digital, embedding, ekstraksi, Haar integer wavelet transform, steganografi

Pendahuluan

Pada era *big data* saat ini, tidak dipungkiri lagi bahwa data dan informasi adalah merupakan sumberdaya utama bagi semua pihak. Hal tersebut juga mengindikasikan bahwa data dan informasi adalah merupakan asset yang harus senantiasa dipelihara dan dijaga agar tidak disalahgunakan oleh pihak yang tidak bertanggung jawab. Perubahan paradigma tentang data dan informasi tersebut yang selama ini masih dipandang sebagai sumberdaya penunjang atau pendukung harus digaungkan. Hal tersebut sejalan dengan pernyataan Bapak Joko Widodo selaku Presiden Republik Indonesia pada pidato tahunan dalam Sidang Umum MPR tanggal 16 Agustus 2019 yang mengatakan, "Kita harus siaga menghadapi ancaman kejahatan siber termasuk kejahatan penyalahgunaan data. Data adalah jenis kekayaan baru bangsa kita, kini data lebih berharga dari minyak. Karena itu kedaulatan data harus diwujudkan hak warga negara atas data pribadi harus dilindungi. Regulasinya harus segera disiapkan tidak boleh ada kompromi. Sekali lagi, inti dari regulasi adalah melindungi kepentingan rakyat, serta melindungi kepentingan bangsa dan negara*"* (pidato kenegaraan Presiden RI, 2019).

Upaya menumbuhkan kesadaran akan keamanan data dan informasi adalah merupakan suatu hal yang sangat mendesak dan sangat penting oleh semua pihak, apalagi oleh apparat penegak hukum. Hal tersebut sebagai upaya pencegahan terhadap berbagai macam bentuk kejahatan siber, secara khusus ialah pencurian data dan manipulasi data. Secara khusus upaya yang dilakukan pemerintah telah mengeluarkan Undang-undang No. 27 tahun 2022 tentang Perlindungan Data Pribadi. Pemrosesan data pribadi dilakukan dengan melindungi keamanan data pribadi dari pengaksesan yang tidak sah, pengungkapan yang tidak sah, pengubahan yang tidak sah, penyalahgunaan, pengrusakan dan atau penghilangan data pribadi (UU No. 27/2022).

Disisi lain, secara teknologi dapat dilakukan upaya teknis agar data dan informasi digital yang tersimpan bisa aman ialah dilakukan dengan penyandian pesan atau kriptografi dan penyembunyian pesan atau steganografi. Kriptografi menekankan pada penyandian isi pesan, sedangkan steganografi menyembunyikan keberadaan pesan (Chanu et al., 2012). Sehingga data dan informasi penting atau rahasia hanya bisa diakses oleh pihak yang berhak saja dan aman dari upaya pencurian dan manipulasi data pihak-pihak yang tidak bertanggung jawab. Hal tersebut sejalan dengan konsep dasar pembangunan sistem informasi dalam buku *The Basics of Information Security* (Andress, 2011) yang harus minimal memenuhi tiga (3) faktor yaitu kerahasian, integrasi dan ketersedian (*Confidentiality, Integrity, Availability* – CIA).

Terdapat beberapa teknik steganografi yang digunakan dalam penyembunyian pesan rahasia ke dalam citra digital, yaitu spasial domain dan frekuensi domain. Pada teknik spasial domain, pesan rahasia disembunyikan pada domain spasial, dimana pesan rahasia disembunyikan ke dalam piksel dari *cover image* dan hanya mengalami sedikit perubahan nilai piksel. Pada teknik frekuensi domain, pesan rahasia disembunyikan pada domain frekuensi, dimana pesan rahasia disembunyikan dalam koefisien-koefisien frekuensi yang didapatkan dengan melakukan transformasi pada *cover image*.

Teknik spasial domain yang umum digunakan adalah *least significant bit* (LSB). Susunan bit dalam satu *byte* terdapat 2 jenis bit, yaitu *most significant bit* (MSB) dan *least significant bit* (LSB). Satu *byte* terdapat delapan (8) bit, dengan empat (4) bit pertama merupakan MSB dan empat (4) bit terakhir merupakan LSB. Jika bit pada MSB diganti dapat menyebabkan perubahan *byte* yang cukup besar. Sedangkan, pergantian bit pada LSB tidak begitu

berpengaruh. Pada metode LSB, penyembunyian pesan dilakukan dengan menyembunyikan pesan pada bit terendah pada media penyembunyian, dalam hal ini citra digital. Pesan rahasia diubah ke dalam bentuk biner, kemudian disisipkan ke dalam citra digital dengan metode LSB (Ashok, 2010). Beberapa peneliti yang melakukan steganografi dengan Teknik spasial domain di antaranya (Kadam, et.al., 2012), (Nehru, et.al., 2012), (Kamdar, et al., 2013), (Wiryawan, et.al., 2019), dan (Fateh, et.al., 2021).

Beberapa teknik frekuensi domain yaitu *fast Fourier transform* (FFT), *wavelet transform, discrete wavelet transform* (DWT), dan *integer wavelet transform* (IWT). IWT merupakan pengembangan dari *discrete wavelet transform* (DWT). Pada teknik DWT, koefisien wavelet yang dihasilkan berbentuk *floating point* sehingga dapat menyebabkan masalah untuk menyembunyikan data yang besar sehingga diterapkan skema *lifting* pada DWT untuk mendapatkan nilai *integer* menghasilkan IWT (Jayasudha, 2013). Selanjutnya, pesan rahasia akan diubah ke dalam bit dan akan disisipkan menggunakan 1-LSB (Ashok, 2010). Peneliti lainnya yang menggunakan teknik frekuensi domain diantaranya (Chen, et.al., 2006), (Sakkara, et.al., 2012), (Bassam, et.al., 2013), (Kumar, et.al., 2013), (Atawneh, et.al., 2015), (Singh, et.al. 2015), (Kumar, et.al., 2016), (Kasana, et.al., 2017), dan (Jamel, 2020).

Dalam paper ini dibahas tentang perancangan dan implementasi algoritma steganografi (penyembunyian) pesan rahasia digital (*secret text*) ke dalam media penyembunyian berupa citra digital (*cover image*), menggunakan kombinasi teknik spasial domain dan teknik frekuensi domain. Adapun teknik frekuensi domain yang digunakan ialah IWT-Haar dan Teknik spasial domain yang digunakan ialah *least significant bit* (LSB). Hal tersebut dilakukan sebagai upaya meningkatkan keamanan pesan rahasia (*secret text*).

Tinjauan Literatur

Terkait dengan data digital yang diugunakan sebagai cover ialah beruta data citra digital, pertama dijelaskan tentang konsep data citra digital.

