

KINERJA ROBUSTNESS PADA WATERMARKING CITRA YANG REVERSIBLE MENGGUNAKAN SKEWED HISTOGRAM SHIFTING

ROBUSTNESS PERFORMANCE ON REVERSIBLE WATERMARKING IMAGE USING SKEWED HISTOGRAM SHIFTING

Aulya Rahmawati¹, Ledy Novamizanti², Gelar Budiman³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
¹aulyarahmawati@student.telkomuniversity.ac.id, ²ledyaldn@telkomuniversity.ac.id,
³gelarbudiman@telkomuniversity.ac.id

Abstrak

Kemudahan akses gambar digital adalah salah satu dampak positif perkembangan teknologi. Adapun dampak negatif dari kemudahan tersebut gambar rentan disalahgunakan oleh pihak lain. Untuk menghindari terjadinya hal tersebut, gambar harus dilindungi oleh sistem keamanan. *Watermarking* adalah teknik yang digunakan untuk melindungi gambar dengan menyembunyikan informasi ke dalam gambar. Metode *watermarking* yang digunakan pada Tugas Akhir ini adalah *Skewed Histogram Shifting* dan *prediction error*. *Watermark* disisipkan jika *prediction error* adalah 0, dan disisipkan pada *Positive Histogram Skewed* (PHS) atau *Negative Histogram Skewed* (NHS). Penyisipan pada PHS *pixel* akan bergeser ke kanan, dan penyisipan pada NHS *pixel* akan bergeser ke kiri. Teknik *reversible* pada penelitian ini berhasil pada citra dengan kondisi tanpa *underflow* dan *overflow*, citra yang berhasil pulih memiliki PSNR rekonstruksi *infinity* dan BER 0. Citra yang tidak berhasil pulih memiliki nilai PSNR rekonstruksi lebih dari 90 dB dan BER 0. Data uji menggunakan 10 citra berukuran 512×512. Sistem ini *robust* terhadap serangan *speckle* saat LM dikompresi dengan variansi 1×10^{-3} dan 1×10^{-4} rata-rata BER yang dihasilkan 0.1404 dan 0 secara berurutan, dan serangan *salt and pepper* saat LM dikompresi dengan *density* 1×10^{-4} rata-rata BER adalah 0.1159.

Kata kunci : *Watermarking, Data Hiding, Reversible Watermarking, Watermarking Citra, Histogram Shifting.*

Abstract

Ease of access to digital images is one of the positive impacts of technological advancements. However, the negative impact is that images are vulnerable to misuse by others. To avoid this, images must be protected by security systems. Watermarking is a technique used to protect images by hiding information within the image. Watermark inserted if the prediction error is 0 and is inserted in either Positive Histogram Skewed (PHS) or Negative Histogram Skewed (NHS). Insertion on PHS pixels shift to the right, and insertion on NHS pixels shift to the left. Reversible technique used in this study successful for images without underflow and overflow, with the successfully restored image PSNR reconstruction is infinite and BER 0. Images that failed to restore had a reconstruction PSNR value more than 90 dB and BER 0. The test data used 10 images with a size of 512×512. System is robust against speckle attacks when LM compressed with a variance of 1×10^{-3} and 1×10^{-4} , with an average resulting BER of 0.1404 and 0, respectively, and salt and pepper attacks when LM compressed with a density of 1×10^{-4} , with an average resulting BER of 0.1159.

Keyword: *Watermarking, Data Hiding, Reversible Watermarking, Watermarking Image, Histogram Shifting*

1. Pendahuluan

1.1 Latar Belakang

Saat ini proses pertukaran dan akses citra sangat mudah untuk dilakukan. Dengan begitu, citra dapat diakui, dicuri, maupun dimodifikasi dengan mudah oleh siapapun. Untuk menghindari terjadinya hal yang tidak diinginkan tersebut, suatu citra harus dilindungi dengan sistem keamanan. *Watermarking* merupakan teknik yang dapat digunakan untuk melindungi suatu citra dengan cara menyisipkan suatu informasi ke dalam citra. Penggunaan teknik *watermarking* ini dilakukan untuk melindungi hak cipta dan informasi penting yang terdapat dalam citra. Dalam *watermarking*, informasi ini disebut *watermark*. *Watermark* yang disisipkan merupakan suatu tanda kepemilikan atau hak cipta citra tersebut. Dalam situasi tertentu, ada kebutuhan untuk mengakses kembali citra asli, seperti untuk keperluan *editing* atau penyebaran ulang. Untuk mengatasi hal ini, teknik *watermarking* yang *reversible* dapat menjadi pilihan yang tepat. *Reversible watermarking* memungkinkan pengekstraksian *watermark* tanpa merusak kualitas citra asli. Penggunaan teknik ini memastikan bahwa citra asli dapat dikembalikan dalam kondisi yang sama tanpa adanya penurunan kualitas [1]. Secara umum, *watermarking* pada citra harus memperhatikan nilai *pixel* citra tersebut untuk menghindari terjadinya *overflow* dan *underflow*. *Overflow* dan *underflow* dapat terjadi jika nilai *pixel* diluar rentang yang ditentukan. Untuk mengatasi masalah

tersebut, *location map* dapat digunakan untuk menentukan posisi *pixel* yang memiliki nilai diluar rentang yang ditentukan. Sehingga, *pixel* tersebut dapat dikoreksi sebelum terjadi masalah *overflow* atau *underflow*.

Pada Tugas Akhir ini, dibangun sistem *reversible watermarking* dengan metode *skewed histogram shifting* dan *prediction error*. Sistem ini dibuat dengan memindai *pixel* yang berada di dalam *border pixel* dimulai dari $2:n-2$ dan $2:m-2$. Dalam penelitian ini digunakan beberapa referensi terkait *reversible watermarking* dan metode yang digunakan. Referensi pertama berjudul *Skewed Histogram Shifting for Reversible Data Hiding using a Pair of Extreme Predictions*. Penelitian tersebut menghasilkan PSNR sebesar 60 dB [2]. Referensi kedua berjudul *Reversible Data Hiding Using Improved Gradient Based Prediction and Adaptive Histogram Bin Shifting*. Penelitian tersebut menghasilkan PSNR sebesar 41.54 dB [3]. Referensi ketiga berjudul *Border following-based reversible watermarking algorithm for images with resistance to histogram overflowing*. Penelitian yang menggunakan metode *histogram shifting* berbasis *border pixel* tersebut menghasilkan rata-rata PSNR sebesar 59.12 dB [4].

Border pixel yang digunakan dalam penelitian ini berada diluar ketentuan $2:n-2$ dan $2:m-2$. Pada penelitian [2], *border pixel* yang digunakan berada diluar ketentuan $2:n-1$ dan $2:m-1$ untuk menyimpan informasi *pre-processing*. Dengan n dan m adalah ukuran citra. Penelitian [4], menggunakan citra biner dimana *border pixel* ditentukan jika nilai *pixel* sama. *Border pixel* dapat digunakan untuk menyimpan berbagai informasi mengenai pengolahan citra. Oleh karena itu, *border pixel* dianggap memiliki kegunaan yang sangat baik dalam *watermarking* citra.

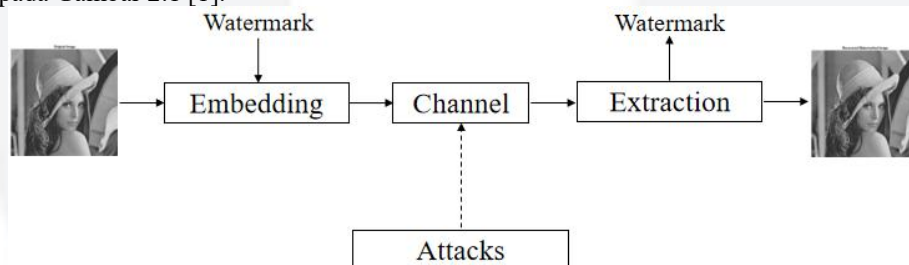
2. Konsep Dasar

2.1 Watermarking

Watermarking merupakan teknik untuk melindungi citra. Teknik melindungi citra dikelompokkan ke dalam *watermarking*, dan steganografi. Dalam *steganografi* informasi disimpan pada citra dengan tujuan agar hanya pengirim dan penerima yang mengetahui adanya informasi dalam citra, dan apa isi informasi tersebut. Sedangkan dalam *watermarking*, *watermark* disembunyikan untuk memberikan perlindungan kepada citra, untuk menghindari penggandaan, perlindungan hak cipta, *fingerprinting*, dan autentikasi citra. *Watermarking* dikarakteristikan dalam empat aspek yaitu *capacity*, *visibility*, *security*, dan *robustness*. *Capacity* merupakan banyaknya *watermark* yang dapat disisipkan pada citra. *Visibility* merupakan kemampuan untuk mendeteksi *watermark* pada citra. *Security* merupakan kemampuan melindungi *watermark* dari pihak yang tidak berwenang. *Robustness* berarti bahwa citra dapat tahan terhadap modifikasi, dan dapat dideteksi saat proses ekstraksi [5].

2.1.1 Reversible Watermarking

Teknik *reversible watermarking* digunakan agar citra yang telah melalui proses penyisipan dapat dikembalikan seperti semula. Teknik ini juga disebut *lossless watermarking* yang memungkinkan citra dapat kembali setelah proses ekstraksi tanpa adanya kerusakan. Blok diagram *reversible watermarking* secara umum dapat dilihat pada Gambar 2.1 [1].



Gambar 2.1 Reversible Watermarking.

2.1.2 Robustness

Robustness adalah sebuah kemampuan sistem untuk tetap kokoh meskipun terjadi beberapa perubahan pada data yang diolah. *Robustness* merupakan salah satu aspek penting dalam *watermarking* [5]. Suatu citra yang telah disisipkan *watermark* akan sangat baik kualitasnya jika citra tersebut *robust*. Jika citra ter-*watermark robust*, maka *watermark* dalam citra tersebut akan tetap dapat dideteksi meskipun citra tersebut dimodifikasi oleh pihak lain. Sistem *watermarking* citra dikatakan *robust* jika PSNR rekonstruksi citra bernilai *infinity*.

