Otentikasi Citra dengan *Fragile Watermarking* pada Citra GIF

Farid Firdaus
School of Electrical Engineering and Informatics
Institute Technology of Bandung
10th Ganeca Street
Bandung, Indonesia
firdaus.farid22@gmail.com

Dr. Ir. Rinaldi Munir, MT.
School of Electrical Engineering and Informatics
Institute Technology of Bandung
10th Ganeca Street
Bandung, Indonesia
rinaldi@informatika.org

***Abstrak*— Penggunaan citra GIF yang cukup signifikan menjadikan citra GIF rawan dimanipulasi oleh pihak yang tidak bertanggungjawab. Untuk mencegah hal tersebut, dapat digunakan teknologi *fragile watermarking.* Karakteristik khusus dari citra GIF sebagai citra berbasis palet menjadikan tidak semua metode *fragile watermarking* dapat diterapkan pada citra GIF.**

Dalam penerapan teknologi *fragile watermarking*, metode dasar yang paling umum digunakan adalah metode LSB Steganografi. Teknik ini menyisipkan data *watermark* kedalam LSB citra GIF. Metode lain yang dapat digunakan untuk menyisipkan data pada citra GIF adalah EzStego. Pengembangan metode EzStego dilakukan oleh Fridrich dan memberikan kualitas citra yang lebih baik jika dibandingkan EzStego. Kedua metode ini membuka peluang penerapan *fragile watermarking* pada citra GIF.

Dalam penelitian ini, metode yang diimplementasikan adalah metode LSB Steganografi dan metode EzStego dengan *parity bit.* Implementasi menggunakan bahasa Java dengan paradigma pemrograman *object-oriented.* Program mampu menyisipkan *fragile watermark* pada citra GIF, dan mampu mengekstraksi *watermark* pada citra GIF yang telah disispkan *watermark.* Pengujian dilakukan dengan meyisipkan *watermark*, menyerang citra GIF tersebut, dan mengekstraksi *watermark* dari citra GIF yang telah diserang. Pengujian implementasi dilakukan dengan tiga buah citra dengan karakteristik khusus.

Hasil pengujian menunjukkan metode EzStego dengan *parity bit* merupakan metode yang cocok diterapkan sebagai teknik *fragile watermarking* pada citra GIF. Metode EzStego dengan *parity bit* memberikan kualitas yang baik, terkecuali pada citra natural berwarna. Metode EzStego mampu mendeteksi serangan tanpa menimbulkan *noise* yang tidak diperlukan. Metode LSB Steganografi tidak cocok dikarenakan timbulnya *noise* yang cukup banyak. *Noise* tersebut muncul akibat warna baru yang muncul tidak terdefinisi di dalam palet citra GIF.

Kata Kunci - EzStego, fragile watermarking, LSB Steganografi, parity bit.

I. PENDAHULUAN

Di zaman *digital* saat ini, penyebaran *file digital* dapat dilakukan dengan sangat mudah. Dampak dari arus penyebaran *file digital* adalah kemudahan individu untuk mendapatkan *file* tersebut dengan mudah.

Salah satu alat yang dapat digunakan untuk mengotentikasi keaslian suatu *file* khususnya *file* citra adalah dengan menggunakan *fragile watermarking. Fragile watermarking* adalah teknik menyisipkan *watermark* yang rapuh kedalam citra utama yang akan diotentikasi. Tujuan dari penyisipan *watermark* yang rapuh, agar saat dilakukan manipulasi pada citra tersebut, *watermark* dari citra tersebut akan rusak dan memunculkan pola manipulasi sehingga bagian yang dimanipulasi dapat teridentifikasi dengan baik.

Citra GIF merupakan salah satu format citra digital yang menggunakan metode kompresi *lossless compression.* Citra GIF banyak digunakan dalam forum internet dikarenakan hasil kompresi yang ringan, namun tetap memiliki kualitas citra yang baik. Oleh karena itu, penyebaran citra GIF menjadi sangat cepat sehingga rawan mengalami serangan / manipulasi oleh pihak yang tidak bertanggungjawab.

Metode dasar yang sering digunakan dalam menerapkan *fragile watermarking* adalah menggunakan metode LSB (*least significant bit*) Steganografi. Metode LSB Steganografi dilakukan dengan menyisipkan data yang akan disembunyikan kedalam LSB citra penampung.

Salah satu metode penyisipan data yang dapat digunakan pada citra berbasis palet adalah menggunakan EzStego. Metode steganografi yang dapat diterapkan pada citra berbasis palet, juga diusulkan oleh Fridrich [3]. Kedua metode ini digunakan untuk menyembunyikan pesan ke dalam citra berbasis palet dengan melakukan manipulasi pada *pixel* dan palet citra yang akan disisipkan data.