Suatu citra digital dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$ dimana x dan y merupakan koordinat spasial dan nilai f dengan koordinat (x, y) disebut intensitas atau tingkat keabuan citra pada titik tersebut. Ketika x, y, dan nilai f adalah berhingga, bernilai diskrit, sebuah citra dapat dikatakan sebagai citra digital. Citra digital tersusun dari berhingga elemen yang masing-masing mempunyai lokasi dan nilai tertentu. Elemen tersebut disebut picture elements, image elements, pels, dan pixel. Pixel adalah istilah yang paling sering digunakan untuk mendefinisikan elemen dari citra digital (Gonzalez & Woods, 2008). Misalkan terdapat citra digital dengan dimensi $M \times N$, maka representasi matriks dari citra digital sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix}$$

Dari representasi matriks citra digital di atas, M merupakan banyaknya baris matriks atau tinggi citra dan N merupakan banyaknya kolom matriks atau lebar citra. Semua elemen matriks tersebut dinamakan dengan nilai intensitas (Gonzalez & Woods, 2008).

Citra *grayscale* terbentuk dari kumpulan piksel yang tiap pikselnya terdiri dari nilai tunggal yang bersesuaian dengan tingkat keabuan citra di lokasi tertentu. Dalam citra *grayscale*, masing- masing warna akan disimpan dalam 1 byte (8 bit) memori pada tiap pikselnya. Artinya, tingkat keabuan warna dari tiap piksel direpresentasikan ke dalam bentuk biner sepanjang 8 bit.

Pada citra 8 bit terdapat 256 tingkat keabuan. Citra ini memiliki warna hitam sebagai warna minimal yang direpresentasikan sebagai nilai 0 dan warna putih sebagai warna maksimal yang direperesentasikan sebagai nilai 255, sedangkan warna abu-abu terletak di antara warna hitam dan putih. Secara kasat mata, sulit untuk membedakan tingkat keabuan dengan tepat (Sachs, 1996) .

Selanjutnya akan dijelaskan tentang teknik steganografi, yang merupakan hal utama dari paper ini. Teknik steganografi digunakan untuk mengamankan pesan agar pesan tersebut tidak dapat diketahui oleh pihak ketiga. Properti yang digunakan dalam steganografi menurut Thiyagarajan, (2012) adalah:

a. Secret message: pesan yang disembunyikan
b. Cover image: citra yang digunakan untuk menyembunyikan pesan rahasia
c. Stego image: cover image yang mengandung pesan rahasia

Jenis file cover yang dapat digunakan pada steganografi untuk menyisipkan secret message di antaranya teks, citra, audio, atau video. Tipe steganografi ada tiga menurut Ashok, (2010):

a. Pure Steganography Pure Steganography merupakan sistem steganografi yang tidak memerlukan pertukaran kunci rahasia (stego-key). Pengirim dan penerima pesan mengetahui akan keberadaan secret message sehingga tidak diperlukan adanya stego-key.
b. Secret Key Steganography Secret Key Steganography merupakan sistem steganografi yang memerlukan stego-key dalam pertukaran pesan. Secret message disisipkan ke dalam file cover menggunakan stego-key. Stego-key hanya diketahui pengirim dan penerima pesan.
c. Public Key Steganography Public Key Steganography mengambil konsep dari Public Key Cryptography. Dalam menjaga kerahasiaan pada pertukaran pesan diperlukan public key dan private key. Pengirim pesan menggunakan public key untuk menyisipkan pesan rahasia dan penerima pesan menggunakan private key untuk memperoleh pesan rahasia.

Pada paper ini steganografi yang digunakan adalah Pure Steganography. Pure Steganography dipilih karena implementasinya lebih banyak dibandingkan dengan Secret Key Steganography dan Public Key Steganography.

Ada dua proses steganografi, yaitu embedding dan extracting. Proses embedding adalah proses penyisipan pesan rahasia ke dalam cover image. Pada proses ini dibutuhkan dua input, yaitu cover image dan secret message. Penyisipan secret message ke dalam cover image menghasilkan stego image. Proses extracting adalah proses pengambilan pesan rahasia dari stego image. Pada proses ini dibutuhkan satu input, yaitu stego image. Setelah proses extracting selesai akan dihasilkan secret message.

Menurut Dinca (2011) steganografi dikatakan baik jika memenuhi kriteria sebagai berikut:

a. Security: kesulitan dalam mendeteksi pesan tersembunyi pada stego image
b. Capacity: kapasitas penyembunyian secret message pada cover image harus cukup
c. Robustness: kesulitan dalam menghapus pesan tersembunyi pada stego image

Transformasi wavelet umumnya memiliki koefisien transformasi berbentuk floating point. Ketika data yang dimasukkan merupakan integer, seperti file citra, maka hasil output tidak lagi berupa integer. Ini menunjukkan bahwa rekonstruksi suatu citra asli yang sempurna tidak dapat dilakukan. Dengan menggunakan IWT didapatkan hasil rekonstruksi yang akurat. IWT merupakan pengembangan dari DWT yang didapatkan dengan melakukan skema lifting (Jayasudha, 2013). Metode IWT yang digunakan adalah Haar IWT. Berikut ini diperlihatkan bagaimana cara menggunakan skema lifting untuk mendapatkan Haar IWT.

Konsep utama pada transformasi Haar wavelet (Haar IWT) adalah dekomposisi *averages* dan *differences*. Hal tersebut dipicu karena proses utama transformasinya terletak pada dekomposisi piksel (Calderbank et al., 1996).

Misal, diberikan piksel dalam dimensi 1 (1D) dengan panjang piksel $2n$ sebagai berikut:

x_1	y_1	x_2	y_2	x_3	y_3	...	x_n	y_n

Untuk proses dekomposisi *averages* dihitung menggunakan rumus pada persamaan (1) berikut:

$$a_i = \frac{x_i + y_i}{2}, \quad \text{untuk } i = 1,2,3,\dots,n \tag{1}$$

Sedangkan untuk proses dekomposisi *differences* dihitung menggunakan rumus sesuai persamaan (2) berikut:

$$d_i = x_i - y_i, \quad \text{untuk } i = 1,2,3,\dots,n \tag{2}$$

dengan a_i dan d_i merupakan nilai piksel dari hasil dekomposisi *averages* dan *difference*s. Susunan piksel-piksel hasil dekomposisi *averages* dan *differences* ditunjukkan sebagai berikut;

a_1	d_1	a_2	d_2	a_3	d_3	...	a_n	d_n

Untuk mendapatkan transformasi Haar IWT digunakan skema *lifting* (Calderbank et al, 1998). Adapun proses dekomposisi *averages* dan *differences* pada transformasi Haar *wavelet* dinyatakan sebagaimana persamaan (3) dan persamaan (4), yaitu:

