

Analisis Perbandingan Histogram dan Kualitas Citra Pada *Image Steganografi* Menggunakan Metode *One Bit Least Significant Bit*

Aditya Aziz Fikhri¹, Hendrawaty²

^{1,2} Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

¹aditfreedom11@gmail.com

²waty.hendra@yahoo.com

Abstrak— Kemajuan teknologi informasi terutama pada *smartphone* Android sangat berkembang dengan cepat, memungkinkan penggunaanya saling berkirim data/informasi dengan mudah dan cepat contohnya dengan menggunakan *WhatsApp*, *E-Mail*, *Telegram*, dan *Facebook*. Data/informasi yang dikirim melalui internet masih rawan terhadap pencurian dan penyadapan. Oleh karena itu dibutuhkan cara untuk mengamankan data/informasi yang akan dikirim. *Steganografi* merupakan salah satu cara yang dapat digunakan untuk mengamankan data/informasi. *Steganografi* adalah teknik yang digunakan untuk menyembunyikan pesan kedalam suatu media seperti gambar, audio dan video. Paper ini akan membahas mengenai analisis perbandingan kualitas dari sebuah gambar sebelum dan setelah mengalami penyisipan pesan dengan menggunakan *Histogram* dan nilai *PSNR* yang dihasilkan dengan menggunakan metode *One Bit Least Significant Bit*.

Kata kunci— *One Bit Least Significant Bit*, *Histogram Analisis*, *Stego Image*, *Cover Image*, *Steganografi*.

Abstract— The advancement of information technology, especially on Android smartphones, is growing rapidly, allowing users to send data / information easily and quickly, for example by using *WhatsApp*, *E-Mail*, *Telegram*, and *Facebook*. Data / information sent via the internet is still prone to theft and tapping. Therefore we need a way to secure data / information that will be sent. *Steganography* is one way that can be used to secure data / information. *Steganography* is a technique used to hide messages into a medium such as images, audio and video. This paper will discuss the comparative analysis of the quality of an image before and after experiencing message insertion using the *Histogram* and *PSNR* values generated using the *One Bit Least Significant Bit* method.

Keywords— *One Bit Least Significant Bit*, *Histogram Analysis*, *Stego Image*, *Cover Image*, *Steganography*.

I. PENDAHULUAN

Informasi merupakan hal yang sangat penting dan berharga, dewasa ini informasi sering dipertukarkan melalui internet sehingga rentan terhadap pencurian informasi. Pencurian informasi tersebut dapat dilakukan dengan cara membobol server apabila informasi tersebut ada di dalam sebuah server. Banyak jenis informasi yang menggunakan pengamanan khusus seperti pin, *password*, dan data diri. Informasi tersebut tidak boleh diketahui oleh pihak lain karena dapat memberikan kerugian kepada pemilik informasi.

Dengan hadirnya *smartphone* android pada saat ini, Memungkinkan penggunaanya untuk berbagi informasi dengan mudah dan cepat melalui internet. Salah satu cara pengguna android berbagi informasi yaitu dengan media *chatting*, yakni mengirim pesan secara pribadi dari satu pengguna ke pengguna lainnya menggunakan aplikasi Android seperti *WhatsApp*, *Facebook*, *Telegram* dan *E-Mail*. Sebagai contoh yaitu memberi informasi yang sangat penting dan rahasia seperti *password* sebuah akun media sosial atau memberikan informasi pin sebuah brankas dengan media *chatting* tersebut. Tetapi kenyataan yang terjadi saat ini, isi sebuah *chat* pun dapat diketahui orang lain karena kemungkinan untuk disadap atau diretas.

Salah satu teknik yang dapat membantu mengamankan informasi yaitu *Steganografi*. Dengan menggunakan teknik *Steganografi*, pesan teks yang bersifat rahasia dapat disembunyikan ke dalam sebuah gambar sebelum dikirim melalui internet. Walaupun gambar yang telah disisipi pesan rahasia tersebut berhasil dicuri oleh pihak lain, tetapi pesan teks yang ada didalamnya tetap tidak dapat diketahui. Pihak pencuri informasi tidak akan menyadari ada pesan rahasia pada gambar, selama gambar tersebut tidak mengundang kecurigaan.

Paper ini bertujuan untuk menganalisa dengan cara membandingkan *stego image* dan *cover image* dengan menggunakan histogram dan nilai *PSNR* yang dihasilkan dengan menggunakan metode *one bit least significant bit* serta untuk menganalisa *cover image* dan *stego image* dari serangan salah satunya yaitu histogram attack.

Beberapa penelitian yang berhubungan dengan *steganografi* dengan metode *Least Significant Bit* sudah pernah dilakukan sebelumnya. Berikut penjelasan secara garis besar beberapa penelitian yang terkait dengan *steganografi* berbasis Android.

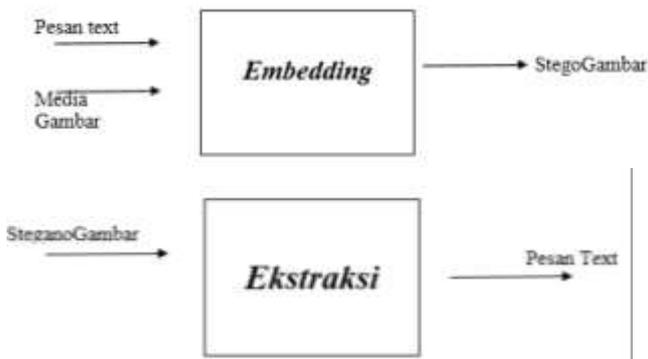
Pada tahun 2013, Lip Yee et al.[1] telah melakukan penelitian tentang analisis dari metode *Least Significant Bit* serta menjelaskan jenis-jenis dari metode *Least Significant Bit* salah satunya metode *One Bit Least Significant Bit* dan menjelaskan tentang menyisipkan data kedalam gambar secara sekuensial. Pada tahun 2013, Erwin et al.[2] telah melakukan penelitian tentang analisis *steganografi* antara penyisipan pesan secara sekuensial dan acak serta menentukan mana lebih baik dalam penyisipan data ke dalam gambar dengan metode *Least Significant Bit*. Pada tahun 2015, Arun et al.[3] telah melakukan penelitian tentang penggunaan *steganografi* algoritma *One Bit Least Significant Bit* dalam menyembunyikan pesan ke dalam citra grayscale namun dalam menyembunyikan pesan tersebut menggunakan kunci yang sama untuk menyisipkan dan mengekstraksi pesan rahasia dari gambar *stego*. Pada tahun 2015, Aditya et al.[4] telah melakukan penelitian tentang implementasi *steganografi* menggunakan *smartphone* Android dengan metode *One Bit Least Significant Bit* dengan menggunakan media *cover image* RGB 24-bit dan mengimplementasikannya pada sosial media. Pada tahun 2016, Michael et al.[5] telah melakukan penelitian tentang modifikasi terhadap algoritma *SLSB* untuk meningkatkan ketahanannya terhadap serangan yang mampu menghilangkan informasi didalamnya. Dimana Algoritma

yang termodifikasi ini mampu mendeteksi informasi pada gambar berformat BMP atau PNG yang dibalik dan juga memiliki tingkat keberhasilan sebesar 99% dalam penyisipan informasi.

A. Landasan Teori

1) *Steganography* : Steganografi berasal dari dua bahasa Yunani “*steganos*” yang artinya tersembunyi atau terselubung dan “*graphein*” yang artinya menulis. Steganografi dapat diartikan sebagai tulisan tersembunyi (*cover writing*) [6]. Dengan kata lain Steganografi adalah seni dan ilmu menulis atau menyembunyikan pesan dengan suatu cara sehingga selain *sender* dan *receiver*, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia [7].

Terdapat dua proses utama dalam Steganografi yaitu penyisipan (*embedding*) dan penguraian (*extraction*). Proses *embedding* yaitu proses untuk menyisipkan pesan ke dalam media *cover* sedangkan proses *extracting* yaitu proses menguraikan pesan pada media *stego*.



Gambar 1. Blok diagram proses penyisipan dan ekstraksi

Suatu metode yang terkenal dan sederhana pada steganografi adalah menyembunyikan data rahasia ke least significant bit (LSB) dari setiap pixel dalam sebuah gambar [8].

