SISTEM STEGANOGRAFI DENGAN METODE LEAST SIGNIFICANT BIT (LSB) TERACAK

Yohanes Julianto[1], Kristoforus Jawa Bendi[2]

Abstract:

Information security has become important today. Steganography is one of the ways to make any information secure. It hides information by inserting it in a media or cover. LSB (Least Significant Bit)steganographic technique has become popular because of its simplicity. LSB (Least Significant Bit) is one algorithm from steganography. The weakness of LSB is bits of embedded are inserted sequentially. It can be easily tracked. Our research using text as embedded message and WAV audio format as cover-object. We combine LSB and LCG (Linear Congruetial Generator) to determinant the random position of replacement bit. Outcome of this research is a software that was develop by waterfall model and coded in Visual Basic. The result shows that embedded message can be encode dan decode.

Keywords : *Steganography, LSB, LCG, WAV audio format.*

1. PENDAHULUAN

Masalah keamanan merupakan salah satu aspek penting dalam pengiriman data dan informasi dalam suatu jaringan. Salah satu cara yang biasa digunakan untuk mengamankan data adalah dengan memanfaatkan steganografi. Steganografi adalah suatu seni untuk menyembunyikan suatu data, dimana data tersebut disembunyikan ke dalam suatu media yang tampak biasa saja [12].

Digital steganografi memerlukan suatu media sebagai tempat penyembunyian informasi. Secara teori penyisipan informasi pada data digital dengan menggunakan teknik steganografi dapat dilakukan pada semua format data digital yang ada dalam komputer sebagai media *cover*nya seperti format teks, format gambar, bahkan untuk format audio dan sebagainya asalkan *file-file* tersebut mempunyai bit-bit data *redundan* yang dapat dimodifikasi [5].

Salah satu format multimedia yang dapat digunakan sebagai media steganografi yaitu *file* multimedia dengan format *wav*. Jika dibandingkan dengan mp3, pada awalnya mp3 merupakan *file* multimedia yang berformat *wav*, tetapi mengalami kompresi. Sehingga, bisa dipastikan bahwa ukuran *file* mp3 dan *wav* akan berbeda. Kualitas suara yang dihasilkan oleh kedua *file* multimedia ini juga akan berbeda, yang menurut teorinya bahwa *file* multimedia yang berformat *wav* memiliki kualitas suara lebih bagus dibandingkan *file audio* mp3, namun membutuhkan kapasitas penyimpanan yang cukup besar [9].

Penelitian ini bertujuan untuk membangun sebuah sistem steganografi untuk menyembunyikan pesan teks dalam media audio berformat WAV dengan teknik *Least Significant Bit* (LSB).

2. TINJAUAN PUSTAKA

Steganografi membutuhkan dua properti yaitu: wadah penampung dan data/infomasi yang akan disembunyikan. Wadah penampung dapat berupa media digital seperti citra, suara (*audio*), teks dan citra bergerak (*video*). Pesan yang disembunyikan juga dapat berupa citra,

Julianto[1], Jawa Bendi[2], *Adalah Dosen Program Studi Informatika, Universitas Katolik Musi Charitas Palembang*

suara, teks atau video. Keuntungan penggunaan steganografi adalah memungkinkan pengiriman pesan secara rahasia tanpa diketahui bahwa pesan sedang dikirim. Ini membuat pihak ketiga tidak menyadari keberadaan pesan. Steganografi juga memiliki kelemahan yaitu memerlukan banyak ruang untuk dapat menyembunyikan beberapa bit pesan. Akan tetapi, kelemahan ini sedikit demi sedikit dapat diatasi seiring dengan perkembangan teknik-teknik dalam melakukan steganografi [5].