.

$$s_{1,l} = \frac{s_{0,2l} + s_{0,2l+1}}{2} \tag{3}$$

$$d_{1,l} = s_{0,2l+1} - s_{0,2l} \tag{4}$$

$s_{0,2l}$ merupakan nilai pada elemen genap pada suatu bidang dan $s_{0,2l+1}$ merupakan nilai pada elemen ganjil pada suatu bidang. Sedangkan $s_{1,l}$ dan $d_{1,l}$ merupakan dekomposisi *averages* dan *differences* yang diletakkan di bidang yang baru. Adapun persamaan invers dari

persamaan (3) dan persamaan (4) adalah sebagaimana persamaan (5) dan persamaan (6) berikut:

$$s_{0,2l} = s_{1,l} + \frac{d_{1,l}}{2} \tag{5}$$

$$s_{0,2l+1} = s_{1,l} - \frac{d_{1,l}}{2} \tag{6}$$

Dapat dilihat pada persamaan (3) terdapat pembagian dengan dua sehingga transformasi tidak selalu menghasilkan nilai integer. Berikut ditunjukkan bahwa nilai yang dibuang untuk menghasilkan nilai integer tidak berpengaruh. Gunakan $s_{0,2l}$ dan $s_{0,2l+1}$ pada persamaan (3) dan persamaan (4). Sehingga $s_{1,l}$ pada persamaan (3) dibuat menjadi integer dalam bentuk persamaan (7) sedangkan nila $d_{1,l}$ tetap sebagaimana persamaan (4).

$$s_{1,l} = \left\lfloor \frac{s_{0,2l}+s_{0,2l+1}}{2} \right\rfloor \tag{7}$$

Pembulatan pada persamaan (7) menggunakan fungsi *floor* yang berarti menghilangkan bagian pecahannya untuk mendapatkan bagian integer. Perhatikan bahwa penjumlahan dan pengurangan dari dua bilangan integer meghasilkan bilangan genap keduanya atau bilangan ganjil keduanya. Sehingga didapatkan invers dari persamaan (7) dan persamaan (4) sebagai berikut:

$$s_{0,2l} = s_{1,l} - \left\lfloor \frac{d_{1,l}}{2} \right\rfloor \tag{8}$$

$$s_{0,2l+1} = s_{1,l} + \left\lfloor \frac{d_{1,l+1}}{2} \right\rfloor \tag{9}$$

Persamaan (3) dan (4) dapat diubah untuk mendapatkan hasil integer dengan menulis ulang persamaan dalam dua langkah yang dilakukan secara sekuensial. Langkah pertama, hitung *differences*, lalu gunakan *differences* di langkah kedua untuk menghitung *averages*. Berikut persamaan *differences* dan *averages* yang baru:

$$d_{1,l} = s_{0,2l+1} - s_{0,2l} \tag{10}$$

$$s_{1,l} = s_{0,2l} + \frac{d_{1,l}}{2} \tag{11}$$

Persamaan (3) sama dengan persamaan (11), hal tersebut dapat ditunjukkan dengan cara persamaan (10) disubstitusikan ke persamaan (11), sehingga diperoleh:

$$s_{1,l} = s_{0,2l} + \frac{d_{1,l}}{2} \tag{12}$$

$$= s_{0,2l} + \frac{s_{0,2l+1}-s_{0,2l}}{2} \tag{13}$$

$$= s_{0,2l} + \frac{s_{0,2l+1}}{2} - \frac{s_{0,2l}}{2} \tag{14}$$

$$= \frac{s_{0,2l}}{2} + \frac{s_{0,2l+1}}{2} \tag{15}$$

$$s_{1,l} = \frac{s_{0,2l} + s_{0,2l+1}}{2}$$

(16)

Jadi terbukti bahwa persamaan (3) sama dengan persamaan (11).

Selanjutnya akan dicari invers dari persamaan (17) dan (18). Pertama diperoleh sampel genap dari *averages* dan *differences*. Lalu, diperoleh sampel ganjil menggunakan sampel genap yang telah diperoleh sebelumnya serta *differences*. Didapatkan persamaan invers sebagai berikut:

$$s_{0,2l} = s_{1,l} - \frac{d_{1,l}}{2}$$

(17)

$$s_{0,2l+1} = d_{1,l} + s_{0,2l}$$

(18)

Persamaan (17) dan (18) dapat diubah sehingga menghasilkan nilai integer dengan menghilangkan pembagian dalam dua langkah:

$$d_{1,l} = s_{0,2l+1} - s_{0,2l}$$

(19)

$$s_{1,l} = s_{0,2l} + \left\lfloor \frac{d_{1,l}}{2} \right\rfloor$$

(20)

Persamaan (22) sama dengan persamaan (13) karena

$$s_{0,2l} + \left\lfloor \frac{d_{1,l}}{2} \right\rfloor = s_{0,2l} + \left\lfloor \frac{s_{0,2l+1}}{2} - \frac{s_{0,2l}}{2} \right\rfloor = \left\lfloor \frac{s_{0,2l}}{2} + \frac{s_{0,2l+1}}{2} \right\rfloor.$$

(21)

Dengan skema lifting didapatkan persamaan invers sebagai berikut:

$$s_{0,2l} = s_{1,l} + \left\lfloor \frac{d_{1,l}}{2} \right\rfloor$$

(22)

$$s_{0,2l+1} = d_{1,l} + s_{0,2l}$$

(23)

Persamaan (24) sama dengan persamaan (16) dengan mensubstitusi menggunakan $d - \lfloor d\ 2 \rfloor = \lfloor d{+}1\ 2 \rfloor$ pada persamaan (16).

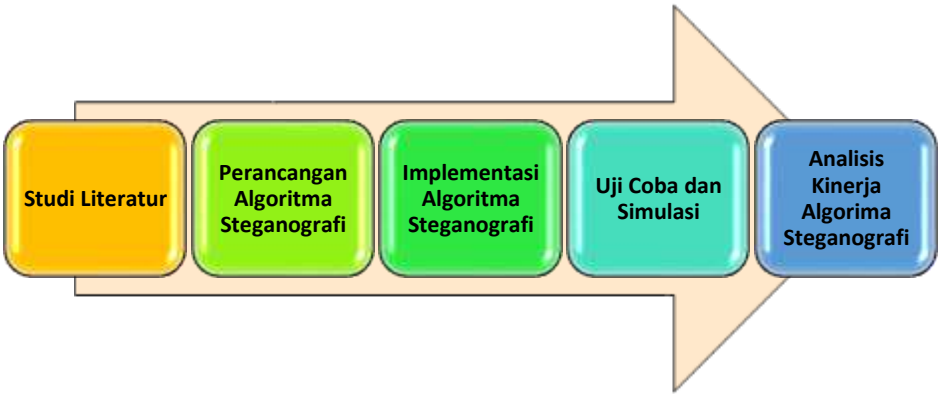
Persamaan (21) dan (22) merupakan bentuk transformasi yang menghasilkan nilai integer menggunakan skema lifting dan persamaan (23) dan (24) merupakan bentuk inversnya.. Transformasi ini biasa disebut Haar Integer Wavelet Transform atau IWT–Haar. Maka, bentuk dekomposisi *averages* dan *differences* pada persamaan (7) dan (8) yang baru, yaitu:

$$d_i = x_i - y_i \text{ dengan } i = 1, 2, 3, \dots, n \quad (24)$$

$$a_i = y_i + \left\lfloor \frac{d_1}{2} \right\rfloor \text{ dengan } i = 1, 2, 3, \dots, n \quad (25)$$