Pada makalah ini, akan dibahas tentang penerapan metode *fragile watermarking* pada citra GIF menggunakan beberapa metode yang telah disebutkan sebelumnya. Hasil pengujian menggunakan metode tersebut akan digunakan untuk menentukan metode terbaik yang dapat digunakan untuk *fragile watermarking* pada citra GIF.

II. STUDI LITERATUR

A. Citra Digital

Menurut Sachs [13] dalam Adityas [1] mendefinisikan citra digital sebagai sebuah representasi elektronik/virtual sebuah citra yang tersimpan di dalam memori komputer. Citra digital tersimpan di dalam memori dalam bentuk kumpulan *bit* yang teratur. Ukuran dari *bit* tersebut bergantung pada format ukuran citra yang digunakan. Citra digital dapat dikatakan sebagai larik persegi dari *pixel* yang disebut *bitmap*

Citra digital memiliki beberapa jenis. Pembagian jenis ini berdasarkan karakteristik citra, baik ditinjau dari sisi ukuran *bit* penyimpanan atau dari sisi konten secara visual. Citra biner adalah citra digital dengan ukuran penyimpanan hanya satu *bit*. Pada citra biner, setiap *pixel* hanya terdiri dari dua buah nilai yaitu 0 dan 1 (terdiri dari warna hitam dan putih). Citra hitam putih adalah citra digital dimana *bit* untuk setiap *pixel* merepresentasikan derajat keabuan (*grayscale*). Citra berwarna dihasilkan dengan mengkombinasikan *pixel* dengan ukuran 3 *byte* (24 *bit*). Masing-masing *byte* pada citra berwarna menyimpan tiga buah warna dasar yaitu merah, hijau, dan biru (pada citra dengan warna RGB).

B. PSNR (Peak Signal-to-Noise Ratio)

Peak Signal-to-Noise Ratio atau disingkat PSNR adalah rasio perbandingan antara maksimum kekuatan sinyal pada citra terhadap maksimum kekuatan *noise* yang ada sehingga mempengaruhi kualitas citra. Selain diguankan untuk mengukur besar pengaruh *noise* terhadap gambar, PSNR dapat digunakan untuk membandingkan dua gambar yang berbeda [15].

PSNR dapat dihitung dengan menentukan nilai MSE (*mean square error*) terlebih dahulu. Jika sebuah citra monokrom *I* berukuran *m* x *n* dan noise *K*, dan jumlah rentang nilai *pixel* sebanyak MAX, maka nilai MSE dapat dihitung melalui persamaan (II.1) dan PSNR dapat dihitung dengan persamaan (II.2), (II.3), dan (II.4).

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \qquad \text{(II.1)}$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad \text{(II.2)}$$

$$= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \qquad \text{(II.3)}$$

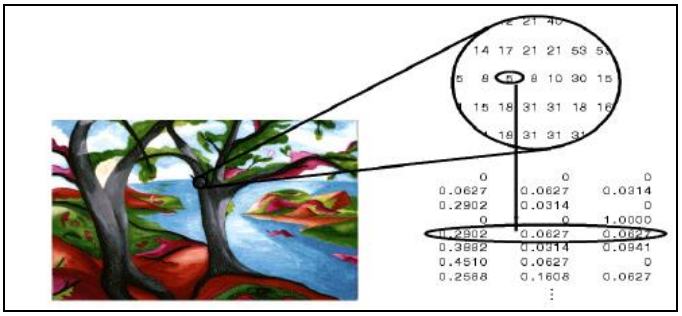
$$= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \qquad \text{(II.4)}$$

Karena rentang nilai PSNR yang cukup besar, maka PSNR direpresentasikan menggunakan satuan logaritmik desibel (dB). PSNR sering digunakan dalam pengukuran kualitas kompresi dari citra yang diolah. Semakin tinggi nilai dari PSNR, maka semakin bagus kualitas yang dihasilkan.

C. Citra GIF

GIF pertama kali diperkenalkan oleh *CompuServ* pada tahun 1987. Format GIF menggunakan metode *lossless data compression* dengan algoritma Lempel-Ziv-Welch. Format GIF mendukung fitur gambar bergerak (animasi). Gambar bergerak diciptakan dengan menampilkan gambar yang memiliki kemiripan konten secara terus-menerus. Hal ini menimbulkan kesan seolah-olah gambar tersebut sedang bergerak.