Sebuah sistem steganografi terdiri atas: (1) *embedded message*: pesan atau informasi yang disembunyikan, (2) *cover-object*: pesan yang digunakan untuk menyembunyikan *embedded message*, (3) *stego-object*: pesan yang sudah berisi embedded message, (4) *encoding*: proses menyembunyikan *embedded message* dalam *cover-object*, dan (5) *decoding*: proses mengambil/membaca *embedded message* dari *stego-object*. Dalam menyembunyikan pesan ada beberapa kriteria yang harus dipenuhi sebagai berikut [12].

1. *Fidelity*; mutu *cover-object* tidak banyak berubah setelah disisipi *embedded message*. Secara indrawi pengamat dapat membedakan *cover-object* dan *stego-object*.
2. *Robustness*; *embedded message* harus tahan (*robust*) atau tidak hilang apabila terjadi manipulasi pada *stego-object*, seperti pemotongan (*cropping*) dan sebagainya.
3. *Recovery*; embedded message harus dapat diungkapkan kembali (*reveal*) melalui proses *decoding*.

Secara teoritis seluruh media digital dapat digunakan sebagai *cover-object*. Salah satunya adalah file suara berformat wav (*Waveform Audio Format*). File wav memiliki kapasitas yang cukup besar untuk menampung pesan rahasia dan memiliki kualitas suara yang baik karena belum mengalami kompresi [9]. Beberapa penelitian sebelumnya [15] dan [9] mengemukakan bahwa file atau pesan rahasia dapat disisipkan ke dalam media wav, dan dapat dikatakan tidak mengalami perubahan kualitas dan kapasitas. Penelitian-penelitian tersebut juga mengemukakan bahwa ukuran pesan rahasia tidak boleh lebih besar dari pada ukuran file wav. Terlihat jelas bahwa jika ukuran pesan rahasia yang disisipkan cukup besar, maka waktu yang dibutuhkan untuk menyisipkan juga akan lebih lama. Penelitian [15] menemukan bahwa file wav yang sudah disisipkan pesan rahasia tidak dapat dideteksi oleh *software* yang dapat mendeteksi file stego.

Ada banyak teknik atau algoritma yang dapat digunakan untuk proses *encoding* dan *decoding* pada steganografi, namun teknik yang dianggap paling tidak menyebabkan gangguan atau *noise* pada file stego-audio adalah dengan menggunakan LSB. Sehingga banyak peneliti yang menggunakan algoritma ini untuk melakukan penyisipan data, walaupun dengan menggunakan *cover-object* yang beragam.

Penelitian-penelitian sebelumnya yang menggunakan metode LSB menemukan hasil yang beragam. Pada penelitian [2] dan [3] menemukan bahwa dengan penggunaan algoritma LSB, tidak terdapat perbedaan antara cover-object dengan *stego-object*, baik kualitas dan kapasitasnya. Sedangkan peneliti lainnya [1], [4], [7], dan [8], menemukan adanya perubahaan dari *cover-object* dengan *stego-object*, baik secara kualitas, kapasitas maupun noise yang dihasilkan. Perubahan yang terjadi diakibatkan besarnya jumlah pesan rahasia yang disisipkan, semakin besar ukuran pesan rahasia yang disisipkan, maka akan semakin besar pula perubahan yang terjadi pada file tersebut. Hal ini menunjukkan bahwa penggunaan dengan metode LSB, aspek *fidelity* dari steganografi dapat terpenuhi.

Berdasarkan aspek *recovery*, penelitian-penelitian sebelumnya menemukan bahwa dengan penggunaan metode LSB seluruh *embedded message* diambil kembali dari *stego-object* melalui proses *decoding*. Tidak banyak penelitian yang membahas aspek *robustness* dalam temuan penelitiannya. [6],[8] dan [16] mengungkapkan bahwa apabila dilakukan kompresi pada *stego-object*, maka akan terjadi kerusakan pada *embedded message*.

Pada metode LSB, bit-bit *embedded message* akan disisipkan pada bit terakhir (bit

paling tak berarti) dari setiap byte data *cover-object*. