ANALISIS PSNR PADA STEGANOGRAFI LEAST SIGNIFICANT BIT DENGAN PESAN TERENKRIPSI ADVANCED ENCRIPTION SYSTEM

Rimbun Siringoringo

Manajemen Informatika, Fakultas Ekonomi, Universitas Methodist Indonesia
Jl. Hang Tuah No 4. Medan
ringorbnsrg@gmail.com

Abstract

Steganography is the art of hiding data on a medium other. Today, steganography is used to secure data by hiding it in other media so that the existence of the message is unintelligible. LSB steagnografi is the simplest method among other methods. LSB steganographic techniques susceptible to steganalisis. In this study the authors combine this with the LSB steganography method by cryptographic techniques. Cryptographic algorithm used is AES. Messages will be inserted first encrypted using the AES method. There are five image dataset being tested. Overall proficiency level datasets have different capacities. There are five types of hidden text with a capacity of 1K, 5K, 10K, 15K and 20K. Testing the quality of the image before and after the embedding process was conducted by PSNR and MD. From the test is known that embedding the message affects the pixel values at specific coordinates on the cover image, the more the characters are pasted on the cover image PSNR value of its smaller, it indicates that the image quality is getting lower and berdanding PSNR value proportional to the value of MD image.

Keywords *: steganography; least signigicant bit; advanced encription system*

I. Pendahuluan

Keamanan dan kerahasiaan merupakan dua aspek penting dalam komunikasi dan jaringan. Ke dua aspek tersebut sangat dibutuhkan dalam proses pertukaran informasi melalui dunia maya. Tanpa adanya jaminan keamanan, orang lain dapat dengan mudah mendapatkan informasi yang dikirimkan melalui internet [1]. Berbagai macam teknik keamanan telah dikembangkan untuk melindungi dan menjaga kerahasiaan informasi agar terhindar dari pihak yang tidak berhak, salah satunya yaitu teknik kriptografi. Kriptografi adalah suatu ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan pesan ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Pesan yang telah disandikan tersebut tentu saja akan memiliki bentuk yang sangat berbeda dari pesan yang umum sehingga hal tersebut akan menarik perhatian pihak ketiga yang curiga terhadap pesan hasil keluaran kriptografi tersebut. Algoritma kriptografi klasik memiliki banyak kelemahan jika dibandingakan dengan algoritma kriptografi modern. Kriptografi modern tidak beroperasi dalam modus karakter alfabet seperti pada algoritma kriptografi klasik. Kriptografi modern beroperasi pada mode bit, yang berarti semua data dan informasi (kunci, plainteks, maupun cipherteks) dinyatakan dalam rangkaian (string) bit biner, 0 dan 1 [2]. Salah satu teknik kriptografi modern adalah algoritma *Advanced Encription Standard* (AES).

Teknik menjaga kerahasiaan pesan tidak hanya menggunakan kriptografi. Teknik lain yang dapat digunakan yaitu steganografi [3]. Steganografi adalah seni dan ilmu untuk menyembunyikan pesan rahasia (teks, gambar, video, suara) di dalam pesan lain(teks, gambar, video, suara). sehingga keberadaan pesan rahasia tersebut *unintelligible* atau tidak dapat diketahui eksistensinya (Stalling, . Berbeda dengan kriptografi yang merahasiakan makna pesan namun keberadaan pesan tetap ada, steganografi merahasiakan dengan menutupi atau menyembunyikan pesan .

Salah satu metode steganografi citra digital adalah *Least Significant Bit* (LSB), dengan teknik penyembunyian pesan pada lokasi bit terendah dalam citra digital. Pesan dikonversi ke dalam bentuk bit biner dan disembunyikan pada citra digital dengan metode LSB. Implementasi metode LSB tanpa dilengkapi dengan sistem keamanan berpeluang untuk dapat dibongkar dengan mudah melalui teknik pemecahan analisis frekuensi dengan membaca bit terendah. Kualitas citra hasil penyisipan dengan metode LSB lebih baik dibandingkan dengan menggunakan metode yang lain, namun metode LSB sudah sangat umum dan mudah dipecahkan [4]. Dengan demikian keamanan dari metode ini sudah tidak baik lagi. Pada penelitian ini penulis menggabungkan steganografi metode LSB ini dengan dengan kriptografi metode AES, dimana pesan yang akan disisipkan terlebih dahulu dienkripsi dengan menggunakan metode AES. Hasil dari enkripsi tersebut kemudian disisipkan ke media citra digital. Dengan penggabungan kedua metode ini, maka pesan akan sulit untuk dipecahkan, karena memiliki dua tingkat keamanan.