Citra GIF merupakan citra yang berbasis indeks dan palet. Nilai-nilai yang tertera pada data citra GIF merupakan nilai indeks yang akan merujuk ke palet citra GIF. Oleh dikarenakan ukuran palet yang terbatas, maka citra GIF memiliki jumlah maksimal warna yang dapat direpresentasikan kedalam citra. Ilustrasi dari struktur citra GIF dapat dilihat pada Gambar 1.



Gambar 1. Ilustrasi Struktur Citra GIF (Munir, 2015)

D. Konsep Umum Digital Watermarking

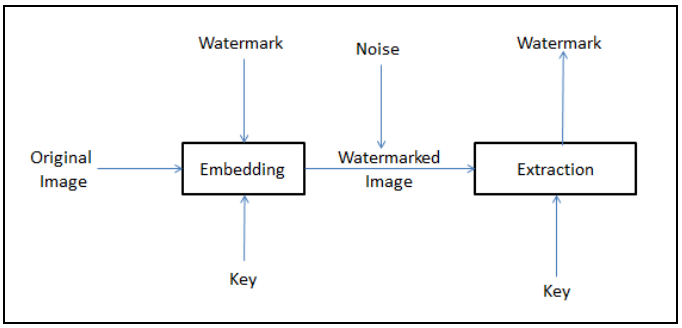
Peak Signal-to-Noise Ratio atau disingkat PSNR adalah rasio perbandingan antara maksimum kekuatan sinyal pada citra terhadap maksimum kekuatan *noise* yang ada sehingga mempengaruhi kualitas citra. Selain diguankan untuk mengukur besar pengaruh *noise* terhadap gambar, PSNR dapat digunakan untuk membandingkan dua gambar yang berbeda [15].

Digital watermarking adalah aksi menyembunyikan pesan pada entitas yang berhubungan dengan sinyal digital (seperti citra, suara, atau video) dengan menyisipkan pesan pada sinyal digital itu sendiri. Pemberian *watermark* yang berbeda pada setiap produk digital yang dibeli oleh konsumen, dapat digunakan sebagai tanda bukti apabila suatu saat nanti produk tersebut tersebar secara ilegal di jaringan internet, penyebar produk tersebut dapat dilacak melalui *watermark* pada produk yang tersebar. Proses penyisipan dan ekstraksi *watermark* kedalam produk digital dapat dilihat pada Gambar 2.

Untuk mengenali *watermark* tersebut, digunakan kunci, dimana kunci ini akan digunakan kembali saat proses ekstraksi. Penggunaan kunci yang salah akan mengakibatkan kegagalan pada proses ekstraksi.

Untuk menguji kinerja dari *watermark* yang digunakan, citra yang telah disisipi *watermark* akan diserang. Pengecekan kinerja dilakukan dengan mengekstraksi kembali *watermark* dari citra yang telah diserang menggunakan kunci yang telah digunakan saat proses *embedding* sebelumnya.

Salah satu jenis *watermark* adalah *fragile watermarking*. *Fragile watermarking* digunakan untuk proses otentikasi suatu media digital. Salah satu media digital yang dimaksud adalah media citra. Kelebihan dari *fragile watermarking* adalah pada citra yang telah diserang, area pada citra yang diserang dapat dikuantisasi dengan baik.



Gambar 2. Diagram skema *watermarking* pada media citra secara umum.

Pada ilustrasi Gambar 3, citra ber-*watermark* yang telah dimanipulasi , saat dilakukan ekstraksi pada *watermark*, bagian yang diserang akan terkuantisasi dengan baik sehingga pengguna dapat mengidentifikasi bagian citra yang telah diserang

Gambar 3. Citra ber-*watermark*, Citra ber-*watermark* yang mengalami serangan ,*dan Watermark* yang telah diekstrak. (Wong dkk., 2000)

E. LSB Steganografi

Salah satu metode steganografi yang dapat digunakan untuk teknik *watermarking* adalah metode LSB Steganografi. Sebagian besar teknik *fragile watermarking* melakukan penyisipan *watermark* langsung ke dalam domain spasial dari citra penampung [2].

Metode ini menyisipkan data kedalam LSB dari citra penampung. Pemilihan penyisipan pada LSB dikarenakan nilai LSB tidak begitu mempengaruhi kualitas citra. Semakin banyak kanal LSB yang digunakan dalam sebuah *byte pixel*, maka semakin besar perbedaan kualitas gambar yang dihasilkan. Untuk setiap kanal LSB pada sebuah *pixel* dapat menampung data sebanyak satu bit.