2) *One Bit LSB* : *One Bit Least Significant Bit* merupakan jenis dari metode LSB yang menyisipkan data pada satu bit LSB pada setiap pixelnya [1]. Terdapat 8 pixel pada gambar yang digunakan untuk menyisipkan 1 karakter saja dikarenakan 1 karakter memiliki 8 bit, warna yang berubah yaitu warna biru saja dengan jumlah 1 lebih tinggi atau 1 lebih rendah nilainya. Sehingga metode ini dapat membuat *Stego Image* terlihat lebih kecil serta lebih halus perubahannya saat setelah bit pesan disisipkan kedalam *Cover Image*.

R				G				B			
1	1	1	1	1	1	1	1	1	1	1	1

Gambar 2. Ilustrasi one bit LSB

Sebagai ilustrasi, Gambar 3 adalah contoh dari urutan bit 3 pixel pada *cover image* 24 bit.

R	G	B
(00100111	10100111	00100110)
(00100111	10100111	00100110)
(00100111	10100111	00100110)

Gambar 3. Bit Cover Image

Misalkan bit pesan yang akan disisipkan yaitu **110**, maka dengan menggunakan metode *One bit LSB* akan dihasilkan urutan bit pada *stego image* seperti yang tampak pada Gambar 4.

R	G	B
(00100111	10100111	00100110)
(00100111	10100111	00100111)
(00100111	10100111	00100111)

Gambar 4. Bit hasil penyisipan

Dapat dilihat pada Gambar 4, merupakan hasil dari penyisipan 1 byte data pada 3 pixel pada gambar menggunakan metode *One Bit Least Significant Bit*. Hasil pada proses penyisipan yaitu berupa *stego image* 24-bit. Maka demikian contoh sederhana dari penerapan bagaimana metode LSB ini untuk menggantikan bit yang kurang berarti dari setiap pixel pada gambar.

3) *Histogram* : Suatu histogram menyediakan informasi yang berkaitan dengan kontras dan distribusi intensitas keseluruhan citra [9]. Histogram citra digambarkan secara sederhana sebagai suatu bar grafik dari intensitas pixel-pixel. Intensitas pixel di-plot sepanjang sumbu x dan jumlah pemunculan untuk tiap intensitasnya direpresentasikan pada sumbu y. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan dan kontras dari sebuah citra. Karena itu histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif, meskipun begitu metode ini tidak selalu merupakan metode yang baik untuk peningkatan kualitas citra khususnya untuk citra warna di mana pada saat ekualisasi ketiga komponen R, G, B timbul distorsi warna dalam pemrosesannya.

4) *Mean Square Error* : Untuk mendapatkan nilai kualitas citra dilakukan dengan cara penilaian secara objektif dengan menggunakan besaran MSE dan PSNR kedua besaran tersebut membandingkan pixel-pixel pada posisi yang sama dari dua citra yang berlainan [10]. Secara matematis rata-rata kuadrat nilai kesalahan (*Mean Square Error*) antara citra asli dengan citra hasil pengolahan didapat sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x,y) - g(x,y)|^2 \tag{1}$$

5) *Peak Signal Noise Ratio* : Untuk nilai perbandingan antara harga maksimum warna pada citra hasil filtering dengan kuantitas gangguan (noise) didapat dengan PSNR, yang dinyatakan dalam satuan desibel (dB), Bercak (noise) yang dimaksud adalah akar rata-rata kuadrat nilai kesalahan (MSE) [10]. Secara matematis, nilai PSNR dapat dirumuskan sebagai berikut:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \tag{2}$$

II. METODOLOGI PENELITIAN

Penelitian ini menganalisa perbandingan *cover image* dan *stego image* yang dihasilkan oleh aplikasi Steganografi *One Bit Least Significant Bit* berbasis Android. Perbandingan dilakukan dengan membandingkan gambar sebelum dan sesudah proses penyisipan, histogram dari nilai R, G dan B sebelum dan sesudah proses penyisipan pesan kedalam gambar dengan tool Matlab R2010a serta membandingkan nilai PSNR dari gambar hasil proses penyisipan. Setelah melakukan ketiga perbandingan yang telah disebutkan, akan diambil kesimpulan dari penelitian yang dilakukan.

A. Hardware dan Software

Hardware yang digunakan dalam penelitian ini yaitu :

1. Notebook ASUS X200M dengan *harddisk* 500 GB, prosesor Intel(R) Celeron(R) CPU N285 1.86 GHZ, RAM 2GB(*Giga Byte*).
2. Smartphone Xiaomi Redmi 4A dengan versi android 6.1 Marshmallow, RAM 2GB (*Giga Byte*), Internal storage 16GB.

Adapun Software yang digunakan yaitu :

1. Aplikasi Steganografi *One Bit Least Significant Bit* berbasis Android.
2. Matlab R2010a

B. Perancangan Algoritma Proses Penyisipan

Proses penyisipan merupakan proses utama yang berguna untuk menyembunyikan atau menyisipkan pesan teks kedalam gambar RGB 24-bit. Adapun inputan dari proses penyisipan yaitu berupa gambar RGB 24-bit berformat JPG dan pesan teks. Output yang dihasilkan dari proses penyisipan yaitu gambar hasil penyisipan (*stego image*) RGB 24-bit. Perancangan algoritma proses penyisipan pesan kedalam *cover image* ditampilkan pada Tabel 1.

Tabel 1. Algoritma proses penyisipan

Algoritma 1. Proses Penyisipan	
Input	Cover Image RGB 24-bit (JPG), Pesan Teks
Output	Stego Image RGB 24-bit (PNG)
Step 1.	Ubah cover image dalam bentuk biner
Step 2.	Ubah pesan teks kedalam bentuk biner
Step 3.	Mengambil panjang pesan kedalam integer
Step 4.	Konversi panjang pesan kedalam biner
Step 5.	Penyisipan panjang pesan dari pixel (0,0)-(0,19)
Step 6.	Penyisipan pesan teks dari pixel (0,20) sampai dengan batas panjang pesan*8
Step 7.	Ubah bit hasil penyisipan kedalam integer
Step 8.	Ubah integer kedalam Pixel

Pada Tabel 1 *user* menginput gambar RGB 24-bit sebagai *cover image* yang kemudian dinormalisasi menjadi ukuran 500x500 pixel. Selanjutnya sistem menghitung nilai

dimensinya dan dirubah kedalam bentuk biner yang mana bit terakhir pada warna birunya akan disisipkan dengan cara mengganti bit LSB dari pesan yang diinputkan.

Selanjutnya yaitu menginput pesan teks yang akan disembunyikan dengan syarat ukuran pesan teks tidak boleh lebih besar daripada *cover image*. Kemudian sistem melakukan pembacaan nilai desimal dari jumlah karakter panjang pesan. Setelah dibaca dan diperoleh informasi jumlah karakter panjang pesan, maka informasi panjang pesan tersebut dirubah kedalam bentuk bit dan pesan tersebut disisipkan pada 20 pixel pertama yang dimulai dari pixel (0,0) sampai dengan pixel(0,19) dengan menggunakan metode *One Bit Least Significant Bit*. Setelah disisipkan informasi panjang pesan pada 20 pixel pertama pada gambar, maka sistem akan menyisipkan bit pesan pada bit ke (0,20) hingga seterusnya dengan cara nilai decimal dari panjang pesan dikalikan dengan 8.

Setelah dilakukan penyisipan pesan maka sistem akan mengubah hasil bit gambar yang telah melalui proses penyisipan pesan kedalam bentuk pixel dengan cara menyusun kembali nilai RGB nya kembali sehingga menghasilkan *stego image* dan disimpan pada penyimpanan pengguna.

C. Perancangan Algoritma Proses Ekstraksi

Proses ekstraksi merupakan proses yang berguna untuk mengekstraksi pesan teks yang disembunyikan dari *stego image* RGB 24-bit. Inputan dari proses ekstraksi pesan yaitu gambar hasil penyisipan (*stego image*) RGB 24-bit berformat PNG. Output yang dihasilkan dari proses ekstraksi yaitu berupa pesan teks hasil penyisipan. Perancangan algoritma proses ekstraksi pesan teks dari *stego image* RGB 24-bit ditampilkan pada Tabel 2.

