

Penerapan Algoritma Outguess dalam Penyembunyian Pesan Pada Citra Berwarna

Yusni Elis Saragih

Program Studi Teknik Informatika Universitas Budi Darma Medan

Email: yusnieliss@gmail.com

Abstrak—Penyimpanan ataupun peredaran pesan rahasia rawan atas pengaksesan atau penyadapan oleh pihak-pihak yang tidak memiliki otoritas yang legal. Oleh karena itu, untuk menghindari hal tersebut dan juga untuk mengamankan suatu pesan rahasia dikenal suatu proses enkripsi yang dinamakan kriptografi. Agar pesan tersebut tetap terjaga kerahasiaannya maka dilakukan penyisipan pesan ke dalam objek citra. Menyisipkan pesan dalam sebuah gambar atau lebih sering dikenal dengan istilah steganografi dimana pengembangan dari metode kriptografi yang sudah dikenal sebelumnya. Metode dengan steganografi dapat dikatakan lebih baik dibandingkan dengan sebelumnya kriptografi. Untuk mengamankan pesan rahasia dari pihak ketiga yang tidak berkepentingan maka dilakukan dengan teknik steganografi.

Kata Kunci: Penyimpanan; Pesan; Steganografi

Abstract—The storage or circulation of secret messages is vulnerable to being accessed or wiretapped by parties who do not have legal authority. Therefore, to avoid this and also to secure a secret message known an encryption process called cryptography. In order for the message to be kept confidential, the message is inserted into the image object. Inserting a message in an image or more commonly known as steganography is the development of previously known cryptographic methods. The method with steganography can be said to be better than the previous cryptography. To secure secret messages from unauthorized third parties, steganography is used.

Keywords: Storage; Message; Steganography

1. PENDAHULUAN

Seiring dengan memasuki era *internet* dimana perkembangan pertukaran informasi sangat berkembang pesat. Kini informasi berupa pesan dan gambar dapat dikirimkan sebagai surat elektronik melalui *e-mail*, dipajang di *blog* pribadi sebagai *post* berita, maupun yang belakangan ini populer adalah memajangnya di *social networking* seperti *Facebook*, *Friendster*, dan *Twitter*.

Dengan adanya layanan untuk bertukar informasi di internet dengan memanfaatkan media *online* ini, maka dibukalah lembaran baru dalam metode bertukar informasi. Dengan media *online* ini, dapat dikirimkan pesan berupa tulisan, gambar, bahkan berupa *file* sisipan yang disertakan dalam pesan. *Steganografi* merupakan salah satu cara untuk menyembunyikan suatu pesan atau data rahasia didalam suatu media penampungnya sehingga orang lain tidak menyadari adanya pesan didalam media tersebut. Dalam bidang keamanan komputer, *steganografi* digunakan untuk menyembunyikan data rahasia.

Pada penelitian sebelumnya oleh: Putu Teguh Krisna dan Ni Kadek Ayu Wirdiani, 2014 yang berjudul “Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny”, mengtakan bahwa *Image* atau citra didefinisikan sebagai *a picture of mind*, yaitu suatu gambaran yang ada di dalam benak seseorang. Citra digital adalah menyatakan data citra dalam angka yang mewakili aras keabuan (citra hitam putih) atau koordinat warna (citra berwarna). Data citra digital dapat diperlakukan sebagai sebuah matrik, sehingga dengan menggunakan hasil-hasil dari teori matrik dapat dilakukan pengolahan citra digital sesuai dengan yang dikehendaki. Mode warna *Red*, *Green*, *Blue* (RGB) adalah model warna yang aditif yang dapat dikombinasikan untuk membuat warna yang lainnya. Sistem warna *Red*, *Green*, *Blue* (RGB) terdiri dari tiga bagian yang utama, yaitu *Red* (Merah), *Green* (Hijau), dan *Blue* (biru). Model warna RGB didasarkan pada sistem koordinat kartesius [1].