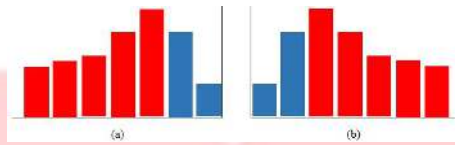
2.2 Histogram Shifting (HS)

Dalam metode *histogram shifting*, nilai *pixel* suatu citra akan ditampilkan dalam bentuk *histogram*. Proses dari metode ini melibatkan modifikasi atau pemindahan intensitas citra dari tingkatan satu ke tingkatan lainnya. *Histogram* citra akan menampilkan titik nol (*zero point*) dan titik puncak (*peak point*) dari intensitas citra. Titik nol memiliki arti bahwa tidak terdapat intensitas *pixel* tersebut pada citra. Titik puncak merupakan intensitas *pixel* yang paling sering muncul. Titik puncak *histogram* akan mempengaruhi nilai kapasitas penyisipan. Semakin besar *peak point*, semakin sedikit ruang yang tersedia untuk menyisipkan data [6].

2.2.1 Skewed Histogram Shifting (Skewed HS)

Metode *skewed HS* akan memanfaatkan kemiringan *histogram* secara asimetris, pada *left skewed HS pixel* yang dimodifikasi atau digeser berada di sebelah kanan. Sedangkan pada *right skewed HS*, *pixel* yang dimodifikasi berada di sebelah kiri. Asimetris *histogram* dapat dilihat pada Gambar 2.2. Warna merah pada

histogram merupakan bagian *long tail*, sedangkan warna biru pada histogram merupakan bagian *short tail*. Keuntungan dari metode ini adalah hanya *pixel* pada bagian *short tail* yang digeser, dan *pixel* pada bagian *long tail* hampir tidak dimodifikasi. Hal ini akan mengurangi adanya distorsi.

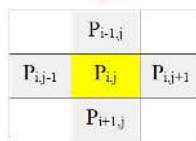


Gambar 2.2 (a) Left Skewed Histogram (b) Right Skewed Histogram.

2.2.2 Skewed Prediction Error Histogram

Metode ini akan memanfaatkan asimetris histogram dan nilai ketetanggaan *pixel* [7]. Pada Gambar 2.4, *pixel* yang ditandai warna kuning merupakan *pixel* awal dan *pixel* yang ditandai warna abu-abu merupakan *pixel* ketetanggaan. Formula untuk mendapatkan nilai *prediction error* (p_e) secara umum dapat menggunakan persamaan 2.1 [3]. Dimana, a adalah *pixel* awal, dan \hat{a} adalah *pixel* ketetanggaan yang diprediksi. \hat{a} dalam persamaan 2.1, akan ditentukan sebagai high estimate atau low estimate. High estimate terjadi jika nilai \hat{a} merupakan nilai *pixel* ketetanggaan tertinggi, sedangkan low estimate terjadi jika nilai \hat{a} merupakan nilai *pixel* ketetanggaan terendah.

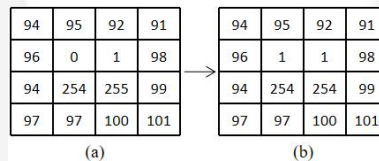
$$p_e = a - \hat{a} \tag{2.1}$$



Gambar 2.3 Ilustrasi *pixel* yang diprediksi dan *pixel* ketetanggaan.

2.3 Overflow dan Underflow

Overflow dan underflow merupakan istilah yang digunakan dalam pemrosesan citra digital jika nilai *pixel* citra keluar dari rentang nilai yang ditentukan. Pada citra grayscale, overflow terjadi saat nilai *pixel* lebih dari 255 dan underflow terjadi saat nilai *pixel* kurang dari 0. Tujuan mengatasi overflow dan underflow adalah untuk menghindari kerusakan pada citra.



Gambar 2.4 Ilustrasi perubahan *pixel* 0 dan 255.

Dalam watermarking, salah satu cara untuk mengatasi masalah ini adalah menggunakan location map. Pixel citra yang bernilai 0 dan 255 dapat diubah menjadi 1 dan 254 sebelum proses penyisipan, dan riwayat perubahan akan disimpan dalam location map. Informasi riwayat perubahan *pixel* citra ini akan digunakan untuk memulihkan nilai *pixel* citra kembali. Persamaan 2.2 digunakan untuk menyimpan riwayat perubahan *pixel* dalam location map dan persamaan 2.3 untuk perubahan nilai *pixel* [2]. LM_i pada persamaan 2.2 adalah LM pada index ke- i . Lalu, p_i pada persamaan 2.3 adalah nilai *pixel* setelah diubah pada index ke- i . Gambar 2.5 (a) merupakan citra berukuran 4x4 yang memiliki *pixel* 0 dan 255. Gambar 2.5 (b) merupakan citra berukuran 4x4 yang *pixel* 0 dan 255 telah diubah. Location map dengan persamaan 2.2 akan menyimpan riwayat tersebut, yaitu $LM_1 [1], LM_2 [0], LM_3 [0], LM_4 [1]$.