Tabel 2. Algoritma proses ekstraksi

Algoritma 2. Proses Ekstraksi	
Input	Stego Image RGB 24-bit (PNG)
Output	Pesan Teks
Step 1.	Ubah stego image dalam bentuk biner
Step 2.	Ambil panjang karakter dari pixel (0,0)-(0,19)
Step 3.	Ekstraksi bit pesan dari pixel (0,20) sampai dengan panjang pesan*8
Step 4.	Konversi bit pesan yang diekstraksi kedalam ASCII
Step 5.	Konversi ASCII ke Char

Pada Tabel 2 dijelaskan bahwa proses ekstraksi dilakukan dengan menginput *stego image* terlebih dahulu. Setelah itu *stego image* tersebut dirubah kedalam bilangan biner, selanjutnya yaitu mengambil informasi panjang karakter pesan teks dengan cara mengekstraksi bit panjang karakter pesan dari pixel (0,0) hingga pixel (0,19).

Setelah didapatkan deretan bit panjang pesan, maka selanjutnya yaitu mengkonversi bit panjang pesan kedalam bentuk desimal sehingga nantinya didapatkan informasi panjang karakter pesan dalam bentuk bilangan bulat. Setelah

didapatkan informasi panjang pesan, maka selanjutnya yaitu mengekstraksi bit pesan dengan menggunakan metode *One Bit Least Significant Bit* dari pixel ke (0,20) hingga batasnya yang mana batas dari ekstraksi pesan tersebut yaitu nilai desimal dari panjang karakter pesan yang telah didapatkan lalu dikalikan dengan 8 karena 1 karakter bernilai 8 bit.

Setelah dapat deretan bit pesan yang diekstraksi, maka selanjutnya yaitu mengkonversi bilangan bit tersebut kedalam bentuk bilangan bulat sehingga nantinya bilangan tersebut dikonversikan kedalam bentuk karakter.

III. HASIL DAN PEMBAHASAN

Pada tahap ini dilakukan pembahasan hasil pengujian yang terdiri dari perbandingan gambar stego (*stego image*) hasil penyisipan dengan gambar asli (sebelum penyisipan), Histogram R,G dan B sebelum dan sesudah proses penyisipan dan tabel nilai PSNR dari *stego image*. Terdapat 5 contoh gambar yang diuji pada penelitian ini dengan menggunakan gambar dengan berdimensi 500x500 pixel, Tabel 3 menunjukkan perbandingan gambar sebelum dan sesudah proses penyisipan dengan ukuran kapasitas penyisipan maksimum.

Tabel 3. Perbandingan sebelum dan sesudah penyisipan

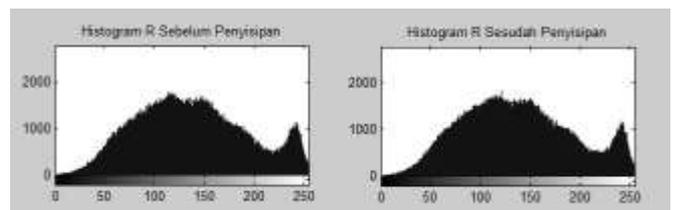
No	Gambar Asli	Gambar Stego
I	 Baboon.jpg 500x500 pixel 139 kb	 Baboon.png 500x500 pixel 542 kb
II	 Paprika.jpg 500x500 pixel 75.3 kb	 Paprika.png 500x500 pixel 334 kb
III		

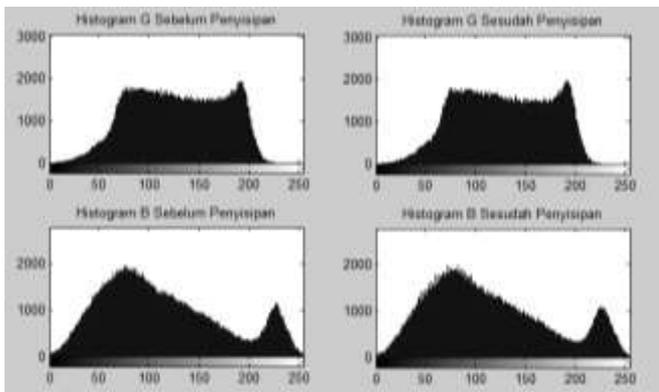
	Azul.jpg 500x500 pixel 86.3 kb	Azul.png 500x500 pixel 344 kb
IV	 Bliss.jpg 500x500 pixel 83.8 kb	 Bliss.png 500x500 pixel 389 kb
V	 Tulip.jpg 500x500 pixel 95.2 kb	 Tulip.png 500x500 pixel 243 kb