Algoritma *steganografi OutGuess* diusulkan oleh Neils Provos untuk melawan serangan *chi-square*. Dalam tahap awal algoritma ini mirip dengan J-Steg, Outguess adalah perangkat steganografi universal yang memungkinkan penyisipan informasi rahasia ke dalam sumber data apapun (tidak harus gambar) yang memiliki bit redundan. Program ini memerlukan *handler* yang spesifik terhadap data yang akan mengekstrak bit-bit redundan dan menuliskannya kembali setelah diadakan modifikasi Xiang. Steganografi menyembunyikan adanya komunikasi. Sistem steganografi klasik bergantung pada menjaga kerahasiaan sistem encoding, sedangkan steganografi modern hanya dapat dideteksi jika informasi rahasia berupa kunci rahasia tertangkap. Karena sifat sistemnya yang menyusup ke dalam, steganografi meninggalkan jejak dalam karakteristik dari medium penyimpanannya [2].

2. METODOLOGI PENELITIAN

2.1 Steganografi

Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah “menulis tulisan yang tersembunyi atau terselubung”. Secara umum steganografi merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari

pesan rahasia tersebut. Dari definisi di atas, maka dapat disimpulkan bahwa steganografi dibuat untuk membantu mengamankan informasi dengan cara menyembunyikan pesan pada media gambar, audio, ataupun video, agar pihak lain tidak mengetahui keberadaan informasi rahasia tersebut, kecuali si pengirim pesan dan penerima pesan [2],[3].

2.2 Pengolahan Citra

Pengolahan citra merupakan proses untuk menghasilkan citra sesuai dengan keinginan atau kualitasnya menjadi lebih baik. Inputannya adalah citra dan keluarannya juga citra tapi dengan kualitas lebih baik dari pada citra masukan. Misal citra warnanya kurang tajam, kabur (*blurring*) dan mengandung noise (misal bintik-bintik putih) sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit di interpretasikan karena informasi yang disampaikan menjadi berkurang[4]-[8].

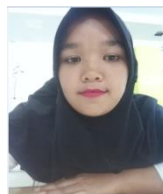
2.3 Metode Outguess

Algoritma steganografi *Outguess* diusulkan oleh Neils Provos untuk melawan serangan *chi-square*. Dalam tahap awal algoritma ini mirip dengan J-Steg. *Outguess* adalah perangkat steganografi universal yang memungkinkan penyisipan informasi rahasia ke dalam sumber data apapun (tidak harus gambar) yang memiliki *bit redundan*. Program ini memerlukan *handler* yang spesifik terhadap data yang akan mengekstrak *bit-bit redundan*. *Outguess* menjaga statistik berdasarkan hitungan frekuensi. Oleh karena itu tes statistik tidak akan bisa menentukan ada atau tidaknya penyembunyian data. Sebelum memasukkan data ke gambar, *outguess* dapat mengukur berapa besar pesan yang dapat di sembunyikan. *Outguess* menggunakan objek iterator umum untuk memilih bit mana yang harus dimodifikasi. Sebuah seed bisa dimanfaatkan untuk mengatur perilaku iterator. Perubahan disimpan di dalam data bersama dengan sisa dari pesan tersebut. Dengan mengubah seednya, *outguess* mencoba mencari sebuah urutan *bit* yang memiliki banyak perubahan minimal [2].

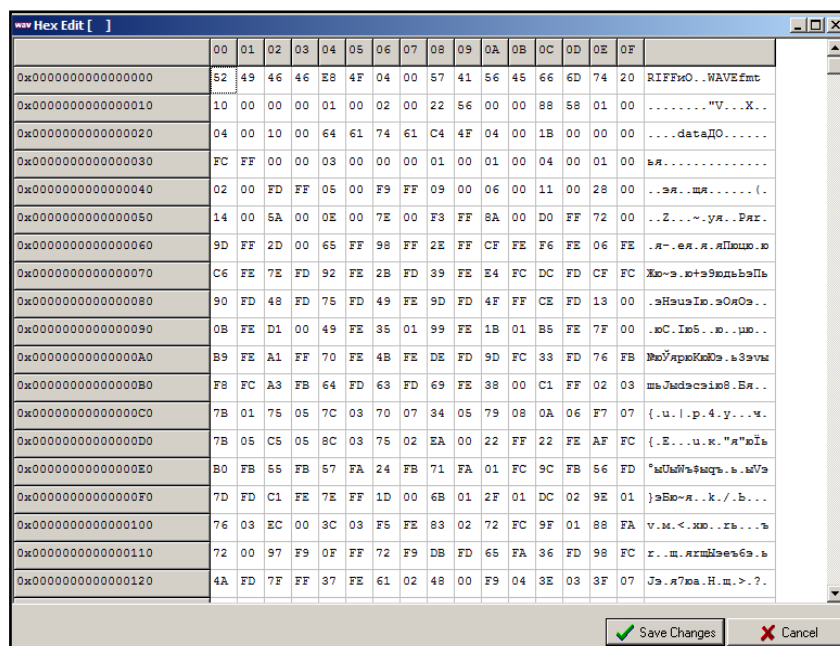
3. HASIL DAN PEMBAHASAN

3.1 Analisa Citra Sebagai Media

Metode yang dipakai dalam proses penyisipan bit-bit data ke dalam *byte-byte* citra adalah dengan menggunakan metode *outguess*. Adapun contoh gambar *audio* yang digunakan sebagai media dapat dilihat pada gambar 4, sedangkan gambar 5 merupakan hasil hexadecimal dari gambar tersebut.



Gambar 4. Cover Image



Gambar 5. Citra dalam Hexa

3.2 Analisa Teks

Untuk melakukan analisa terhadap teks maka proses pertama yang akan dilakukan yaitu dengan mengubah nilai-nilai karakter menjadi bentuk biner. Operasi dilakukan per *byte* dimana data teks di ubah kedalam bentuk biner.

Plaintext: lis saragih

Jumlah karakter pada plainteks untuk pesan rahasia yang disembunyikan yaitu 10 karakter. Berikut tabel yang digunakan sebagai *plaintext*, dimana nilai-nilai karakter diubah menjadi nilai biner yang terlihat pada tabel 1:

Tabel 1. Data Plainteks dari nama

No	Character	ASCII Value (decimal)	Hexadecimal	Binary
1.	l	101	65	01100101
2.	i	100	64	01100100
3.	s	121	79	01111001
4.	spasi	32	20	00100000
5.	s	112	70	01110000
6.	a	114	72	01110010
7.	r	97	61	01100001
8.	a	110	6E	01101110
9.	g	111	6F	01101111
10.	i	116	74	01110100
11.	h	111	6F	01101111

3.3 Proses Penyisipan Pesan (Encoding)

Proses penyisipan pesan rahasia yaitu bagaimana pesan rahasia disisipkan pada sebuah *file* citra sehingga *file* tersebut tidak diketahui keberadaannya. Pada proses ini, proses penyisipan membutuhkan dua buah masukan yaitu *media cover* sebagai tempat penyisipan pesan dan pesan rahasia. *Media cover* yang digunakan adalah *file* citra dan pesan rahasia yang dapat disisipkan berupa teks. *File* citra yang digunakan adalah *file* citra berformat **bmp*. Sedangkan pada proses penguraian hanya dibutuhkan satu buah masukan. Masukan tersebut adalah *file* citra yang telah disisipi pesan rahasia. Untuk melanjutkan cara kerja dalam penyisipan pesan rahasia ini, cara kerja yang dilakukan adalah dengan menginputkan *file* citra dan menyimpan *file* citra tersebut kemudian *inputkan* sebuah pesan rahasia, buat nama *file* citra yang telah disisipi pesan dan simpan *file* citra yang telah disisipkan pesan. *Output* yang dihasilkan juga bergantung pada proses yang akan dilakukan. Pada proses penyisipan, keluaran yang dihasilkan adalah citra yang telah disisipi pesan rahasia (*file stego*). Sedangkan pada proses ekstraksi, keluarannya adalah pesan rahasia yang dapat berupa teks.

Misalkan media suara yang akan disisipkan mempunyai panjang bit 8, dengan nilai seperti berikut. Berikut adalah *file audio* sebagai media penampung pesan.

Penyisipan pesan rahasia yaitu dengan mengganti bit-bit akhir. Setiap bit pesan rahasia disisipkan ke dalam bit media yaitu pada setiap bit akhir media. Adapun proses penyisipan dilakukan dengan operasi *And* dan *Or*. Operasi logika *And* digunakan apabila bit pesan yang disisipkan yaitu bernilai 0 dan operasi logika *Or* digunakan apabila bit pesan yang disisipkan yaitu bernilai 1. Maka *stego* yang dihasilkan adalah sebagai berikut:

```

01010010 and 0 = 01010010
01001001 or 1 = 01001001
01000110 or 1 = 01000111
01000110 and 0 = 01000110
11101000 and 0 = 11101000
01001111 or 1 = 01001111
00000100 and 0 = 00000100
00000000 or 1 = 00000001

01010111 and 0 = 01010110
01000001 or 1 = 01000001
01010110 or 1 = 01010111
01000101 and 0 = 01000100
01100110 and 0 = 01100110
01101101 or 1 = 01101101
01110100 and 0 = 01110100
00100000 and 0 = 00100000

00010000 and 0 = 00010000
00000000 or 1 = 00000001
00000000 or 1 = 00000001
00000000 or 1 = 00000001