$$LM_i = \begin{cases} 0 & \text{jika } p_0 = 1 \text{ atau } 254 \\ 1 & \text{jika } p_0 = 0 \text{ atau } 255 \end{cases} \tag{2.2}$$

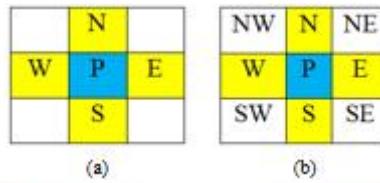
$$p_i = \begin{cases} 254 & \text{jika } p_0 = 255 \\ 1 & \text{jika } p_0 = 0 \\ p_0 & \text{lainnya} \end{cases} \tag{2.3}$$

2.4 Local Complexity

Nilai LC merupakan gambaran tingkat kompleksitas *pixel* citra dalam wilayah tertentu. LC digunakan dalam pemilihan *pixel* untuk mengurangi terjadinya distorsi. Semakin kecil nilai LC, semakin tinggi tingkat kemiripan *pixel* yang diprediksi dengan *pixel* ketetanggaannya. Penyisipan akan dilakukan mulai dari *pixel* dengan nilai LC terendah, karena memiliki distorsi yang lebih sedikit. Berikut adalah cara menghitung LC [8]:

1. LC four-pixel version

$$LC(p) = |N - W| + |W - S| + |S - E| + |N - E| + |W - E| + |N - S| \tag{2.4}$$



Gambar 2.5 Posisi pixel ketetangaan (a) N,W,S, dan E (b) extended.

2. LC extended version

$$\tilde{LC}_{(P)} = LC_{(P)} + LC_{(NW)} + LC_{(NE)} + LC_{(SE)} + LC_{(SW)} \tag{2.5}$$

dimana P adalah pixel tengah yang diprediksi pada wilayah tertentu, dengan ukuran blok 3×3 . N, W, S , dan E merupakan pixel ketetangaan P . Sedangkan, NW, NE, SE, SW adalah pixel ketetangaan extended.

2.5 Serangan Pada Watermarking

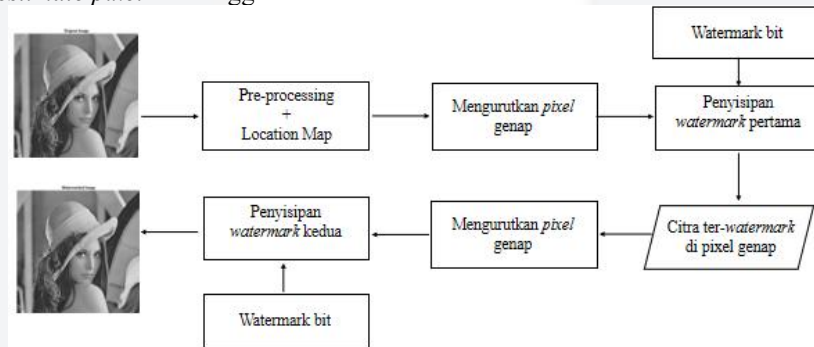
Serangan diberikan setelah melewati proses penyisipan, serangan dilakukan pada citra terwatermark untuk menguji ketahanan dari teknik yang digunakan. Serangan yang diberikan adalah noise. Dengan serangan noise, data atau pesan tanpa arti akan ditambahkan ke dalam citra ter-watermark. Terdapat beberapa serangan noise, antara lain gaussian, salt & pepper, dan speckle.

3. Model Sistem dan Perancangan

3.1 Desain Sistem

3.1.1 Proses Penyisipan dan Ekstraksi

Gambar 3.1 dan Gambar 3.2 merupakan diagram proses penyisipan dan ekstraksi sistem reversible watermarking menggunakan metode skewed histogram shifting. Pada Gambar 3.1, input merupakan citra asli. Lalu, citra akan di proses untuk menghindari masalah overflow dan underflow serta menyiapkan payload di pre-processing dan location map. Payload akan dibagi menjadi dua untuk disisipkan pada citra secara berurut mulai dari pixel genap lalu pixel ganjil. Informasi dari proses ini akan disimpan dalam border pixel citra hasil pre-processing. Kemudian, citra hasil pre-processing akan diurutkan pixel genapnya saja menggunakan local complexity. Selanjutnya, pixel genap citra akan disisipkan watermark pertama. Selain pixel genap citra, watermark juga merupakan input proses ini. Penyisipan ini menggunakan 3 prediktor untuk menentukan high estimate dan low estimate pixel ketetangaan.



Gambar 3.1 Proses Penyisipan.