2. Tinjauan Pustaka

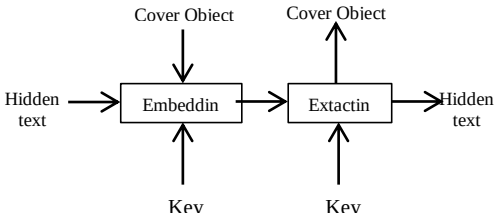
A. Steganografi digital

Steganografi digital (*digital steganograhpy*) menggunakan media digital sebagai wadah penampung data rahasia yang akan disembunyikan. Media digital tersebut misalnya citra (*image*), suara (audio), teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video.

B. Proses steganografi

Proses steganografi terdiri atas dua bagian besar yakni proses *encoding/embedding* dan

decoding/extracting. Proses *encoding* adalah penyisipan *hiddentext* ke dalam *covertext*. Output dari proses *encoding* disebut dengan *stegotext*. Selanjutnya proses *decoding* adalah penguraian (*extracting*) *stegotext* menjadi *coverimage* dan *hiddentext*. Proses steganografi diperlihatkan pada gambar 1 [5]



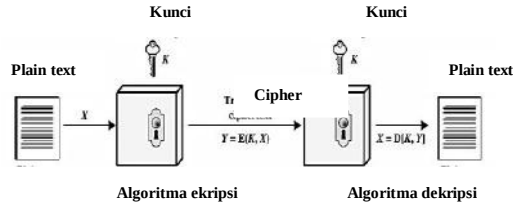
Gambar 1. Proses steganografi

C. Least Significant Bit

Pendekatan paling sederhana untuk menyembunyikan data dalam file citra disebut penyisipan *Least Significant Bit* (LSB). Metode *Least significant bit* (LSB) adalah pendekatan yang umum untuk menanamkan informasi dalam media citra. Dengan metode LSB, sebagian atau seluruh dari *byte* dalam sebuah gambar diubah menjadi sebuah bit dari pesan rahasia. Dalam metode yang ada, dibutuhkan representasi biner dari data yang akan disembunyikan dengan metode LSB

D. Kriptografi

Kriptografi memiliki dua proses utama, yaitu enkripsi (*encryption*) dan dekripsi (*decryption*). Enkripsi adalah proses penyandian *plaintext* menjadi *cipherteks*. Dekripsi adalah proses mengembalikan *ciphertext* menjadi *plaintext* semula. Enkripsi dan dekripsi membutuhkan kunci (*key)* sebagai parameter yang digunakan untuk transformasi. Gambar 2 memperlihatkan skema enkripsi dan dekripsi dengan menggunakan kunci [6]



Gambar 2. Proses Kriptografi

E. Algoritma AES

Algoritma AES mempunyai 3 parameter yaitu :

1. *Plainteks* adalah array yang berukuran 16 byte, yang berisi data masukan.
2. *Cipherteks* adalah array yang berukuran 16 byte, yang berisi hasil enkripsi. A-2
3. *Key* adalah array yang berukuran 16 byte, yang berisi kunci *ciphering* (*cipher key*).