Metode Penelitian

Sejalan dengan tujuan yang ingin dicapai upaya meningkatkan keamanan pesan rahasia (*secret text*), maka metode penelitian yang digunakan ialah metode kuantitatif dengan eksplorasi terhadap data uji yang digunakan pada suatu racangan algoritma steganografi berbasis transformasi IWT-Haar dan LSB. Adapun tahapan penelitian yang dilakukan sebagaimana tampak pada Gambar 1.



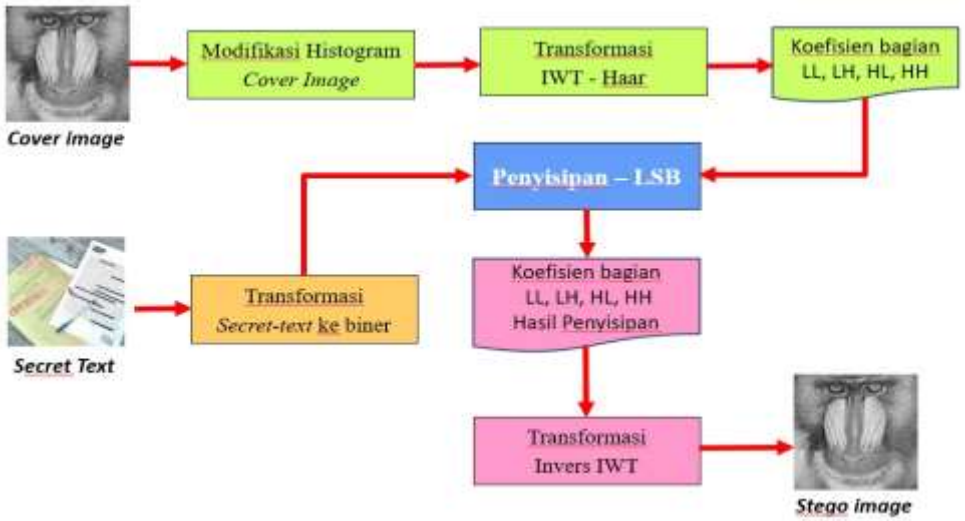
Gambar 1 Tahapan Penelitian Pengamanan Data Digital dengan Metode Steganografi

Tahapan awal dari penelitian yang dilakukan yakni melakukan studi literatur. Hal tersebut sudah dijelaskan pada bagian tinjauan literatur. Berdasarkan tinjauan tersebut dari berbagai macam teknik atau metode steganografi yang ada dan berkembang, maka ditetapkan bahwa dalam penelitian ini dilakukan upaya pengamanan data digital dengan menggunakan metode steganografi dengan kombinasi dua (2) metode yaitu metode IWT-Haar dan metode *least significant bit* (LSB). Pendekatan metode tersebut dilakukan sebagai upaya untuk meningkatkan keamanan data atau pesan rahasia (*secret text*). Tahapan selanjutnya dilakukan perancangan algoritma steganografi, yang dijelaskan pada bagian berikut.

Perancangan Algoritma Steganografi.

Algoritma steganografi terdiri dari 2 proses yaitu proses *embedding* dan proses *extracting* dengan menggunakan IWT-Haar dan LSB. Proses *embedding* merupakan proses untuk mengamankan data digital rahasia dengan cara disembunyikan pada data lain yang berupa data teks, data gambar (*image*), data suara atau data video. Pada paper ini, data digital yang diamankan berupa data teks dan disembunyikan pada data gambar (*image*). Data digital rahasia berupa teks yang akan diamankan disebut dengan isttilah *secret text* atau plaintext dan file data digital berupa gambar (*image*) yang akan menjadi tempat penyembunyian *secret text* disebut dengan istilah *cover image*. Adapun metode yang digunakan pada paper ini adalah menggunakan transformasi IWT-Haar yang dikombinaksikan dengan teknik penyisipan LSB.

Proses *embedding* tersebut disajikan dalam bentuk diagram sebagaimana tampak pada Gambar 2.



Gambar 2 Proses *Embedding* dengan Transformasi IWT-Haar dan LSB

Gambar 2, menunjukkan proses *embedding* yakni proses untuk menyembunyikan file *secret text* ke dalam *cover image*. Kedua file tersebut sebagai input dari proses *embedding*. File *Cover image* ditransformasikan dengan IWT-Haar sehingga menghasilkan 4 bagian koefisien, yaitu LL, LH, HL dan HH. Hal tersebut dilakukan setelah proses modifikasi histogram dari *cover image*. Sedangkan file *secret text* ditransformasikan ke dalam bentuk biner. Selanjutnya file *secret text* dalam bentuk biner disisipkan dengan metode LSB ke dalam koefisien dari 4 bagian pada *cover image*. Metode LSB tersebut dapat mensubstitusi semua bit pada bagian LSB dengan bit pesan rahasia dan nilai susunan bit tetap tidak berubah secara signifikan dibandingkan dengan subtitusi pada bagian MSB. Jika metode 1-LSB (1-bit paling kanan dari LSB) dan 4-LSB (4-bit paling kanan dari LSB) dibandingkan, maka metode 1-LSB mempunyai hasil yang lebih bagus dibanding 4-LSB, hanya saja kapasitasnya pesan yang dapat disubstitusi menjadi lebih sedikit karena banyak bit LSB yang dapat disubstitusi dengan bit pesan rahasia menjadi lebih sedikit. Pada paper ini digunakan metode 1-LSB dengan bit pesan rahasia disisipkan pada 1 bit paling kanan dari LSB dengan cara mensubtitusi 1 bit LSB tersebut dengan 1 bit pesan rahasia.