Persamaan 3.1, persamaan 3.2, dan persamaan 3.3 digunakan untuk menentukan high estimate (\hat{P}_h) dan low estimate (\hat{P}_l) pixel ketetangaan menggunakan prediktor 1, 2, dan 3 secara berurut. Dimana, x_4, x_3, x_2 dan x_1 merupakan pixel ketetangaan dari yang tertinggi hingga terendah. Jika nilai \hat{P}_h sama dengan \hat{P}_l , maka nilai \hat{P}_l akan dikurang 1 untuk menghindari penyisipan ganda [2].

$$\hat{P}_h = x_4, \hat{P}_l = x_1 \tag{3.1}$$

$$\hat{P}_h = \left\lfloor \frac{x_4 + x_3}{2} \right\rfloor, \hat{P}_l = \left\lceil \frac{x_2 + x_1}{2} \right\rceil \tag{3.2}$$

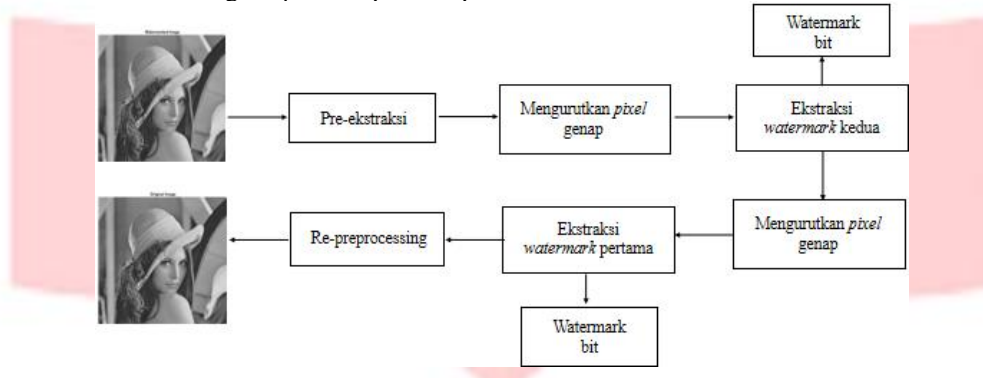
$$\hat{P}_h = \left\lfloor \frac{x_4 + x_3 + x_2}{3} \right\rfloor, \hat{P}_l = \left\lceil \frac{x_3 + x_2 + x_1}{3} \right\rceil \tag{3.3}$$

Selanjutnya, menghitung prediction error. Penyisipan watermark akan terjadi jika nilai prediction error adalah 0. Penyisipan menggunakan metode skewed histogram shifting dapat terjadi pada bagian Positive Histogram Skewed (PHS) atau pada bagian Negative Histogram Skewed (NHS). Untuk mengetahui pada bagian mana penyisipan akan terjadi dapat menggunakan persamaan 3.4 dan persamaan 3.5 [2].

$$PHS = P' = \begin{cases} P + m & \text{jika } P_e = P - \hat{P}_h = 0 \\ P + 1 & \text{jika } P_e = P - \hat{P}_h > 0 \\ P & \text{lainnya} \end{cases} \tag{3.4}$$

$$NHS = P'' = \begin{cases} P' - m & \text{jika } P_e = P' - \hat{P}_l = 0 \\ P' - 1 & \text{jika } P_e = P' - \hat{P}_l < 0 \\ P' & \text{lainnya} \end{cases} \tag{3.5}$$

Output dari proses penyisipan *watermark* pertama adalah citra ter-*watermark* di *pixel* genap. Kemudian, mengurutkan *pixel* ganjil menggunakan *local complexity* dan menyisipkan *watermark* kedua pada *pixel* ganjil yang telah diurutkan. Proses penyisipan *watermark* kedua sama dengan proses penyisipan *watermark* pertama, dimana penyisipan bisa terjadi pada bagian PHS atau NHS. Pada proses penyisipan *watermark* kedua, *watermark* bit juga akan dimasukkan sebagai *input*. Output dari proses tersebut adalah citra ter-*watermark*.



Gambar 3.2 Proses Ekstraksi.

Pada Gambar 3.2, *input* merupakan citra ter-*watermark*. Kemudian, pada proses pre-ekstraksi, sistem akan membaca informasi mengenai panjang *payload* dan prediktor yang digunakan dari *border pixel* citra. Lalu, citra akan diekstraksi pada *pixel* ganjil terlebih dahulu, *pixel* ganjil diurutkan menggunakan *local complexity*. Selanjutnya ekstraksi *watermark* kedua, jika penyisipan terjadi pada PHS maka ekstraksi juga terjadi pada PHS. Untuk mengetahui ekstraksi terjadi pada bagian apa dapat menggunakan persamaan 3.8 dan persamaan 3.9. Lalu, ketentuan bit *watermark* dapat dilihat pada persamaan 3.6 dan persamaan 3.7 [2]. Setelah proses ekstraksi, bit *watermark* akan keluar menjadi output. Lalu *pixel* citra akan diurutkan bagian genapnya menggunakan *local complexity*. Setelah itu, mengekstraksi *watermark* pertama pada *pixel* genap menggunakan persamaan 3.6, persamaan 3.7, persamaan 3.8, dan persamaan 3.9. Bit *watermark* pertama akan keluar menjadi output dan citra akan masuk ke proses *re-preprocessing* untuk mengembalikan proses *pre-processing*. Output dari proses ini adalah citra rekonstruksi.

$$m_2 = \begin{cases} 0 & \text{jika } P_e = P'' - \hat{P}_l = 0 \\ 1 & \text{jika } P_e = P'' - \hat{P}_l = -1 \end{cases} \quad (3.6)$$

$$m_1 = \begin{cases} 0 & \text{jika } P_e = P' - \hat{P}_h = 0 \\ 1 & \text{jika } P_e = P' - \hat{P}_h = 1 \end{cases} \quad (3.7)$$

$$NHS P' = \begin{cases} P'' + 1 & \text{jika } P_e = P'' - \hat{P}_l < 0 \\ P'' & \text{lainnya} \end{cases} \quad (3.8)$$

$$PHS P = \begin{cases} P' - 1 & \text{jika } P_e = P' - \hat{P}_h > 0 \\ P' & \text{lainnya} \end{cases} \quad (3.9)$$

3.2 Parameter Performansi

Untuk mengetahui performansi dari teknik yang diusulkan, dilakukan perhitungan dengan parameter. Parameter yang digunakan antara lain: PSNR dan BER.

3.2.1 Peak to Signal Ratio

PSNR digunakan sebagai parameter penilaian kualitas citra ter-*watermark* dan citra rekonstruksi. Citra asli digunakan sebagai acuan untuk menilai citra ter-*watermark* dan citra rekonstruksi. Semakin besar nilai PSNR, semakin baik kualitas dari citra ter-*watermark*. Nilai PSNR dihitung menggunakan persamaan 3.10 dan persamaan 3.11 [6]. Dimana, 255 merupakan nilai *pixel* tertinggi dari citra *grayscale*. Sedangkan pada persamaan 3.11, M dan N merupakan ukuran dimensi suatu citra. I merupakan koefisien *host* citra pada lokasi (i, j) , dan I' merupakan koefisien citra ter-*watermark* atau citra rekonstruksi pada lokasi (i, j)

$$PSNR = 10 \times \log_{10} \left(\frac{255 \times 255}{MSE} \right) \quad (3.10)$$

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - I'(i, j)]^2 \quad (3.11)$$

3.2.2 Bit Error Rate

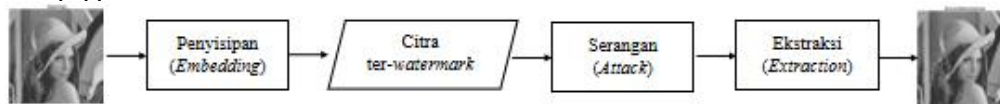
BER merupakan nilai hasil perbandingan jumlah bit *watermark* yang salah terdeteksi dengan jumlah total bit *watermark* secara keseluruhan. Kualitas citra terbaik akan menunjukkan nilai BER mendekati atau sama dengan 0. Persamaan 3.12 adalah cara menghitung nilai BER.

$$BER = \frac{\sum \text{bit watermark yang salah terdeteksi}}{\sum \text{bit watermark keseluruhan}} \quad (3.12)$$

4. Pengujian Sistem dan Analisis

Pengujian yang dilakukan terdiri dari dua skenario. Pertama pengujian terhadap *location map*, jika *location map* dikompresi dan tidak dikompresi tanpa serangan. Kedua, serangan pada citra ter-*watermark* saat *location*

map dikompresi dan tidak dikompresi. Pengujian dilakukan menggunakan 10 citra, termasuk *standard images* lena, baboon, peppers, dan F16.



Gambar 4.1 Proses serangan pada citra ter-watermark.

4.1.1 Location Map

Pada Tabel 4.1, PSNR 1 adalah PSNR citra ter-watermark. Berdasarkan Tabel 4.1, rata-rata PSNR citra ter-watermark jika LM dikompresi adalah 62.1181 dB dan jika LM tidak dikompresi adalah 61.1165 dB. PSNR citra ter-watermark lebih besar jika LM dikompresi karena *payload* yang disisipkan lebih kecil dibandingkan dengan *payload* saat LM tidak dikompresi. *Payload 1* pada Tabel 4.1 merupakan *payload* yang disisipkan saat LM dikompresi dan *payload 2* merupakan *payload* yang disisipkan saat LM tidak dikompresi.

Tabel 4.1 Hasil pengujian LM dikompresi dan tidak dikompresi.