Proses ekstraksi *watermarking* dilakukan dengan mengambil nilai LSB sebanyak jumlah kanal LSB yang digunakan per setiap *pixel*. Bit yang telah diekstraksi akan disusun kembali sesuai urutan penyisipannya untuk membentuk kembali data yang telah disisipkan. Urutan penyusunan bit harus sama dengan urutan saat menyisipkan data tersebut agar bit yang diekstraksi dapat merepresentasikan data dengan benar.

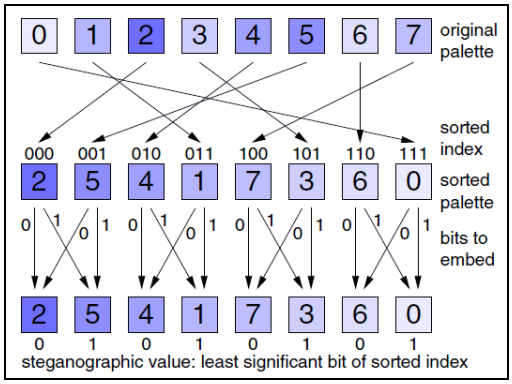
F. EzStego

EzStego (Machado, 1996) merupakan salah satu algoritma steganografi yang digunakan pada citra berbasis palet. Dikarenakan jumlah warna yang dapat disimpan menggunakan palet terbatas, maka penyimpanan data dilakukan dengan memanipulasi data citra dan palet citra yang telah ada.

Palet warna akan diurutkan berdasarkan kedekatan warna. Jarak antara dua buah warna dapat dihitung menggunakan rumus jarak *euclidan*. Jarak *euclidan* antara dua buah warna dapat dihitung menggunakan persamaan (II.5).

$$d = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \qquad \text{(II.5)}$$

Data yang akan disimpan akan dibandingkan dengan indeks palet yang sedang dirujuk. Pada Gambar 4, jika data yang akan disimpan merupakan bit bernilai 1, warna yang akan dijadikan penampung bit adalah indeks palet nomor dua, maka warna yang baru yang akan menggantikan warna pada indeks palet nomor dua adalah warna pada indeks palet nomor lima. Pemilihan indeks palet nomor lima dikarenakan kedekatan warnanya.



Gambar 4. Pengurutan warna pada palet didasarkan pada kedekatan warna. (Pfitzmann, 2006)

Skema *parity bit* merupakan skema modifikasi dari EzStego [3]. Perbedaan dari skema *parity bit* dan skema EzStego biasa terletak pada pemilihan warna pada palet yang akan digunakan untuk menyimpan data.

Palet warna yang diurutkan berdasarkan kecerahan dan memiliki kelemahan yaitu terdapat loncatan warna yang disebabkan dua buah warna yang memiliki tingkat kecerahan berdekatan, namun merepresentasikan warna yang berbeda. Untuk menghindari loncatan warna saat melakukan pengurutan pada palet, maka pemilihan warna yang akan digunakan sebagai media penyimpanan data, dipilih berdasarkan nilai *parity bit*. Perhitungan nilai *parity bit* dari warna dengan format RGB dapat dihitung menggunakan persamaan (II.6)

$$ParityBit = (R + G + B) \bmod 2 \qquad \text{(II.6)}$$

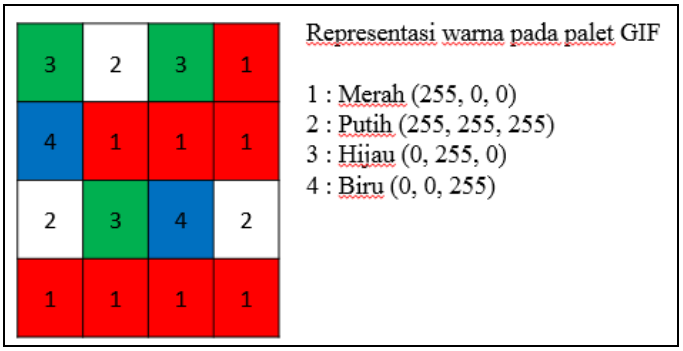
III. PEMBAHASAN ANALISIS DAN USULAN SOLUSI

A. Analisis Masalah

Citra GIF merupakan citra digital yang berbasis palet. Setiap citra GIF memiliki palet yang unik. Palet merupakan kumpulan warna unik yang direpresentasikan ke dalam citra GIF. Palet pada citra GIF memiliki jumlah warna yang

terbatas. Setiap citra GIF memiliki ukuran palet yang dapat menampung sebanyak 256 warna.