Hasil penyisipan tersebut kemudian ditransformasikan dengan invers IWT sehingga menghasilkan *stego image*. *Stego image* merupakan file *cover image* yang sudah disisipkan *secret text*, yang secara kasat mata (secara kualitatif) *stego image* sangat mirip atau sama persis dengan *cover image*. Hal tersebut dapat dibuktikan pula dengan pengujian secara kuantitatif menggunakan ukuran *mean square error* (MSE) dan *peak signal noise ratio* (PSNR). Sehingga pesan rahasia (*secret text*) tersebut sudah terlindugi dengan cara disembunyikan pada file *cover image* berupa *stego image*.

Proses tersebut dapat disajikan dalam bentuk algoritma dengan notasi pseudocode sebagai berikut:

Algoritma Embedding.

1. Input file citra digital, sebagai cover image

2. Input file text digital sebagai pesan rahasia (secret text)
3. Ubah pesan rahasia ke dalam bentuk biner
4. Lakukan modifikasi histogram pada cover image
5. Lakukan transformasi IWT Haar pada cover image menjadi koefisien LL, LH, HL, dan HH.
6. Periksa nilai koefisien LL, LH, HL, dan HH untuk melakukan penyisipan pesan rahasia ke dalam koefisien tersebut.
7. Lakukan transformasi balik (*Inverse Integer Wavelet Transform* – invers IWT) pada koefisien LL, LH, HL, dan HH hasil penyisipan dan nyatakan stego image

Langkah selanjutnya adalah melakukan proses proses *extrackting*, sebagai upaya yang dilakukan oleh pihak yang berwenang mendapatkan informasi dari pesan rahasia (*secret text*) tersebut. Proses yang dilakukan ialah mengekstraksi file *stego image* sedemikian sehingga diperoleh pesan rahasia (*secret text*) yang disembunyikan pada file tersebut. Prosedur tersebut tergambarkan dengan jelas pada Gambar 3.

Gambar 3. Proses *Extracting* dengan Transformasi IWT-Haar dan LSB

Gambar 3 menunjukkan bahwa input dari proses tersebut berupa file *stego image* yang akan diekstrak untuk dapat diperoleh pesan rahasia (*secret text*) atau pesan aslinya. Langka yang dilakukan adalah dengan melakukan transformasi IWT-Haar terhadap file *stego image* sehingga diperoleh keofisien dari empat (4) bagian yaitu LL, LH, HL dan HH. Selanjutnya setiap masing-masing keofisien tersebut ditransformasi menggunakan struktur logika AND terhadap 1, untuk mendapatkan bit *secret text*. Kemudian hasil tersebut direkonstruksi untuk memperoleh pesan rahasia atau pesan asli yang dimaksud, sebagai output dari proses *extracting*.

Proses tersebut dapat disajikan dalam bentuk algoritma dengan notasi pseudocode sebagai berikut:

Algoritma Extracting.

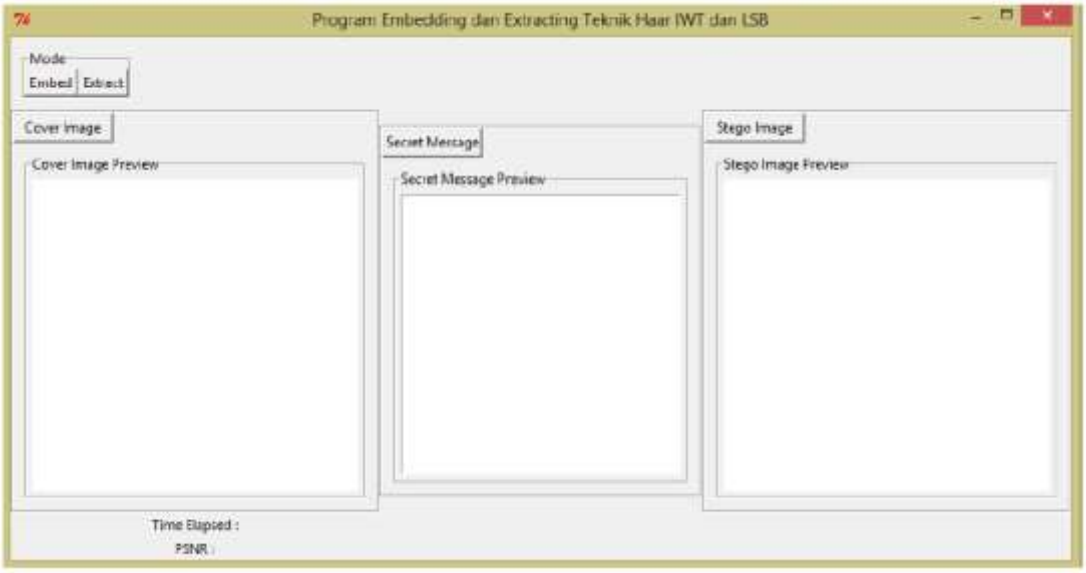
1. Input file stego image
2. Lakukan transformasi *Integer Wavelet Transform* – Haar (IWT – Haar)

3. Koefisien LL, LH, HL, dan HH hasil transformasi IWT – Haar ditransformasikan dengan logic AND terhadi 1.
4. Diperoleh bit secret text dari setiap koefisien LL, LH, HL, dan HH.
5. Rekonstruksi bit secret text menjadi secret text

Setelah rancangan algoritma steganografi berhasil dikembangkan, maka dilanjutkan dengan tahapan implementasinya guna memperoleh program aplikasi steganografi. Proses implementasinya dijelaskan pada bagian berikut.

Implementasi Program Aplikasi Steganografi

Berdasarkan rancangan algoritma steganografi yang telah dijelaskan sebelumnya, maka dibuatlah program aplikasinya menggunakan bahasa pemerograman *open source* yaitu bahasa pemrograman Python versi 2.7.2 dan *graphical user Interface* (GUI) menggunakan modul Tkinter. Fitur yang berhasil dbuat pada program aplikasi tersebut yaitu menu utama, submenu proses embed dan submenu proses extract. Adapun tampilan layar dari program aplikasi tersebut tampak pada Gambar 4 sampai dengan Gambar 8.



Gambar 4 Tampilan menu utama program implementasi algoritma Haar IWT dan LSB

Pada Gambar 4, tampak ada tombol *Embed* dan *Extract* yang dikelompokan pada pilihan mode proses dari program aplikasi ini. Tampilan awal, secara defaultnya adalah isi dari *cover image*, *secret message* dan *stego image* masih kosong. Jika ingin dilakukan proses *embedding*, maka dilakukan langkah menekan tombol *cover image* untuk memasukan file gambar (*image*) yang diinginkan sebagai *cover image*. Selanjutnya ditekan tombol *secret message* untuk memasukan file pesan rahasia berupa file text. Tampilan layarnya sebagaimana tampak pada Gambar 5.

Gambar 5. Tampilan Layar Hasil Input Data Proses *Embedding*

Selanjutnya dilakukan proses *embedding* dengan cara menekan tombol embed, dan program aplikasinya akan memprosesnya dan menampilkan hasilnya berupa file *stego image* yang tampak pada Gambar 6 bagian sebelah kanan.

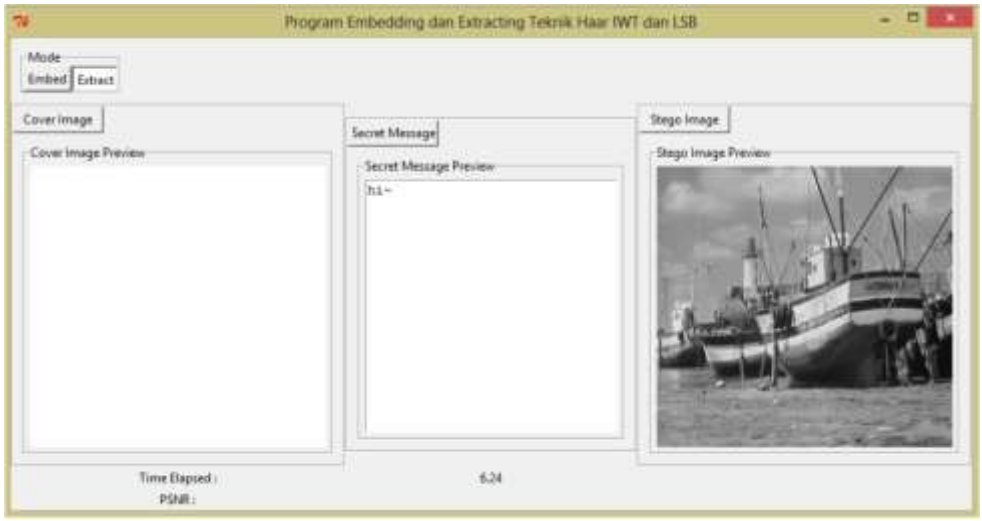
Gambar 6. Tampilan Layar Hasil Proses *Embedding*

Tampak pada Gambar 6, tampilan gambar *cover image* (di sebelah kiri) dan tulisan *secret message* (ditengah) serta *stego image* (di sebelah kanan). File *stego image* merupakan file yang memuat informasi gambar asal (*cover image*) dan juga *secret message* (text). Gambar 6, juga memuat informasi waktu yang butuhkan dalam proses *embedding* dan iformasi nilai PSNR, yang menunjukkan tingkat kemiripan gambar antara *cover image* dengan *stego image*.

Fitur lainnya dari program aplikasi steganografi yang dikembangkan pada paper ini adalah proses *extracting*. Untuk melakukan proses *extracting*, berdasarkan tampilan pada Gambar 7, dilakukan input data berupa *stego image* dengan menekan tombol *stego image* (lihat Gambar 7).

Gambar 7. Tampilan Layar Input *Stego image* untuk Proses *Extracting*

Berdasarkan Gambar 8 tersebut,tampak bahwa file data berupa *stego image* yang dipilih untuk diekstrak terlihat atau muncul pada layer bagian kanan. Selanjutnya untuk melakukan proses ekstraksi dilakukan dengan cara menekan tombol extract,. Sehingga hasil prosesnya tersebut muncul sebagaimana tampilan layarnya pada Gambar 8. Hasilnya yaitu berupa pesan teks digital, yang pada Gambar 8 tampak bagian tengah dibagian *secret message*.



Gambar 8. Tampilan Layar Hasil Proses Ekstraksi

Setelah berhasil dikembangkan program aplikasi steganografi tersebut, maka dilanjutkan dengan tahapan berikutnya ialah tahap uji coba dan simulasi dengan menggunakan data uji. Adapun penjelasannya dilakukan pada bagian berikut ini.

Data Uji Coba dan Simulasi

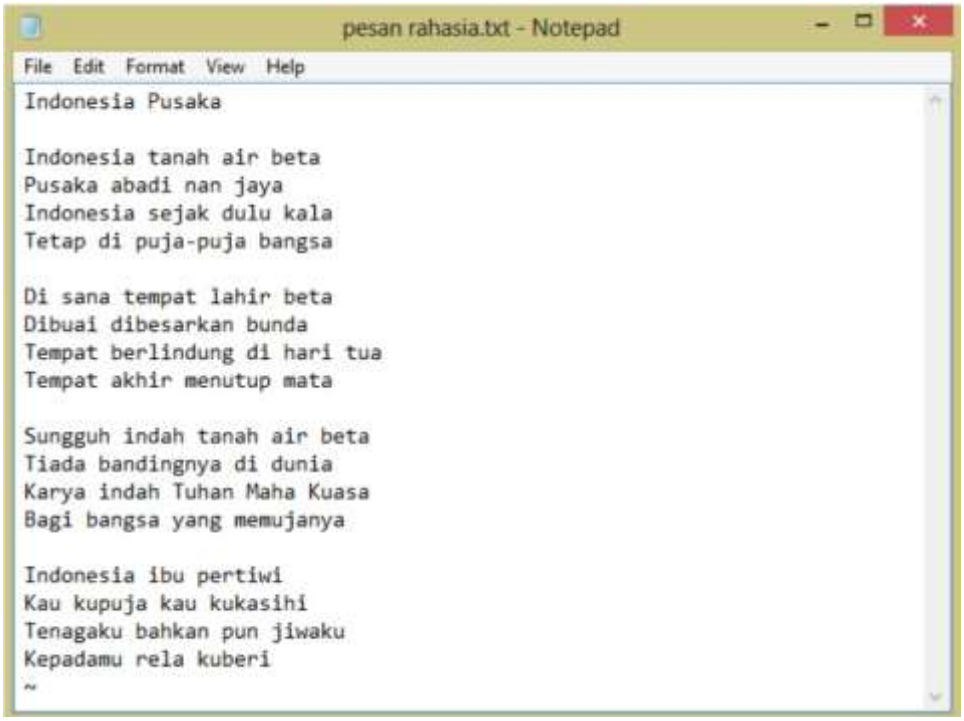
Pelaksanaan uji coba dan simulasi dari program aplikasi steganografi dilakukan menggunakan data uji berupa data teks sebagai *secret text* atau *plaintext* data berupa gambar (*image*) sebagai *cover image*. Adapun data uji gambar yang digunakan adalah berupa file

gambar digital *grayscale* dalam berbagai ukuran sebagai *cover image* yang disajikan pada Tabel 1.