Host Image	LM DIKOMPRESI					LM TIDAK DIKOMPRESI				
	PSNR1	Re PSNR	BER	waktu komputasi	Payload 1	PSNR1	Re PSNR	BER	waktu komputasi	Payload 2
Lena	63.4762	inf	0	2.7991	5038	63.4762	inf	0	2.165	5038
Baboon	59.5739	inf	0	3.3372	5038	59.5739	inf	0	2.7056	5038
Peppers	60.0729	99.3059	0	5.7414	5062	60.0851	99.3059	0	2.8411	5044
F16	66.8706	inf	0	1.8435	5038	66.8706	inf	0	1.7219	5038
Blakeyed	54.5608	102.3162	0	2.1195	5078	51.9876	102.3162	0	2.7703	57375
Clinmill	64.1273	102.3162	0	2.5542	5070	62.0017	102.3162	0	2.0982	8224
Desert	61.0601	102.3162	0	2.3299	5078	57.6474	102.3162	0	2.7055	14946
Fiore	59.7071	inf	0	2.026	5038	59.7071	inf	0	2.2052	5038
Frog	66.6743	102.3162	0	2.6703	5070	65.7706	102.3162	0	2.0116	5885
Hsewoods	65.0579	102.3162	0	2.1366	5070	64.0444	102.3162	0	2.1519	5988

4.1.2 Serangan pada citra ter-watermark

Citra yang digunakan adalah citra ter-watermark, dan akan diberikan serangan *gaussian*, *salt and pepper*, dan *speckle noise*. Tabel 4.2 merupakan pengaruh serangan *gaussian noise* terhadap citra ter-watermark. Berdasarkan Tabel 4.2, sistem *reversible watermarking* ini tidak *robust* terhadap serangan *gaussian noise*. *Watermark* pada citra ter-watermark tidak dapat dideteksi kembali oleh sistem, karena informasi mengenai *payload* pada *border pixel* citra sudah terganggu oleh *noise*. Tabel 4.3 merupakan pengaruh serangan *salt and pepper noise* terhadap citra ter-watermark. Berdasarkan Tabel 4.3 citra ter-watermark yang digunakan tidak *robust* terhadap serangan *salt and pepper noise*. Namun, beberapa citra seperti *blakeyed*, *desert*, dan *hsewoods* mampu menghasilkan BER 0 dengan *density* dan 1×10^{-4} saat LM dikompresi. Tabel 4.4 merupakan pengaruh serangan *speckle noise* terhadap citra ter-watermark. Berdasarkan Tabel 4.4 citra ter-watermark yang digunakan tidak *robust* terhadap serangan *speckle noise*. Hanya satu citra uji yang menghasilkan nilai BER 0 dengan variansi yang cukup kecil yaitu 1×10^{-4} saat LM dikompresi. Sistem ini tidak *robust* terhadap serangan yang digunakan, karena tidak mampu mendeteksi kembali *watermark* yang telah disisipkan pada citra ter-watermark.

Tabel 4.2 Serangan gaussian noise.

Host Image	LM DIKOMPRESI				LM TIDAK DIKOMPRESI			
	v=0.001		v=0.0001		d=0.001		d=0.0001	
	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR
Lena	-	29.9931	-	39.9302	-	29.9931	-	39.9302
Baboon	-	29.9917	-	39.9029	-	29.9917	-	39.9029
Peppers	-	30.0134	-	39.9085	-	30.0134	-	39.9083
F16	-	29.9941	-	39.9404	-	29.9941	-	39.9404
blakeyed	-	30.5129	-	40.0774	-	30.5115	-	40.0139
clinmill	-	30.0572	-	39.9474	-	30.0559	-	39.9372
desert	-	30.1273	-	39.9655	-	30.1251	-	39.9282
fiore	-	29.9929	-	39.905	-	29.9929	-	39.905
frog	-	30.0016	-	39.9435	-	30.0014	-	39.9414
hsewoods	-	30.009	-	39.9391	-	30.009	-	39.9352

Tabel 4.3 Serangan salt and pepper noise.

Host Image	LM DIKOMPRESI				LM TIDAK DIKOMPRESI			
	d=0.001		d=0.0001		d=0.001		d=0.0001	
	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR
Lena	-	35.6985	0.2212	45.7396	-	35.6985	0.2212	45.7396
Baboon	-	35.672	0.3386	46.0507	-	35.672	0.3386	46.0507
Peppers	-	35.5232	0.3154	45.7984	0.5096	35.5233	0.2986	45.7988
F16	-	35.3146	0.0102	44.564	-	35.3146	0.0102	44.564

Blakeyed	-	34.5997	0	45.6594	-	34.5645	0.4998	45.2293
Clinmill	-	35.1339	0.164	46.6127	-	35.1305	0.2666	46.5644
Desert	0.4482	34.7242	0	45.4654	0.4918	34.7122	0.4956	45.3247
Fiore	0.4854	35.296	0.0936	46.8571	0.4854	35.296	0.0936	46.8571
Frog	0.398	35.7984	0.0156	46.0198	0.4128	35.7976	0.0794	46.0112
Hsewoods	0.4862	35.3091	0	45.8397	0.4718	35.3079	0	45.8261

Tabel 4.4 Serangan *speckle noise*.