1) Algoritma Enkripsi

Medeskripsikan Algoritma enkripsi AES [7] dapat dijabarkan sebagai berikut ini :

1. **AddRoundKey** : melakukan X-OR antara state awal (*plainteks*) dengan *cipher key*. Tahap ini disebut juga *initial round.*

2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. **SubByte** : substitusi byte dengan menggunakan tabel substitusi (S-box).
   b. **ShiftRow** : pergeseran baris-baris array state secara *wrapping*.
   c. **MixColumn** : mengacak data pada masing-masing kolom array state.
   d. **AddRoundKey** : melakukan operasi X-OR antara state sekarang dengan *round key.*

3. **Final round** : proses untuk putaran terakhir:
   a. SubByte.
   b. ShiftRow.
   c. AddRoundKey

2) Algoritma Dekripsi

Sedangkan algoritma dekripsi AES dapat dijabarkan sebagai berikut ini :

1. **AddRoundKey** : melakukan X-or antara state awal (cipherteks) dengan cipher key. Tahap ini disebut juga initial round.
2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah :
   a. **InvShiftRow** : pergeseran baris-baris array state secara wrapping.
   b. **InvSubByte** : substitusi byte dengan menggunakan tabel substitusi Inverse S-box.
   c. **AddRoundKey** : melakukan operasi X-OR antara state sekarang dengan round key.
   d. **InvMixColumn** : mengacak data pada masing-masing kolom array state.
3. Final round : proses untuk putaran terakhir:
   a. InvShiftRow; b) Inv SubByte. C) AddRoundKey

F. Peak Signal to Noise Ratio (PSNR)

Menurut [8], *Peak Signal to Noise Ratio* adalah metode yang digunakan dalam penelitian ini untuk mengukur perbandingan kualitas citra antara sebelum dan setelah proses steganografi. Semakin besar nilai PSNR mengindikasikan bahwa kualitas sebuah gambar baik dan sebaliknya. Nilai PSNR yang baik meminimalkan kemungkinan sebuah pesan tersembunyi untuk dideteksi oleh mata manusia. PSNR dapat ditentukan dengan persamaan berikut :

$$PSNR = 10\ log_{10}\left(\frac{C^2_{max}}{MSE}\right) \qquad (1)$$

dimana :
PSNR : nilai PSNR citra dalam dB
C^2_{max} : nilai maksimum piksel
MSE : nilai MSE

G. Mean Square Error (MSE)

Sebelum menentukan PSNR, terlebih dahulu ditentukan *Mean Square Error* (MSE) dan C^2_{max}. MSE dapat ditentukan dengan persamaan berikut ini :

$$MSE = \frac{1}{MN} \sum_{X=1}^{M} \sum_{y=1}^{N} (S_{xy} - C_{xy})^2 \quad (2)$$

dimana :
MSE : *Mean Square Error* dari citra
M : panjang citra dalam pixel
N : lebar citra dalam pixel
x,y : koordinat masing-masing pixel
S : nilai bit citra pada koordinat x,y
C : nilai derajat keabuan citra pada koordinat x,y

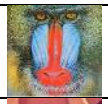
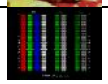
3. Metodologi Penelitian

A. Instrumen Penelitian

1) Dataset citra

Pada penelitian ini, pengujian dilakukan dengan menggunakan *standard dataset image* berupa file citra dengan format **png** 24 bit yang telah distandarisasi oleh SIPI (*Signal and Image Processing Institute*) *laboratory* sebagaimana ditampilkan pada tabel 1 berikut ini.

Tabel 1. Dataset Citra

Cover Image	***Preview***	***Kapasitas (byte)***	***Pixel***
baboon.png (CI-1)		637.192	512 x 512
lenna.png (CI-2)		473.831	512 x 512
peppers.png (CI-3)		538.749	512 x 512
gamma.png (CI-4)		41.394	990 x 768
house.png (CI-5)		441.032	440 X 600

2) Hidden text

*Hidden tex*t yang digunakan dalam penelitian ini terdiri dari lima jenis file teks yang memiliki kapasitas yang berbeda yaitu 1K, 5K, 10K,15K dan 20K.