Pada tabel 3 dapat dilihat bahwa gambar sebelum dan sesudah penyisipan pesan dengan pesan yang sama dengan kapasitas penyisipan maksimum menggunakan metode *One Bit Least Significant Bit* berhasil disisipkan, gambar yang dihasilkan yaitu berformat PNG dengan dimensi 500x500 pixel. Perubahan yang terjadi dari kelima gambar yaitu pada ukuran file yang meningkat dari gambar asli (sebelum penyisipan) namun tidak ada perubahan warna secara kasat mata.

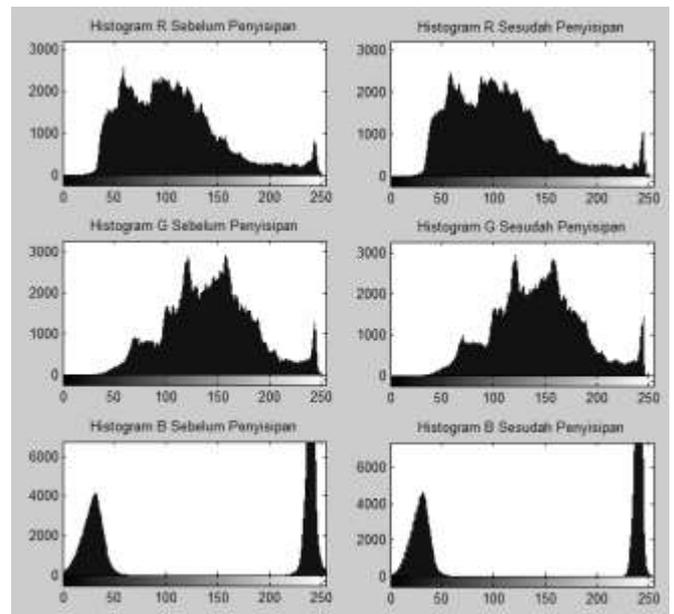
A. Analisis Perbandingan Histogram

Analisis hasil dengan menggunakan histogram bertujuan untuk melihat perubahan gambar yang dilihat dari bentuk grafik pada histogram sebelum dan sesudah proses penyisipan yang dihasilkan sehingga dapat melihat perubahan gambar secara rinci. Gambar 5 sampai dengan Gambar 9 menunjukkan perbandingan histogram R, G dan B pada setiap gambar sebelum proses penyisipan.