Host Image	LM DIKOMPRESI				LM TIDAK DIKOMPRESI			
	$v=0.001$		$v=0.0001$		$v=0.001$		$v=0.0001$	
	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR	BER	Re PSNR
Lena	-	35.6282	-	45.3764	-	35.6282	-	45.3764
Baboon	-	35.4022	-	45.0757	-	35.4022	-	45.0757
Peppers	-	35.7003	-	45.4185	-	35.7014	-	45.4213
F16	-	32.7656	-	42.692	-	32.7656	-	42.692
Blakeyed	0.1404	37.5412	0	46.6574	0.498	37.4727	0.5054	46.1247
Clinmill	-	38.0787	-	47.7356	-	38.0729	-	47.6764
Desert	-	39.125	-	48.6499	-	39.0908	-	48.3588
Fiore	-	40.5055	-	49.6097	-	40.5055	-	49.6097
Frog	-	38.5958	-	48.3228	-	38.5937	-	48.3073
Hsewoods	-	37.945	-	47.5826	-	37.9428	-	47.5623

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil analisis menggunakan 10 citra berukuran 512×512 dan *watermark* 5000 bit terhadap skenario pengujian, dapat diambil kesimpulan sebagai berikut:

1. Teknik *reversible* pada sistem berhasil pada citra dengan kondisi tanpa *underflow* dan *overflow*.
2. Citra yang tidak berhasil *recovery* memiliki nilai PSNR rekonstruksi yang tinggi, yaitu lebih dari 90 dB.
3. Sistem dengan *location map* tidak dikompresi mampu menyisipkan dan mengekstrak kembali pesan lebih cepat dibandingkan saat sistem dengan *location map* dikompresi.
4. Jika *watermark* yang ingin disisipkan berukuran besar, maka LM dikompresi, agar ukuran *payload* menjadi lebih kecil.
5. Sistem *reversible watermarking* ini *robust* terhadap serangan *speckle noise* saat LM dikompresi dengan variansi 1×10^{-3} dan 1×10^{-4} rata-rata BER yang dihasilkan adalah 0.1404 dan 0 secara berurut, dan serangan *salt and pepper noise* saat LM dikompresi dengan *density* 1×10^{-4} rata-rata BER yang dihasilkan adalah 0.1159.

5.2 Saran

Berdasarkan kesimpulan diatas, sistem dapat dikembangkan menjadi lebih baik dengan saran sebagai berikut:

1. Melakukan pengembangan teknik *reversible watermarking* agar dapat mengatasi masalah *overflow* dan *underflow*.
2. Melakukan pengembangan teknik *reversible watermarking* pada citra warna dan video.
3. Melakukan pengembangan teknik *reversible watermarking* yang *robust*.

Daftar Pustaka:

- [1] M. Ortiz, C. Feregrino-Urbe, R. Hasimoto-Beltran, and J. J. Garcia-Hernandez, "A Survey on Reversible Watermarking for Multimedia Content: A Robustness Overview," *IEEE Access*, vol. 7, pp. 132662–132681, 2019.
- [2] S. Kim, X. Qu, V. Sachnev, and H. J. Kim, "Skewed Histogram Shifting for Reversible Data Hiding Using a Pair of Extreme Predictions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3236–3246, 2019.
- [3] R. Uyyala and R. Pal, "Reversible Data Hiding Using Improved Gradient Based Prediction and Adaptive Histogram Bin Shifting," *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 720–726, 2020.
- [4] Tang X, Zhou L, Liu D, Shan W, Zhang Y. "Border following-based reversible watermarking algorithm for images with resistance to histogram overflowing," *International Journal of Distributed Sensor Networks*. vol 16(5), 2020.
- [5] S. Kaur and M. Shukla, "Reversible Data Hiding Technique and its Type, a survey," *IOSR J. Comput. Eng.*, vol. 3, no. 5, pp. 43–48, 2016.
- [6] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–361, 2006.

- [7] Y. Yang and J. L. Wu, "Two-Bit Embedding Histogram-Prediction-Error Based Reversible Data Hiding for Medical Images with Smooth Area," *Computers*, vol. 10, no. 11, 2021
- [8] W. He, G. Xiong, and Y. Wang, "Reversible data hiding based on multi-predictor and adaptive expansion," *IET Image Process.*, vol. 16, no. 3, pp. 888–899, 2022.
- [9] H. Harahap, G. Budiman, L. Novamizanti, Implementasi Teknik Watermarking menggunakan FFT dan Spread Spectrum Watermark pada Data Audio Digital, *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Elektronika*, ISSN (p), 2338-8323, 2016.
- [10] G. Budiman, L. Novamizanti, White Space Steganography On Text By Using LZW-Huffman Double Compression, *International Journal of Computer Networks & Communications*, Vol. 7 (2), 136A, 2015.
- [11] L. Novamizanti, AB Suksmono, D. Danudirdjo, G. Budiman, Robust Reversible Watermarking using Stationary Wavelet Transform and Multibit Spread Spectrum in Medical Images. *Int. J. Intell. Eng. Syst.*, vol. 15, no. 3, pp. 343–354, 2022.
- [12] L. Novamizanti, AB. Suksmono, D. Danudirdjo, & G. Budiman, Robust and Imperceptible Watermarking on Medical Images using Coefficient Pair Modification. *IIUM Engineering Journal*, vol. 24, no. 1, pp. 88-105, 2023.