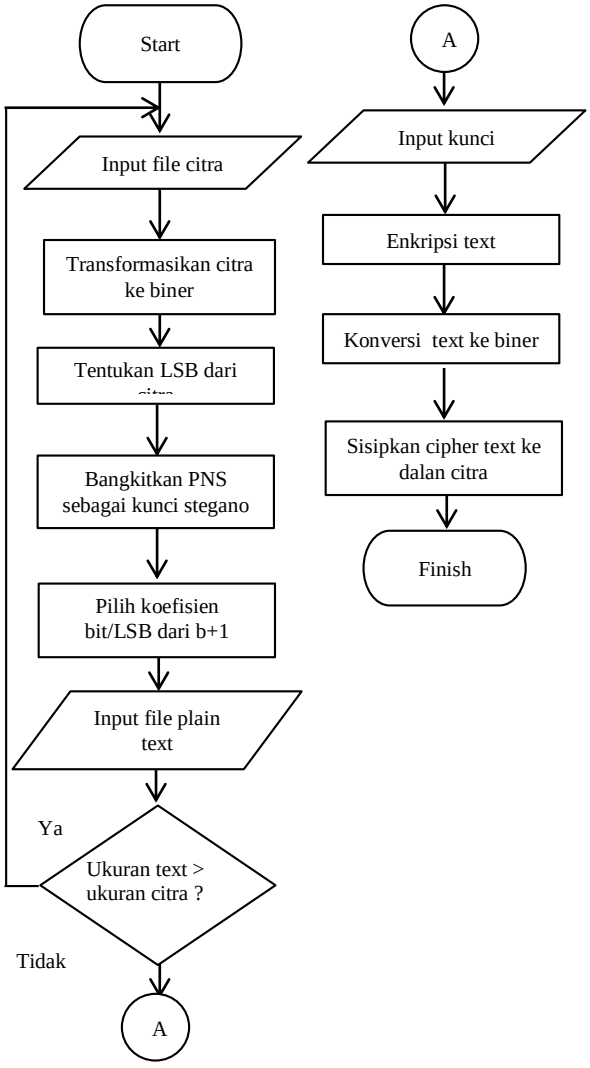
Tabel 2. File Hidden Text

Hiden text	***Kapasitas (KB)***
FileUji1.txt	1K
FileUji2.txt	5K
FileUji3.txt	10K
FileUji4.txt	15K
FileUji5.txt	20K

B. Algoritma Sistem

1) Algoritma Embedding

Proses penyisipan pesan dilakukan dengan menyandikan pesan rahasia atau *plain text* menggunakan algoritma enkripsi AES menjadi bentuk yang tidak dapat dipahami maknanya atau *cipher text*, setelah itu disisipkan pada media atau *cover text* berupa file citra menggunakan metode LSB. Hasil dari proses penyisipan adalah file gambar bitmap 24 bit yang disebut dengan *stego text.*