```

00000001 *or* 1 = 00000001
 00000000 *and* 0 = 00000000
 00000010 *and* 0 = 00000010
 00000000 *or* 1 = 00000001

00100010 *and* 0 = 00100010
 01010110 *and* 0 = 01010110
 00000000 *or* 1 = 00000001
 00000000 *and* 0 = 00000000
 10001000 *and* 0 = 10001000
 01011000 *and* 0 = 01011000
 00000001 *and* 0 = 00000000
 00000000 *and* 0 = 00000000

00000100 *and* 0 = 00000100
 00000000 *or* 1 = 00000001
 00010000 *or* 1 = 00010001
 00000000 *or* 1 = 00000001
 01100100 *and* 0 = 01100100
 01100001 *and* 0 = 01100000
 01110100 *and* 0 = 01110100
 01100001 *and* 0 = 01100000

11000100 *and* 0 = 11000100
 01001111 *or* 1 = 01001111
 00000100 *or* 1 = 00000101
 00000000 *or* 1 = 00000001
 00011011 *and* 0 = 00011010
 00000000 *and* 0 = 00000000
 00000000 *or* 1 = 00000001
 00000000 *and* 0 = 00000000

11111100 *and* 0 = 11111100
 11111111 *or* 1 = 11111111
 00000000 *or* 1 = 00000001
 00000000 *and* 0 = 00000000
 00000011 *and* 0 = 00000010
 00000000 *and* 0 = 00000000
 00000000 *and* 0 = 00000000
 00000000 *or* 1 = 00000001

00000001 *and* 0 = 00000000
 00000000 *or* 1 = 00000001
 00000001 *or* 1 = 00000001
 00000000 *and* 0 = 00000000
 00000100 *or* 1 = 00000101
 00000000 *or* 1 = 00000001
 00000001 *or* 1 = 00000001
 00000000 *and* 0 = 00000000

00000010 *and* 0 = 00000010
 00000000 *or* 1 = 00000001
 11111101 *or* 1 = 11111101
 11111111 *and* 0 = 11111110
 00000101 *or* 1 = 00000101
 00000000 *or* 1 = 00000001
 11111001 *or* 1 = 11111001
 11111111 *or* 1 = 11111111

00001001 *and* 0 = 00001000
 00000000 *or* 1 = 00000001
 00000110 *or* 1 = 00000111

00000000 or 1 = 00000001
 00010001 and 0 = 00010000
 00000000 or 1 = 00000001
 00101000 and 0 = 00101000
 00000000 and 0 = 00000000

 00010100 and 0 = 00010100
 00000000 or 1 = 00000001
 01011010 or 1 = 01011011
 00000000 and 0 = 00000000
 00001110 or 1 = 00001111
 00000000 or 1 = 00000001
 01111110 or 1 = 01111111
 00000000 or 1 = 00000001

Berikut tabel pesan biner yang disisipkan dengan hasil penyisipan pesan terhadap media suara yang dijadikan sebagai penampung.

Tabel 6. Pesan Biner Yang Ingin Disisipkan

0	1	1	0	0	1	0	1
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	1
0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	1	0
0	1	1	0	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	0	1	1	1	1
0	1	1	1	0	1	0	0
0	1	1	0	1	1	1	1

File sebagai pesan rahasia yang sudah disisipi pada sebuah media citra tersebut sudah tersimpan.