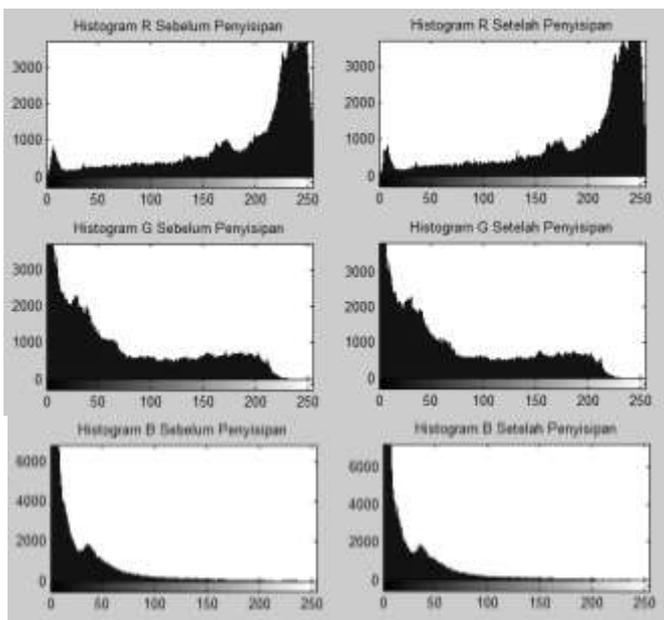




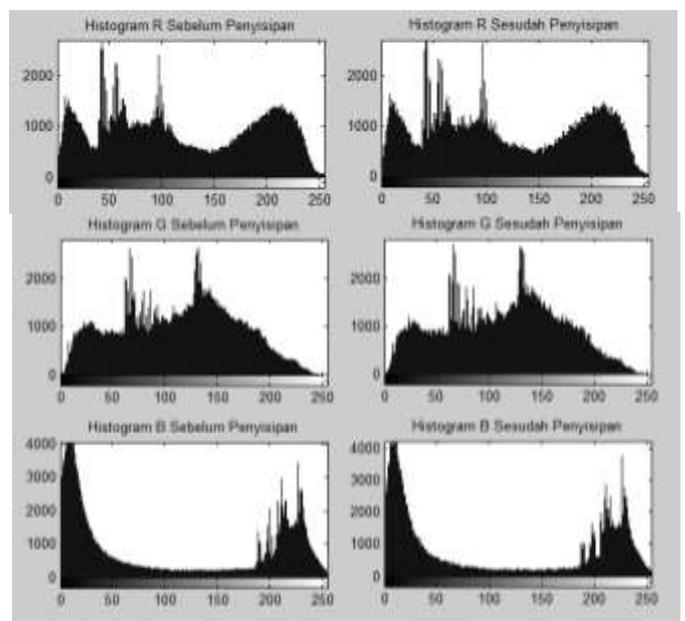
Gambar 5. Analisis histogram dari Baboon.jpg



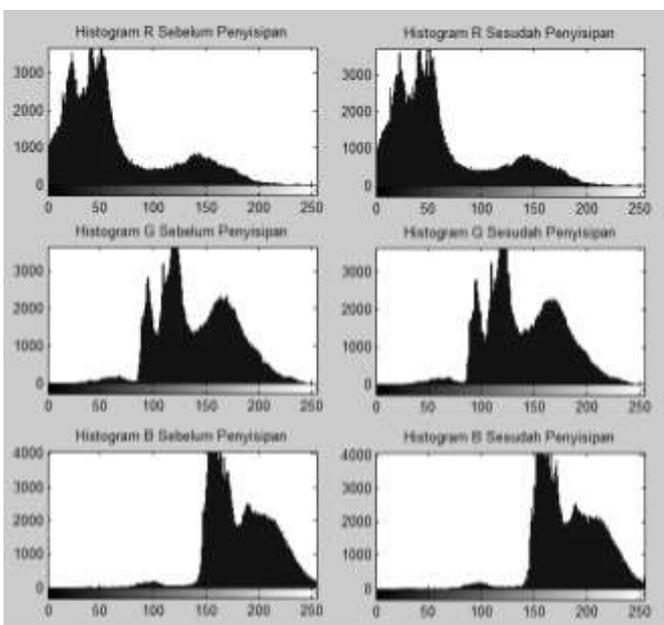
Gambar 8. Analisis histogram dari Bliss.jpg



Gambar 6. Analisis histogram dari Paprika.jpg



Gambar 9. Analisis histogram dari Tulip.jpg



Gambar 7. Analisis histogram dari Azul.jpg

Pada Gambar 5 sampai Gambar 9 dapat dilihat bahwa terdapat histogram R,G dan B sebelum dan sesudah proses penyisipan. Perubahan histogram dari R dan G pada gambar stego hasil penyisipan sedikit ada perubahan bentuk grafiknya dikarenakan proses kompres gambar dari JPG ke PNG namun masih mirip dengan bentuk aslinya(sebelum penyisipan).. Namun pada histogram B terdapat perubahan bentuk yang terlihat tidak signifikan dari grafik aslinya. Hal ini menunjukkan bahwa perubahan pada gambar stego hasil penyisipan pesan menggunakan metode *one bit least significant bit* sangat kecil karena bit yang diganti pada saat proses penyisipan pesan yaitu satu bit saja pada sebuah pixel.

B. Analisis Perbandingan Nilai PSNR

Analisis perbandingan nilai PSNR dilakukan sebagai salah satu hal penting dalam membandingkan perubahan gambar

sebelum dan sesudah proses penyisipan selain membandingkan gambar langsung dengan membandingkan secara histogram. Nilai PSNR sesudah proses penyisipan (*stego image*) dapat dilihat pada gambar Tabel 4.