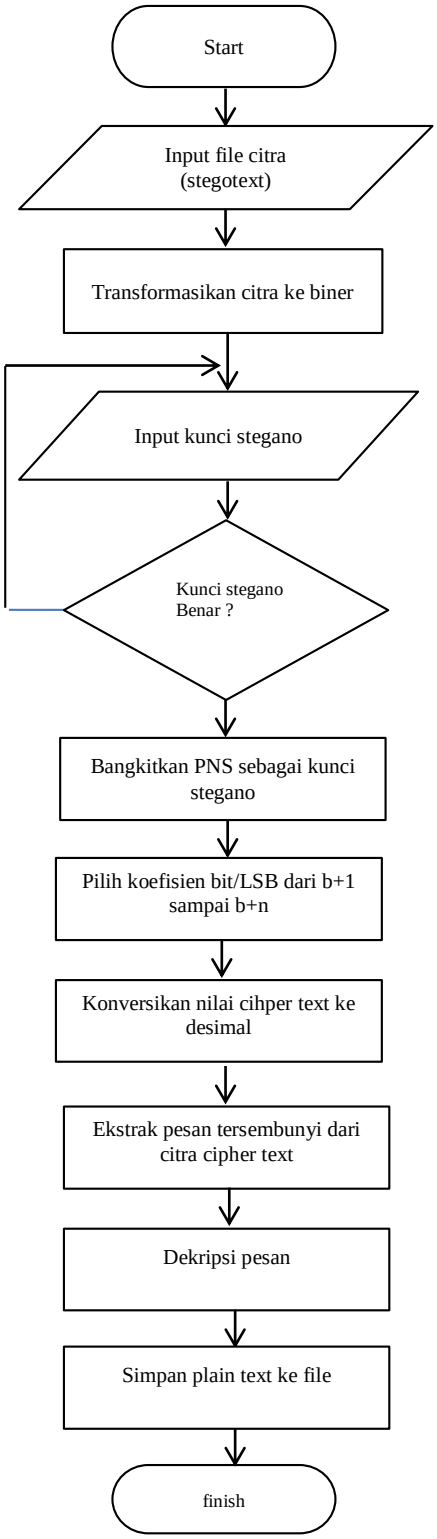


Gambar 3. Algoritma Encoding

2) Algoritma Decoding

Proses penguraian pesan dilakukan dengan mengambil *cipher text* dari *stego text* dan mengubah *cipher text* menjadi *plain text* dengan menggunakan algoritma dekripsi AES. Gambar 4

adalah *flowchart* proses penguraian atau ekstraksi pesan tersembunyi dari *cover image*



Gambar 4. Algoritma Decoding

C. *Algoritma PSNR*

PSNR merupakan metode pengujian yang digunakan untuk mengukur kualitas citra sebelum dan sesudah proses *embedding*. Algoritma PSNR yang diterapkan pada penelitian ini disajikan pada gambar 5 berikut:

```
File file1 = new File();
File file2 = new File();

final int size = citra1.getTinggi()
* citra1.getLebar();
for (int i = 0; i < citra1.getLebar(); i++) {
for (int j = 0; j < citra1.getTinggi(); j++) {
   final Warna warna1 =
   new  Warna(citra1.getRGB(i, j));
   final Warna warna2 =
   new Warna(citra2.getRGB(i, j));
   final double distance
   getWarnaDistance(warna1, warna2);
   totalJarak += Jarak;
if (Jarak > maxJarak) {
      maxJarak = Jarak;
      maxX = i; maxY = j;
      }
   final int merahDiff =
      warna1.getMerah()–
      warna2.getMerah();

      if (merahDiff > maxMerah) {
      maxMerah = merahDiff;
      worstMerahX = i; worstMerahY = j;
   }
   final int greenDiff =
      warna1.getHijau()-
      warna2.getHijau();
   if (HijauDiff > maxHijau) {
      maxHijau = HijauDiff;
      worstHijauX = i;worstHijauY = j;  }
   final int BiruDiff =
      warna1.getBiru() –
      warna2.getBiru();
   if (BiruDiff > maxBiru) {
      maxBiru = BiruDiff;
      worstBiruX = i;worstBiruY = j;  }
   totalMerah += merahDiff * merahDiff;
   totalHijau += HijauDiff * HijauDiff;
   totalBiru += BiruDiff * BiruDiff;  }
}
float meanSquamerahError =
   (totalMerah + totalHijau + totalBiru) /
   (citra1.getLebar() *
   citra1.getTinggi() * 3);

double peakSignalToNoiseRatio =
    10 * StrictMath.log10((255 * 255) /
   meanSquamerahError);   }
}
```

Gambar 5. Algoritma PSNR

4. Hasil Dan Pembahasan

A. *Interface aplikasi*

Berikut ini adalah interface aplikasi proses steganografi dan kriptografi pada penelitian ini.

a) Interface encoding

b) Interface encoding

c) Interface PSNR

Gambar 5. Interface System

B. Pengujian Enkripsi

Berikut ini adalah pasangan plain text dan cipher text yang diperoleh dengan algoritma AES

Plain text :

Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan dengan cara menyamarkannya menjadi bentuk tersandi yang tidak mempunyai makna. Bentuk tersandi ini hanya dapat dibaca oleh pihak yang berhak membacanya.