3.2 Decoding

Untuk menampilkan pesan rahasia ke nilai awal ataupun pesan asli yaitu pada proses ini pesan asli yang telah disisipi file citra diekstraksi. Proses ekstraksi dimulai dengan menginputkan pesan yang sudah disisipi file citra kemudian ekstrak dan simpan nama pesan yang sudah diekstrak. Sebagai contoh akan dilakukan decoding untuk membaca informasi yang disisipkan dengan mengambil nilai bit lsb dari media penampung file citra seperti berikut ini:

01010010=>01010010 and 1=0 nilai=0x2^7=0
 01001001=>01001001 and 1=1
 nilai=1x2^6=64
 01000110=>01000111 and 1=1
 nilai=1x2^5=32
 01000110=>01000110 and 1=0 nilai=0x2^4=0
 11101000=>11101000 and 1=0 nilai=0x2^3=0
 01001111=>01001111 and 1=1 nilai=1x2^2=4
 00000100=>00000100 and 1=0 nilai=0x2^1=0
00000000=>00000001 and 1=1 nilai=1x2^0=1 +
 =101 kode ASCII yaitu "I"

01010111 =>01010110 and 1=0 nilai=0 x2^7=0
 01000001 =>01000001 and 1=1 nilai=1 x2^6=64
 01010110 =>01010111 and 1=1 nilai=1 x2^5=32
 01000101 =>01000100 and 1=0 nilai=0 x2^4=0
 01100110 =>01100110 and 1=0 nilai=0 x2^3=0
 01101101 =>01101101 and 1=1 nilai=1 x2^2=4
 01110100 =>01110100 and 1=0 nilai=0 x2^1=0
00100000 =>00100000 and 1=0 nilai=0 x2^0=0 +
 =100 kode ASCII yaitu "i"

00010000 =>00010000 and 1=0 nilai=0 x2^7=0
 00000000 =>00000001 and 1=1 nilai=1 x2^6=64

00000000 =>0000000**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**1** and 1=1 nilai=1 x2^4=16
 00000001 =>0000000**1** and 1=1 nilai=1 x2^3=8
 00000000 =>0000000**0** and 1=0 nilai=0 x2^2=0
 00000010 =>0000000**0** and 1=0 nilai=0 x2^1=0
00000000 =>0000000**1** and 1=1 nilai=1 x2^0= 1 +
 = 121 kode ASCII yaitu “s”

00100010 =>0010001**0** and 1=0 nilai=0 x2^7=0
 01010110 =>0101011**0** and 1=0 nilai=0 x2^6=0
 00000000 =>0000000**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**0** and 1=0 nilai=0 x2^4=0
 10001000 =>1000100**0** and 1=0 nilai=0 x2^3=0
 01011000 =>0101100**0** and 1=0 nilai=0 x2^2=0
 00000001 =>0000000**0** and 1=0 nilai=0 x2^1=0
00000000 =>0000000**0** and 1=0 nilai=0 x2^0= 0 +
 = 32 kode ASCII yaitu “spasi”

00000100 =>0000010**0** and 1=0 nilai=0 x2^7=0
 00000000 =>0000000**1** and 1=1 nilai=1 x2^6=64
 00010000 =>0001000**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**1** and 1=1 nilai=1 x2^4=16
 01100100 =>0110010**0** and 1=0 nilai=0 x2^3=0
 01100001 =>0110000**0** and 1=0 nilai=0 x2^2=0
 01110100 =>0111010**0** and 1=0 nilai=0 x2^1=0
01100001 =>0110000**0** and 1=0 nilai=0 x2^0= 0 +
 = 112 kode ASCII yaitu “s”

11000100 =>1100010**0** and 1=0 nilai=0 x2^7=0
 01001111 =>0100111**1** and 1=1 nilai=1 x2^6=64
 00000100 =>0000010**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**1** and 1=1 nilai=1 x2^4=16
 00011011 =>0001101**0** and 1=0 nilai=0 x2^3=0
 00000000 =>0000000**0** and 1=0 nilai=0 x2^2=0
 00000000 =>0000000**1** and 1=1 nilai=1 x2^1=2
00000000 =>0000000**0** and 1=0 nilai=0 x2^0= 0 +
 = 114 kode ASCII yaitu “a”

11111100 =>1111110**0** and 1=0 nilai=0 x2^7=0
 11111111 =>1111111**1** and 1=1 nilai=1 x2^6=64
 00000000 =>0000000**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**0** and 1=0 nilai=0 x2^4=0
 00000011 =>0000000**10** and 1=0 nilai=0 x2^3=0
 00000000 =>0000000**0** and 1=0 nilai=0 x2^2=0
 00000000 =>0000000**0** and 1=0 nilai=0 x2^1=0
00000000 =>0000000**1** and 1=1 nilai=1 x2^0= 1 +
 = 97 kode ASCII yaitu “r”