Tabel 4. Nilai kapasitas penyisipan dan PSNR

Nama Gambar	<i>One Bit Least Significant Bit</i>		
	Kapasitas Penyisipan / <i>Embedding Capacity</i> (bit)	MSE	PSNR (db)
Baboon	250000	4.4444	41.6526
Paprika	250000	3.6364	42.5241
Azul	250000	5	41.1411
Bliss	250000	4.4444	41.6526
Tulips	250000	4.4444	41.6526

Dari kelima hasil penelitian dapat dilihat bahwa gambar dengan kapasitas penyisipan sebanyak 250000 bit menghasilkan nilai MSE (*Mean Square Error*) yang berbeda – beda dan nilai PSNR (*Peak Signal Noise Ratio*) diatas 41 yang mana menunjukkan kualitas citra hasil penyisipan (*stego image*) yaitu baik yang mana dikatakan baik memiliki nilai diatas 30 db, sedangkan kualitas citra yang dikatakan tidak baik berada dibawah 30 db.

IV. KESIMPULAN

Dari hasil yang didapatkan, dapat diambil kesimpulan bahwa dengan menggunakan metode *One Bit Least Significant Bit* perubahan pada warna gambar tidak berubah secara kasat mata. Analisis hasil dengan menggunakan

histogram juga terbukti bahwa perubahan bentuk grafik dengan menggunakan metode *One Bit Least Significant Bit* berubah secara signifikan saja (hampir tidak terlihat) pada bagian yang disisipi (bit warna B), hanya saja bit R dan G berubah secara signifikan dikarenakan kompresi file kedalam PNG. Pada analisis nilai PSNR juga terbukti bahwa nilai yang dihasilkan juga berada diatas 41 dimana merupakan kualitas citra yaitu baik dan dikatakan baik apabila nilai PSNR diatas 30 db.

REFERENSI

- [1] Por, L. Y., Beh, D., Ang, T. F., & Ong, S. Y. (2013). An Enhanced Mechanism for Image Steganography Using Sequential Colour Cycle Algorithm . *The International Arab Journal of Information Technology*.
- [2] Hidayat, E. Y., & Hastuti, K. (2013). Analisis Steganografi Metode Least Significant Bit (LSB) Dengan Penyisipan Sekuensial Dan Acak Secara Kuantitatif Dan Visual. *Techno.COM*.
- [3] Singh, A. K., Singh, J., & Singh, H. V. (Vol. 5 Issue 1 January 2015). Steganography in Images Using LSB Technique . *International Journal of Latest Trends in Engineering and Technology (IJLTET)*.
- [4] Fikhri, A. A., & Hendrawaty. (2018). Implementasi Steganografi Text To Image Menggunakan Metode One Bit Least Significant Bit Berbasis Android. *Jurnal Infomedia*.
- [5] Saputra, M. S. (2016). Implementasi Algoritma Selected Least Significant Bit yang Dimodifikasi untuk Menyimpan Informasi pada Gambar. *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana*.
- [6] Harjo, B. T., Kapriati, M., & Susanto, D. A. (2016). Aplikasi Steganografi Menggunakan LSB (Least Significant Bit) dan Enkripsi Triple Des Menggunakan Bahasa Pemrograman C#. *JURNAL SISFOTEK GLOBAL*.
- [7] Martono, & Irawan. (2013). Penggunaan Steganografi dengan Metode End of File (EOF) pada Digital Watermarking. *Jurnal TICOM*.
- [8] Deepesh Rawat, Vijaya Bhandar, Steganography Technique for Hiding Text Information in Color Image using Improved LSB Method. *International Journal of Computer Applications (0975 – 8887) Volume 67– No.1, April 2013*
- [9] Murinto, Putra, W. P., & Handayaningsih, S. (Vol 2, No. 2, Juli 2008). Analisis Perbandingan Histogram Equalization Dan Model Logarithmic Image Processing (Lip) Untuk Image Enhancement. *Jurnal Informatika* .
- [10] Fuad, N., & Melita, Y. (Vol.4 No.2 September 2012). Analisa Hasil Perbandingan Metode Low-Pass Filter Dengan Median Filter Untuk Optimalisasi Kualitas Citra Digital. *Jurnal Teknika* .