Gambar 4. Plain text

Cipher text :

À•ðSê @8__E„%Ÿì·&²°°UÞ™ Œ<Xh„Üªš/$_ü_¤□<Ã»€□,¦_èÊsÎeh“±“k1îU˜òGÛu_Ì”µÓ ;Z@¦U__/š8ú³B÷__ö¼Ê³ƒÓ□šÕ3ÝJ__jsám”†¦ÊÕÌ_^9C4ì _ßI°_¿e¼©µ)êYa‹U]»²Hh§B5l_ Esk1’f– ´4ÐóÙÑ_þ¬fljt‘é<?¡à¡!¨óEwG|_Š¯„E#ÃyzË`*ý,»Øß__˜ê •v_J:

Gambar 5. Cipher text

C. Perbandingan Kapasitas File Enkripsi

File teks yang digunakan sebagai *hidden text* terdiri dari lima file teks dengan kapasitas yang berbeda. Sebelum di sisipkan, ke lima file tersebut dienkripsi terlebih dahulu. File hasil enkripsi terdiri dari lima file dengan ekstensi **aes**.

Tabel 2. Perbandingan Kapasitas File Teks

plain text	**Kapasitas**	**Cipher text**	**kapasitas**
FileUji1.txt	1K	FileUji1.aes	1K
FileUji2.txt	5K	FileUji2.aes	5K
FileUji3.txt	10K	FileUji3.aes	10K
FileUji4.txt	15K	FileUji4.aes	15K
FileUji5.txt	20K	FileUji5.aes	20K

Pada tabel di atas ditampilkan bahwa kapasitas file *plain text* tidak mengalami perubahan dengan file *cipher text*

D. Pengujian Steganografi

1) Perubahan Nilai Pixel Citra

Penyisipan *hidden text* ke dalam *cover text* mempengaruhi nilai pixel tertentu pada *cover text*. Secara kasat mata perubahan tersebut tidak dapat dideteksi oleh mata manusia, tetapi dengan analisis pixel perubagan tersebut dapat di amati. Pada tabel ditampilkan perubahan pixel pada citra **house.png** setelah embedding hidden text (**fileUji1.txt**).

Tabel 3. Perubahan Nilai Pixel Citra

Sebelum				***Sesudah***			
(X,Y)	**R**	**G**	**B**	**(X,Y)**	**R**	**G**	**B**
(0, 0),	255,	255,	255	(0, 0),	254,	254,	254
(87, 0),	255,	255,	255	(87, 0),	**254**,	255,	255
(88, 0),	255,	255,	255	(88, 0),	**254**,	**254**,	255
(89, 0),	255,	255,	255	(89, 0),	**254**,	254,	**254**
(90, 0),	255,	255,	255	(90, 0),	255,	255,	255
(91, 0),	255,	255,	255	(91, 0),	**254, 254, 254**		
(92, 0),	255,	255,	255	(92, 0),	**254**,	255,	**254**
(93, 0),	255,	255,	255	(93, 0),	255,	**254**,	**254**
(94, 0),	255,	255,	255	(94, 0),	**254**,	255,	255
(95, 0),	255,	255,	255	(95, 0),	254,	255,	**254**
(96, 0),	255,	255,	255	(96, 0),	255,	**254**,	**254**
(97, 0),	255,	255,	255	(97, 0),	255,	**254**,	255
(98, 0),	255,	255,	255	(98, 0),	255,	255,	**254**
(99, 0),	255,	255,	255	(99, 0),	**254**,	255,	255
(100, 0),	255,	255,	255	(100, 0),	255,	**254**,	**254**
(439, 599),	34,	64,	0	(439, 599),	34,	64,	0

Pada tabel 3 di atas ditampilkan bahwa citra pada koordinat (91, 0) memiliki nilai RGB 255, 255, 255, setelah proses embedding, nilai pixel pada koordinat tersebut mengalami penurunan menjadi 254, 254, 254.

2) Perubahan kapasitas

Proses penyisipan *hidden text* pada *cover image* mempengaruhi kapasitas *stego image*. Semakin besar kapasitas *hidden text* yang disisipkan maka

semakin besar pula kapasitas file *stego image*. Pada tabel berikut ditampilkan kapasitas file citra dan

Tabel 4. Perbadingan Kapasitas File Citra

Nama Citra	***byte***	***Ukuran citra akhir (byte)***				
		F1	***F2***	***F3***	***F4***	***F5***
Baboon.png	637,192	753,572	753,574	753,591	753,596	753,616
Lenna.png	473,831	739,305	739,345	739,425	739,434	739,560
Pepper.png	538,749	731,797	731,809	731,972	732,228	732,589
House.png	441,032	598,320	603,370	608,064	611,166	613,407

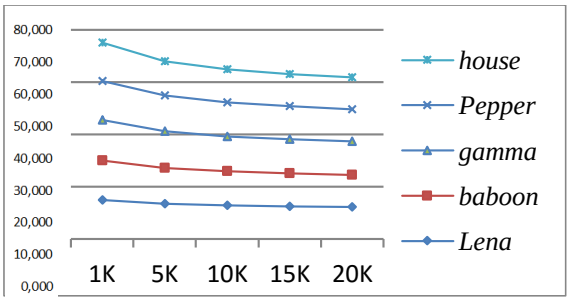
3) Pengujian PSNR citra

Penyisipan *hidden text* ke dalam *cover image* memiliki pengaruh terhadap kualitas citra hasil steganografi (*stego image*).
Perbandingan kualitas citra sebelum dan setelah di sisipkan teks diukur dengan metode PSNR dan dikur dalam satuan dB. Perbandingan PSNR citra dapat disajikan pada tabel berikut.

Tabel 5 Perbandingan Psnr Citra Terhadap Kapasitas File Teks

File text	***KB***	***PSNR (dB)***				
		(CI-1)	***(CI-2)***	***(CI-3)***	***(CI-4)***	***(CI-5)***
FileUji1.txt	1K	75,11	75,19	77,37	75,08	73,00
FileUji2.txt	5K	68,14	68,00	70,52	68,03	66,07
FileUji3.txt	10K	64,77	64,78	67,23	64,82	63,11
FileUji4.txt	15K	63,05	63,05	65,47	62,98	61,70
FileUji5.txt	20K	61,62	61,60	64,00	61,54	60,56

Dari tabel di atas disimpulkan bahwa semakin besar kapasitas *hidden text* yang disisipkan ke dalam *cover text* makanilai PSNR citra semakin kecil atau dengan kata lain kualitas citra semakin buruk. Hal tersebut dapat disajikan dalam bentuk grafik pada gambar 6 berikut



Gambar 6 Grafik perbandingan PSNR Citra terhadap kapasitas file teks

4) Pengujian MD citra

Max Different (MD) adalah nilai maksimum error pada pixel citra sebelum dan setelah proses *embedding*. Nilai MD yang besar mengindikasikan besarnya perubahan pixel citra setelah *embedding.*

Tabel 6. Perbandingan MD Citra **baboon.png**

Stego image	***Horizontal***	***Verikal***
b1.png	2360,951	2112,436
b2.png	11788,259	11654,180
b3.png	25344,978	25354,441
b4.png	37813,712	38112,015
b5.png	52869,938	52721,871

Tabel 7. Perbandingan MD Citra **lenna.png**

	Horizontal	***Verikal***
l1.png	2506.792	3332.185
l2.png	11750.983	12850.606
l3.png	25753.448	26364.953
l4.png	38105.432	38643.777
l5.png	52829.099	53876.844

Tabel 8. Perbandingan MD Citra **peppers.png**

	Horizontal	***Verikal***
p1.png	2381.529	2189.778
p2.png	11926.698	11711.801
p3.png	24863.169	24930.510
p4.png	37406.366	37203.078
p5.png	52282.981	53064.180

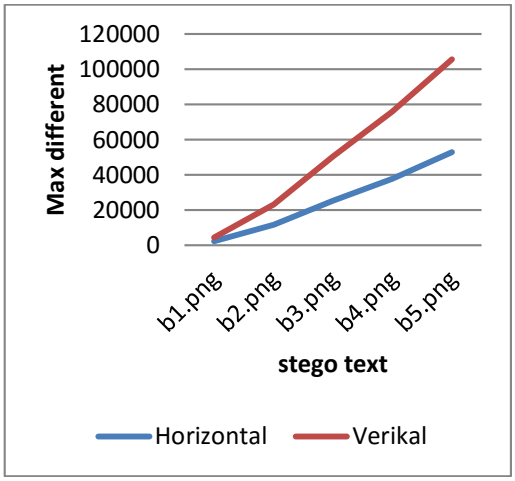
Tabel 9. Perbandingan MD Citra **gamma.png**

	Horizontal	***Verikal***
g1.png	3581.000	3626.000
g2.png	17833.000	17947.000
g3.png	38211.000	38175.000
g4.png	56924.000	56854.000
g5.png	79867.199	80187.742

Tabel 10. Perbandingan MD Citra **baboon.png**

	Horizontal	***Verikal***
h1.png	3537.919	3607.880
h2.png	17596.616	17696.479
h3.png	35306.912	34661.392
h4.png	49979.576	49677.443
h5.png	67364.201	66661.274

Dari tabel di atas ditampilkan bahwasemakin besar nilai *hidden text* yang disisipkan pada *cover image* akan diperoleh nilai MD yang lebih besar. Informasi tersebut disajikan dalam bentuk grafik pada gambar 7 berikut.

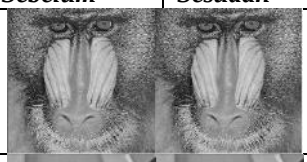
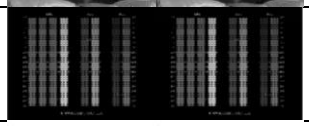


Gambar 7 Grafik MD citra baboon.png

5) HVS (Human Visual System)

Penyisipan teks pada proses steganografi tentu saja tidak mampu dibedakan dengan mata telanjang manusia. Pada tabel berikut ditampilkan HVS (*Human Visual System*) pada kelima gambar hasil steganografi. Ke lima pasangan citra sebelum-sesudah embedding memiliki HVS yang sama.

Tabel 11. HVS Pada Kelima Stegoimage

	Sebelum	***Sesudah***
Baboon.png		
Lenna.png		
Peppers.png		
Gamma.png		
House.png		

5. Kesimpulan

Dari penelitian di atas beberapa kesimpulan yang diperoleh adalah sebagai berikut ini :

1. *Embedding* pesan mempengaruhi nilai pixel pada koordinat tertentu pada cover image
2. Semakin banyak karakter yang disisipkan pada cover image maka nilai PSNR nya semakin kecil, hal tersebut mengindikasikan bahwa kualitas citra semakin menurun
3. Nilai PSNR berdanding lurus dengan nilai MD citra

6. Referensi

[1] Sen, J., 2012. Applied Cryptography and Network Security. In Tech Publisher. Rijeka: Croatia

[2] Munir, R., 2012. Kriptografi. Informatika Bandung. Bandung : Indonesia

[3] Rakhmat, B., Steganografi Menggunakan Metode Least Significant Bit Dengan Kombinasi Algoritma Kriptografi Vigenère Dan RC4. Jurnal Dinamika Informatika . vol. 5, no 2 : 1-17

[4] Patil, S. Sheelvant, S. 2015. Survey on Image Quality Assessment Techniques. International Journal of Science and Research (IJSR). vol 4 no 7 : 1756-1759.

[5] Panchal, K., J,. & Patel F., N. Steganography : A Brief Survey. International Journal of Modern Trends in Engineering and Research. vol 1 no 1 : 247-250

[6] Stalling, W., 2011. Cryptography and Network Security. Prentice Hall. New York: United states

[7] Tonde, A.,R. & Dhande, A.,P. 2014. Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA. International Journal of Current Engineering and Technology. vol.4, no 2 : 1-3

[8] Patel, P., R., Survey on Different Methods of Image Steganography. International Journal of Innovative Research in Computer and Communication Engineering. vol 2 no 12 : 7614-1618