00000001 =>0000000**0** and 1=0 nilai=0 x2^7=0
 00000000 =>0000000**1** and 1=1 nilai=1 x2^6=64
 00000001 =>0000000**1** and 1=1 nilai=1 x2^5=32
 00000000 =>0000000**0** and 1=0 nilai=0 x2^4=0
 00000100 =>0000010**1** and 1=1 nilai=1 x2^3=8
 00000000 =>0000000**1** and 1=1 nilai=1 x2^2=4
 00000001 =>0000000**1** and 1=1 nilai=1 x2^1=2
00000000 =>0000000**0** and 1=0 nilai=0 x2^0= 0 +
 = 110 kode ASCII yaitu “a”

00000010 =>0000001**0** and 1=0 nilai=0 x2^7=0
 00000000 =>0000000**1** and 1=1 nilai=1 x2^6=64
 11111101 =>1111110**1** and 1=1 nilai=1 x2^5=32
 11111111 =>1111111**0** and 1=0 nilai=0 x2^4=0

00000101 =>0000010**1** and 1=1 nilai=1 $x2^3=8$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^2=4$
 11111001 =>1111100**1** and 1=1 nilai=1 $x2^1=2$
11111111 =>1111111**1** and 1=1 nilai=1 $x2^0=1$ +

= 111 kode ASCII yaitu “g”

00001001 =>0000100**0** and 1=0 nilai=0 $x2^7=0$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^6=64$
 00000110 =>0000011**1** and 1=1 nilai=1 $x2^5=32$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^4=16$
 00010001 =>0001000**0** and 1=0 nilai=0 $x2^3=0$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^2=4$
 00101000 =>0010100**0** and 1=0 nilai=0 $x2^1=0$
00000000 =>0000000**0** and 1=0 nilai=0 $x2^0=0$ +

= 116 kode ASCII yaitu “i”

00010100 =>0001010**0** and 1=0 nilai=0 $x2^7=0$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^6=64$
 01011010 =>0101101**1** and 1=1 nilai=1 $x2^5=32$
 00000000 =>0000000**0** and 1=0 nilai=0 $x2^4=0$
 00001110 =>0000111**1** and 1=1 nilai=1 $x2^3=8$
 00000000 =>0000000**1** and 1=1 nilai=1 $x2^2=4$
 01111110 =>1111100**1** and 1=1 nilai=1 $x2^1=2$
00000000 =>0000000**1** and 1=1 nilai=1 $x2^0=1$ +

= 111 kode ASCII yaitu “h”

Hasilnya akan terlihat pesan aslinya yaitu lis saragih.

4. KESIMPULAN

Berdasarkan hasil penelitian disimpulkan penyisipan pesan pada gambar dengan menggunakan metode *outguess* dimana *size* yang dihasilkan lebih kecil. Secara kualitas gambar yang dihasilkan juga menurun dan pada grafik teksture yang dihasilkan juga mengalami perubahan.

REFERENCES

- [1] Putu Teguh Krisna dan Ni Kadek Ayu Wirdiani, 2014, Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny”, Jurnal Merpati, Vol. 2 No. 2, ISSN : 2252-3006.
- [2] Wijaksono. B.A, 2017, “Steganografi Pada Citra Digital Dengan Metode Catmap dan Outguess”, Jurnal String, Vol. 1, No. 3
- [3] A. Ari ulan, dkk, 2017, “Steganografi Pada Video Menggunakan Metode Least Significant Bit (LSB) dan End Of File (EOF)”, Jurnal Informatika Mulawarman, Vol. 12, No. 2, e-ISSN : 2597-4963, p-ISSN : 1858-4853.
- [4] Kadir, Abdul dan Adhi Susanto. 2013. “Teori dan Aplikasi pengolahan citra”. Penerbit Andi, Yogyakarta.
- [5] Hermawati, Fajar Astuti. 2013. “Pengolahan Citra Digital Konsep dan Teori”. Penerbit Andi, Yogyakarta.
- [6] Wijaya, Marvin ch.2007, “Pengolahan Citra Digital Matlab”, Penerbit Andi, Yogyakarta.
- [7] T Sutoyo, Teori Pengolahan Citra Digital. Yogyakarta: ANDI, 2009.
- [8] A. F. Yana, “Implementasi Pengolahan Citra Digital Pada Penghitungan Anak Burung Puyuh Menerapkan Metode Blob,” J. Inf. Syst. Res., vol. 1, no. 4, pp. 237–245, 2020.