Tabel 1. Data Uji Citra Digital Grayscale

Data Uji ke-	Nama Citra	Tampilan Citra	Ukuran Citra (piksel)
1	flowers.png		256×192
2			512×384
3			768×576
4			1000×750
5	boat.png		512×512
6			650×650
7			800×800
8			900×900

Adapun data pesan rahasia yang digunakan adalah file lirik lagu Indonesia Pusaka dengan format .txt yang diperlihatkan pada Gambar 9.



Gambar 9. Tampilan file Pesan Rahasia (*Secret text*)

Tampak bahwa dari hasil uji coba dan simulasi bahwa semua fungsi dari proses embedding dan extracting berjalan sesuai dengan konsep dan rancangan algoritma yang dikembangkan. Hal tersebut berjalan dengan baik berdasarkan semua data uji yang digunakan. Tahapan selanjutnya dilakukan analisis hasil berdasarkan waktu proses, dan nilai PSNR yang dijelaskan pada bagian berikut.

Hasil dan Pembahasan

Semua data uji pada Tabel 1 dan Gambar 9 digunakan dalam proses *embedding* dan *extracting* untuk menghitung waktu yang dibutuhkan selama proses *embedding* dan *extracting*. Selain itu dilakukan juga pengujian kualitas citra dari *stego image* dibandingkan dengan *cover image*.

Analisis Waktu Proses

Simulasi program steganografi baik pada proses *embedding* maupun *extracting* dilakukan dengan menggunakan semua data uji sebagaimana yang ada pada Tabel 1 dan Gambar 9. Setiap data uji yang digunakan, dengan masing-masing proses dijalankan sebanyak 5 kali, kemudian dihitung rata-rata waktu untuk setiap proses *embedding* dan *extracting*. Hasil rata-rata waktu proses *embedding* dan *extracting* yang diperoleh dari simulasi tersebut tampak pada Tabel 2.

Tabel 2. Rata-rata Waktu Proses *Embedding* dan *Extracting*

Data Uji ke-	Ukuran Citra (piksel)	Rata-rata waktu *embedding* (detik)	Rata-rata waktu *extracting* (detik)
1	256×192	1,82	0,60
2	512×384	7,06	2,29
3	768×576	15,91	5,11
4	1000×750	26,73	8,63
5	512×512	9,01	2,93
6	650×650	14,35	4,71
7	800×800	21,80	7,08
8	900×900	28,22	9,23

Berdasarkan Tabel 2, tampak bahwa rata-rata waktu *extracting* jauh lebih cepat dibandingkan dengan rata-rata waktu *embedding*. Pada data uji ke-1, tampak bahwa rata-rata waktu *extracting* sekitar 1/3 kali dari rata-rata waktu *embedding*. Demikian pula pada data uji ke-2 sampai dengan data uji ke-8, menunjukkan bahwa rata-rata waktu extracting sekitar 1/3 kali dari rata-rata waktu embedding. Hal tersebut terjadi karena dalam proses *extracting* hanya berorientasi semata-mata untuk mendapatkan pesan rahasia (*secret text*) saja. Sehingga tidak dilakukan proses mengembalikan koefisen wavelet menjadi bentuk citra. Dapat dilihat juga bahwa pada proses *embedding* dan proses *extracting*, dengan pesan rahasia yang sama, menunjukkan semakin besar piksel yang digunakan membutuhkan waktu proses yang lebih lama.

Analisis PSNR

Untuk mengetahui kualitas *stego image* yang dihasilkan, apakah sama persis atau mirip dengan *cover image*nya atau malah jauh berbeda. Sehingga perlu dilakukan pengujian secara kuantitatif. Pengujiannya dengan mengukur atau menghitung nilai *peak signal to noise ratio* (PSNR). Adapun rumus perhitungannya menggunakan persamaan (1) (Cheddad et al, 2013):

$$PSNR = 10 \log\left(\frac{255^2}{MSE}\right) \quad \ldots (1)$$

dengan:

$$MSE = mean\ square\ error = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j)-F(i,j))^2}{M \times N}$$

$M = panjang\ citra$

$N = lebar\ citra$

$f(i,j) = elemen\ piksel\ cover\ image$

$F(i,j) = elemen\ piksel\ stego\ image$

MSE merupakan nilai *error* rata-rata kuadrat antara *cover image* dengan *stego image*. Semakin besar nilai MSE maka semakin besar pula perbedaan *stego image* dengan *cover image*. Hal tersebut mengakibatkan nilai PSNR semakin kecil. Kualitas *stego image* dikatakan baik apabila memiliki nilai di atas 40 dB (Cheddad et al 2010). Sehingga semakin kecil nilai MSE (mendekati atau sama dengan nol) maka nilai PSNR semakin besar, maka semakin baik pula kualitas *stego image* yang dihasilkan.

Berdasarkan hasil simulasi menggunakan data uji, dilakukanlah perhitungan nilai MSE dan PSNR dari file *stego image* terhadap *cover image*, yang hasilnya tampak pada Tabel 3.

Tabel 3. Nilai MSE dan PSNR dari *stego image* terhadap *cover image*

Data Uji ke-	Ukuran Citra (piksel)	Nilai MSE	Nilai PSNR (dB)
1	256×192	0,071	59,595
2	512×384	0,011	67,453
3	768×576	0,007	69,180
4	1000×750	0,004	71,500
5	512×512	0,199	55,135
6	650×650	0,147	56,448
7	800×800	0,158	56,137
8	900×900	0,160	56,083

Berdasarkan Tabel 3, diperoleh bahwa hubungan nilai MSE dengan nilai PSNR adalah berbanding terbalik. Nilai MSE menunjukkan nilai rata-rata kuadrat dari selisih antara piksel citra stego dengan piksel citra cover. Jika nilai MSE nya mendekati nol hal tersebut berarti nilai piksel yang bersesuaian antara citra stego dan citra cover itu hamper sama. Berdasarkan semua data uji yang digunakan tampak bahwa nilai MSE nya semuanya mendekati nol. Dengan demikian diperoleh nilai PSNR nya akan semakin besar jauh di atas 40 dB. Ini menunjukkan bahwa kualitas *stego image* yang dihasilkan sangat baik. Sehingga *stego image* dan *cover image* sulit dibedakan (tidak ada perbedaan yang signifikan) secara kasat mata.