Pixel pada citra GIF direpresentasikan dalam bentuk indeks pada palet. Nilai indeks ini yang kemudian digunakan untuk mencari representasi warna *pixel* pada palet citra GIF yang bersangkutan. Pengubahan warna pada indeks palet, akan menghasilkan perubahan warna pada *pixel* yang merujuk terhadap indeks pada palet citra GIF. Ilustrasi representasi *pixel* pada citra GIF dapat dilihat pada Gambar 5.



Gambar 5. Representasi *Pixel* pada Citra GIF.

Dalam penelitian ini, salah satu metode yang diajukan untuk diuji adalah metode LSB Steganografi. Metode LSB Steganografi telah banyak digunakan untuk teknik *fragile watermarking* pada format non-GIF. Pada penelitian ini akan dicoba untuk menerapkan LSB Steganografi pada citra GIF tanpa melakukan dekompresi pada citra GIF tersebut.

Pengujian dilakukan dalam dua tahapan yaitu uji penyisipan *watermark* dan uji ekstraksi *watermark.* Uji penyisipan *watermark* dilakukan untuk menguji seberapa bagus kualitas citra setelah disisipkan *watermark.* Uji ekstraksi *watermark* dilakukan untuk menguji kualitas *watermark* dalam mendeteksi serangan yang dilakukan. Hasil pengujian untuk uji penyisipan dapat dilihat pada Gambar 6. Hasil pengujian untuk uji ekstraksi dapat dilihat pada Gambar 7.

Gambar 6. (Kiri) Citra GIF sebelum Disisipkan *Watermark.* (Kanan) Citra GIF setelah Disisipakan *Watermark.*

Gambar 7. *Watermark* yang Diekstraksi dari Citra GIF Menggunakan Metode LSB Steganografi.

Berdasarkan hasil pengujian menggunakan LSB Steganografi, saat menyisipkan, kualitas citra yang dihasilkan secara kasat mata tidak memiliki perbedaan dari citra sebelum disisipkan *watermark*. Adapun dari uji ekstraksi, *watermark* yang dihasilkan tidak memiliki kualitas yang bagus. Hal ini terlihat dari cukup banyak *noise* yang timbul, padahal tidak dilakukan serangan pada citra tersebut sehingga sulit untuk membedakan daerah pada citra yang mengalami serangan.

B. Usulan Solusi

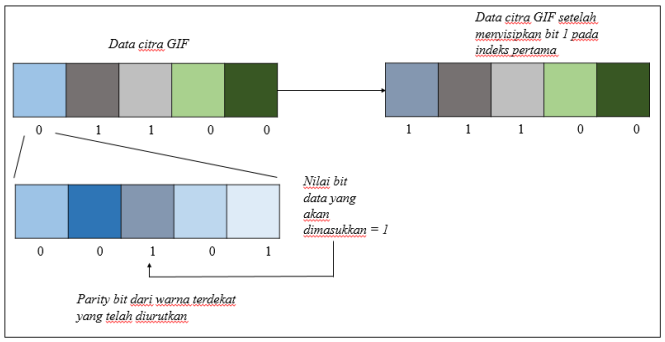
Solusi yang akan ditawarkan merupakan implementasi dari metode EzStego dengan *parity* bit. Secara garis besar, proses tahapan penyisipan *watermark* dilakukan dengan tahapan sebagai berikut:

1. Memilih kunci, citra GIF, dan citra *watermark.*
2. Melakukan konversi citra GIF dan citra *watermark* kedalam bentuk larik *pixel.*
3. Menghasilkan bit *hash* dengan masukan berupa kunci dan dimensi dari citra.
4. Melakukan operasi bit XOR dengan operan nilai bit *hash* dan larik *pixel watermark* sehingga menghasilkan data yang siap disisipkan kedalam citra GIF.
5. Menyisipkan data dengan cara melakukan pemilihan warna pengganti pada palet berdasarkan warna *pixel* dan bit data yang akan disisipkan.
6. Melakukan konversi dari larik *pixel* GIF kembali kedalam bentuk citra GIF yang telah disisipkan *watermark.*

Metode ekstraksi *watermark* pada gambar yang telah disisipkan *watermark* secara garis besar dilakukan dengan tahapan sebagai berikut:

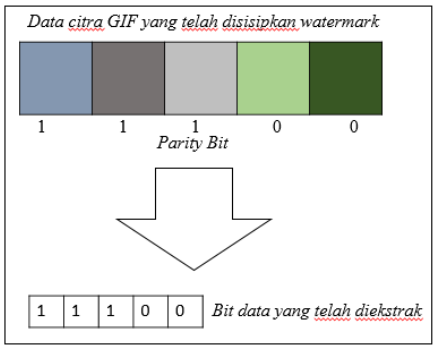
1. Memasukkan kunci dan citra ber-*watermark* sebagai nilai masukan ke dalam fungsi *hash*.
2. Melakukan pra-proses terhadap citra ber-*watermark* yang akan diekstrak ke dalam bentuk larik *pixel.*
3. Mengekstraksi bit data dengan cara menghitung nilai *parity bit* dari *pixel* citra GIF.
4. Bit yang telah diekstraksi, akan dilakukan operasi bit XOR terhadap bit *hash* untuk menghasilkan bit *watermark.*

5. Melakukan konversi dari bit *watermark* ke dalam bentuk citra bitmap.



Gambar 8. Pemilihan Warna Pengganti dengan Menggunakan *Parity Bit*

Proses penyisipan menggunakan *parity bit* dapat dilihat pada Gambar 8. Proses ekstraksi menggunakan *parity bit* dapat dilihat pada Gambar 9.



Gambar 9. Ekstraksi Bit Data Menggunakan *Parity Bit*.

IV. PENGUJIAN

A. *Pelaksanaan Pengujian*

Tujuan dari pengujian ini adalah menguji performansi dari metode EzSteo menggunakan *parity bit* pada teknik *fragile watermarking* pada citra GIF. Performansi *fragile watermarking* terdiri dua jenis, yaitu kualitas citra GIF setelah disisipkan *watermark*, dan performansi *watermark* dalam mendeteksi serangan pada citra GIF yang telah disisipkan *fragile watermark.*

Performansi yang baik ditunjukkan dengan kualitas citra GIF yang tidak berubah secara signifikan setelah disisipkan *watermark.* Secara kuantitatif, performansi ini diukur dengan menghitung nilai PSNR. Semakin besar nilai PSNR, maka semakin bagus kualitas citra yang dihasilkan. Untuk performansi *watermark*, diukur dengan kemampuan *watermark* mendeteksi daerah yang mengalami serangan tanpa adanya *noise* pada *watermark* yang telah diekstrak.

Pengujian pertama dlakukan dengan tahapan sebagai berikut :

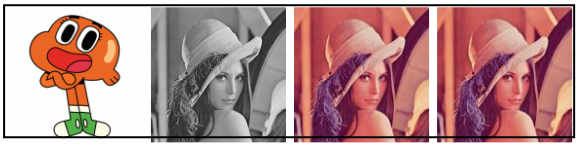
1. Menyiapkan citra GIF dan citra *watermark.*
2. Menyisipkan citra *watermark* ke dalam citra GIF dengan kunci K.
3. Menghitung nilai PSNR dari citra GIF yang telah disisipkan *watermark.*

Tahap kedua dilakukan untuk menguji performansi *watermark* dalam mendeteksi serangan yang akan dilakukan. Pengujian kedua dapat dilakukan dengan memanfaatkan hasil pengujian pertama. Pengujian kedua dilakukan dengan tahapan sebagai berikut :

1. Menyiapkan citra GIF yang telah disisipkan dengan citra *watermark.*
2. Melakukan berbagai serangan visual pada citra yang telah disisipkan *watermark.*
3. Melakukan ekstraksi pada citra ber-*watermark* yang telah diserang. Ekstraksi dilakukan menggunakan kunci K.

Ketiga karakteristik gambar ditunjukkan pada Gambar 10. Karakteristik gambar dan *file* citra yang berkaitan dapat dilihat sebagai berikut :

1. Warna tak-tergradasi : *cartoon.gif*
2. Warna natural *grayscale* : *lena.gif* (*grayscale*)
3. Warna natural berwarna : *lena.gif*



Gambar 10. citra *cartoon.gif*, citra *lena.gif* (*grayscale*), citra *lena.gif* (berwarna), dan citra *watermark.bmp*. (google.com, 2016)

B. *Hasil Pengujian*

Kualitas dari citra yang telah disisipkan dapat diukur menggunakan PSNR (*Peak Signal -to-Noise Ratio*). Semakin tinggi nilai PSNR, maka kualitas citra yang dihasilkan semakin bagus. Satuan dari PSNR adalah dB (desibel). Pengujian dilakukan pada dua jenis citra berdasarkan karakteristiknya, yaitu citra dengan warna tergradasi, dan citra dengan warna tak-tergradasi. Untuk citra dengan warna tergradasi, digunakan citra lena, sedangan untuk yang tidak, digunakan citra kartun. Nilai PSNR dari ketiga citra yang telah disisipkan *watermark* dapat dilihat pada Tabel 1.

Tabel 1. PSNR Hasil Pengujian pada Citra GIF

Citra GIF	Citra *Watermark*	PSNR (dB)
cartoon.gif	watermark.bmp	33.84012030261622
lena.gif (*grayscale*)	watermark.bmp	49.33643225445584
lena.gif (berwarna)	watermark.bmp	28.41304911023644

Pada Tabel 1, menunjukkan penyisipan *watermark* pada citra *grayscale* menunjukkan hasil yang terbaik. Sedangkan penyisipan *watermark* pada citra Lena (berwarna) menunjukkan hasil yang kurang memuaskan. Nilai PSNR yang berada dibawah 30 menunjukkan metode yang digunakan masih belum bagus jika diterapkan pada citra natural berwarna.

Serangan dilakukan dengan melakukan manipulasi terhadap citra yang telah disisipkan *watermark*. Beberapa jenis serangan yang dilakukan adalah melakukan rotasi, *flip, resize*, dan *cropping*. Dari sisi keamanan, akan dilakukan pengujian dengan memasukkan kunci yang salah. Hasil pengujian terhadap performansi *fragile watermarking* dapat dilihat pada Gambar 11.

Gambar 11. Hasil Ekstraksi *watermark* dari Citra GIF yang telah Diserang.

C. Analisis Hasil Pengujian

Nilai PSNR sangat bergantung pada nilai selisih bit sebelum dan sesudah disisipkan dengan *watermark.* Nilai selisih akan semakin besar seiring bertambahnya jarak antara kedua warna tersebut. Pemilihan warna pengganti sendiri sangat bergantung pada nilai *parity bit* dari warna palet yang telah diurutkan. Persebaran nilai *parity bit* yang tidak merata akan mengakibatkan warna pengganti yang dipilih akan semakin jauh. Akibatnya, nilai PSNR yang akan dihasilkan semakin rendah.

Pada pengujian sebelumnya, didapatkan nilai PSNR dari citra *grayscale* memiliki nilai yang paling baik. Hal ini disebabkan nilai merah, hijau, dan biru pada sebuah warna *grayscale* memiliki nilai yang sama. Pola ini menyebabkan penyebaran nilai *parity bit* lebih merata dibandingkan citra berwarna.

Pada Tabel 3, jika nilai elemen penyusun dari warna adalah bilangan ganjil, nilai *parity bit* yang dihasilkan adalah 1, sedangkan untuk bilangan genap, nilai *parity bit* yang dihasilkan adalah 0. Dengan demikian, warna pengganti yang akan dipilih memiliki selisih yang tidak jauh berbeda dari warna aslinya.

Selain persebaran dari nilai *parity bit*, jumlah warna yang tersedia pada palet citra memiliki pengaruh penting. Semakin sedikit jumlah warna pada palet, maka jumlah pilihan warna pengganti yang tersedia akan semakin sedikit. Akibatnya, perbedaan warna yang akan dipilih sebagai warna pengganti, berpotensi memiliki nilai yang besar.

Tabel 3. Nilai Parity Bit Berdasarkan Nilai RGB Penyusunnya pada citra *grayscale*.

Merah (R)	**Hijau (G)**	**Biru (B)**	***Parity Bit***
0	0	0	0
1	1	1	1
2	2	2	0
…	…	…	…
255	255	255	1

Selain persebaran dari nilai *parity bit*, jumlah warna yang tersedia pada palet citra memiliki pengaruh penting. Semakin sedikit jumlah warna pada palet, maka jumlah pilihan warna pengganti yang tersedia akan semakin sedikit. Akibatnya, perbedaan warna yang akan dipilih sebagai warna pengganti, berpotensi memiliki nilai yang besar.

V. KESIMPULAN

Berdasarkan hasil pengujian dan eksperimen dapat diambil kesimpulan yaitu :

A. Kesimpulan

1. Metode yang cocok digunakan untuk menerapkan teknik *fragile watermarking* pada citra GIF adalah metode EzStego dengan pemilihan warna menggunakan metode *parity bit*. Metode LSB Steganografi tidak cocok karena saat melakukan ekstraksi, *watermark* mengandung *noise* yang cukup banyak.