Kesimpulan dan Rekomendasi

Berdasarkan hasil simulasi dengan menggunakan data uji dan analisisnya maka diperoleh kesimpulan sebagai berikut:

a. Rata-rata waktu pada proses *extracting* jauh lebih cepat dibandingkan waktu proses *embedding*. Hal ini terjadi karena pada bagian proses *embedding* yaitu setelah proses penyisipan dilanjutkan dengan proses pengubahan koefisien IWT-Haar menjadi bentuk citra. Sedangkan proses pengubahan keofisien IWT-Haar tersebut tidak dilakukan pada proses *extracting*.

b. Perbedaan frekuensi intensitas warna mempengaruhi nilai PSNR. Pada *cover image* yang tidak mengalami modifikasi histogram, semakin besar ukuran piksel gambar, maka semakin besar pula nilai PSNR. Pada *cover image* yang mengalami modifikasi histogram, semakin banyak piksel dengan nilai intensitas di bawah 15 dan di atas 240, nilai PSNR akan semakin kecil.
c. Nilai PSNR antara *stego image* dan *cover image* jauh di atas 40 dB yakni berkisar antara 55-71,5 dB. Sehingga *stego image* dikategorikan kualitasnya sangat baik dan sulit dibedakan secara kasat mata dengan *cover image* (tidak ada perbedaan yang signifikan). Dengan demikian algoritma steganografi yang dikembangkan ini memiliki tingkat keamanan yang sangat baik dalam melindungi pesan rahasia berupa teks digital.

Selanjutnya rekomendasi yang ditawarkan dari hasil penelitian ini adalah :

a. Diupayakan secara sistematis dan berkesinambungan melakukan program penumbuhan kesadaran dalam keamanan data dan informasi, terkait penggunaan saluran komunikasi (*channel*) umum yang digunakan dalam pengiriman data dan informasi penting.
b. Agar data dan informasi penting tersebut dapat terjamin kerahasiaannya maka dilakukan upaya perlindungan dengan menerapkan program aplikasi steganografi yang dihasilkan dalam penelitian ini.
c. Dalam implementasi program aplikasi steganografi ini hendaknya digunakan file *cover image* yang ukurannya relatif besar agar data dan informasi yang disembunyikan (dilindungi) bisa terjaga dengan optimal.

Daftar Pustaka

Ashok, J, et al., (2010). Steganography: An Overview. *International Journal of Engineering Science and Technology*. Volume: 2, Issue: 10, Pages: 5985-5992.

Atawneh, S, et al., (2013). Steganography in Digital ImagesL Common Approaches and Tools. *IETE Technical Review*. Volume: 30, Issue: 4, Pages: 344- 358.

Atawneh, S.; Putra, S. (2015). An Overview of Frequency-based Digital Image Steganography. *International Journal of Cryptology Research*., Volime 5, Issue 2, page: 15-27.

Bassam J. Mohd, Thaier Hayajneh and Ahmad Nahar Quttoum, (2013). Wavelet-transform steganography: algorithm and hardware implementation., *Int. J. Electronic Security and Digital Forensics*, Vol. 5, Nos. ¾, page: 241-256.

Calderbank, A.R., et al., (1996). Wavelet Transforms That Map Integers to Integers. *Applied and Computational Harmonic Analysis*. Volume: 5, Issue: 3, Pages: 332-369.

Cheddad, A., et al., (2010). Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing*, Volume 90, Issue 3, page: 727-752.

Chen, P.Y., Lin H.J. (2006). A DWT based approach for image steganography. *International Journal of Applied Science and Engineering*. Volume: 4, Issue: 3, Pages: 275-290.

Chun-Lin, L. (2010). *A Tutorial of the Wavelet Transform*. Taiwan: Department of Electrical Engineering National Taiwan University.

Chanu, Y. J., et al., (2012). Image Steganography and Steganalysis: A Survey. *International Journal of Computer Applications*. Volume: 52, No. 2, Pages: 1-11.

Dinca, L. M.. (2011). Survey of the Use of Steganography over the Internet. *Information Economica*. Volume: 15, No. 2, Pages: 153-164.

Fateh, Mansoor., Rezvani, Mohsen., and Irani, Yasser., (2021). A New Method of Coding for Steganography Based on LSB Matching Revisited., *Security and Communication Networks*, Volume 2021, Article ID 6610678.

Jamel, Enas Muzaffer., (2020). Image Steganography Based on Wavelet Transform and Histogram Modification., Ibn Al Haitham Journal for Pure and Applied Science, Vol. 33, Issue: 1, page: 173-186.

Jayasudha, S. (2013). Integer Wavelet Transform Based Steganographic Method Using Opa Algorithm. *International Journal of Engineering and Science*. Volume: 2, Issue: 4, Pages: 31-35.

Kadam, K., Koshti, A., & Dunghav, P. (2012). Steganography Using Least Signicant Bit Algorithm. 2(3), 338–341.

Kamdar, P. Naitik, et al., (2013). Performance Evaluation of LSB based Steganography for optimization of PSNR and MSE. *Journal of Information, 52 Universitas Indonesia Knowledge and Research in Electronics and Communication Engineering*. Volume: 02, Issue: 02, Pages: 505-509.

Kasana, G.; Singh, K.; Bhatia, S. (2017). Data Hiding Algorithm for Images Using Discrete Wavelet Transform and Arnold Transform. KIPS, *Journal of Information Processing Systems*., 13, 5, 1331-1344.

Kumar, M.N., & Srividya, S. (2013). Genetic Algorithm based Color Image Steganography using Integer Wavelet Transform and Optimal Pixel Adjusment Process. *International Journal of Innovative Technology and Exploring Engineering*. Volume: 3, Issue: 5, Pages: 60-65.

Kumar, R. (2016). Audio Steganography using QR Decomposition and Fast Fourier Transform. December 2015. https://doi.org/10.17485/ijst/2015/v8i1/69604.

Nehru, G., & Dhar, P. (2012). A Detailed look of Audio Steganography Techniques using LSB and Genetic Algorithm Approach. 9(1), 402–406.

Pidato Kenegaraan Presiden Republik Indonesia 16 Agustus 2019, Di Depan Sidang Bersama DPD RI dan DPR RI., https://aptika.kominfo.go.id/2019/08/pidato-kenegaraan-presiden-jokowi-tahun-2019/

Sakkara, S., et al., (2012). Integer Wavelet based Secret Data Hiding By Selecting Variable Bit Length. *International Journal of Computer Applications*. Volume: 48, No. 19, Pages: 7-11.

Singh, Arun Kumar., et.al., (2015). Steganography in Images Using LSB Technique., *International Journal of Latest Trends in Engineering and Technology*, Vol. 5, Issue: 1, page: 426-430.

Thiyagarajan, P. (2012). *Stego-Image Generator (SIG) – Building Steganography Image Database*. CDBRSE Lab Department of Computer Science, Pondicherry University.

Wiryawan, I Gede., Sariyasa., & Aris Gunadi, I Gede. (2019). Steganografi Berdasarkan Metode Least Significant Bit pada Citra Digital dengan Teknik Kompresi Lossless., *Jurnal Ilmu Komputer Indonesia* (JIKI), Vol : 4, No. 1, page: 34-40.