2. Hasil pengujian menggunakan EzStego dengan *parity bit* menunjukkan hasil yang memuaskan. *Fragile watermark* dapat tersimpan dengan baik dan mampu mendeteksi serangan yang dilakukan pada citra yang telah disisipkan *watermark.* Metode ini sangat cocok digunakan untuk citra GIF natural (*grayscale*) dan citra GIF dengan warna tak-tergradasi. Penyisipan *fragile watermarking* pada citra GIF dilakukan menggunakan algoritma EzStego pada domain spasial dengan pemilihan warna pengganti menggunakan metode Fridrich. Bit hash diperoleh melalui operasi hash terhadap identitas citra dan kunci. Operasi hash dilakukan agar *watermark* hanya dapat diekstrak oleh pihak yang memiliki kunci yang benar. Penyisipan dilakukan dengan mengganti warna pada posisi (x, y) dengan warna pengganti yang memiliki nilai *parity bit* yang sama dengan bit yang akan disisipkan. Pemilihan warna menggunakan *parity bit* agar warna pengganti memiliki jarak yang dekat dengan warna sebelumnya. Kelemahan dari penggunaan metode *parity bit* adalah waktu eksekusi program yang menjadi cukup lama. Ekstraksi *fragile watermarking* dilakukan dengan menghitung nilai *parity bit* dari seluruh *pixel* pada citra GIF. Bit

tersebut kemudian diubah kembali menjadi bit *watermark* dengan menggunakan fungsi hash dari kunci yang digunakan saat menyisipkan *watermark*. Citra *watermark* kemudian dibentuk kembali dengan menyusun bit-bit tersebut. Proses ekstraksi berlangsung sangat cepat dikarenakan penggunaan metode *parity bit* tanpa mengurangi kualitas *watermark* yang diekstrak.

3. Hasil pengujian menunjukkan metode LSB Steganografi tidak bagus digunakan dalam teknik *fragile watermarking* pada citra GIF. Hal ini disebabkan munculnya *noise* yang cukup banyak sehingga sulit untuk mendeteksi daerah yang mengalami serangan. *Noise* ini muncul akibat warna baru yang tidak terdefinisi di dalam palet GIF sehingga *pixel* yang bersangkutan menjadi rusak dan akan terdeteksi sebagai *noise*.

ACKNOWLEDGMENT

Penulis berterima kasih kepada Bapak Dr.Ir.Rinaldi Munir, MT. yang telah banyak membantu dalam memberikan sumbangan pemikiran, serta saran yang membangun selama pengerjaan penelitian dan makalah ini.

DAFTAR REFERENSI

[1] Adityas, R (2014), Skema Fragile Watermarking dengan Fungsi Hash dan Ketergantungan Blok Tak Deterministik, Teknik Informatika, ITB.

[2] Alomari, R, Ahmed Al-Jaber (2004), *A Fragile Watermarking Algorithm for Content Authentication.* University of Jordan, Jordan.

[3] Fridrich, J (1999), A New Steganographic Method for Palette Images, IS&T PICS : Savannah, Georgia, USA.

[4] Lin, Edward (2001), *A Review of Fragile Image Watermarks*, Purdue University, Indiana, USA.

[5] Liu, SH, dkk (2007), *An image fragile watermark scheme based on chaotic image pattern and pixel-pairs.* Harbin Institute of Technology, Chinese Academy of Science, P.R. China.

[6] http://1.bp.blogspot.com/-eruST5hnkNo/VAHI-5z8_wI/AAAAAAAAAB4/lj1wmkD5M-I/s1600/DARWIN.gif Tanggal Akses : 5 Jan 2016.

[7] http://docsdrive.com/images/ansinet/itj/2010/fig4-2k9-20-26.jpg Tanggal Akses : 6 Januari 2016.

[8] https://en.osdn.jp/projects/sfnet_crypto2011/ Tanggal Akses : 20 November 2015.

[9] Hu, J, dkk (2002). Image fragile watermarking based on fusion of multi-resolution tamper detection. Electronics Letters, 38(24):1512-1513. P.R China.

[10] Munir, R (2015), Chaos-based Modified "EzStego" Algorithm for Improving Security of Message Hiding in GIF Image, IC3INA, STEI ITB, Bandung, Indonesia

[11] Pfitzmann, A (2006), *Information Hiding: Third International Workshop*, IH'99 : Dresden, German.

[12] Ponyton, C (2003), Digital Video and HD: Algorithms and Interfaces.

[13] Sachs, J. (1996). Digital Image Basics. Digital Light & Color.

[14] Wong, P, H. Memon (1999), *A watermarking for image integrity and ownership verification*, in Proc. Final Program and Proceedings of the IS&-T PICS 99, pp. 374-379, Savana, Ga, USA.

[15] Yoo, J, dkk (2013), *Template matching of occluded object under low PSNR*, Digital Signal Processing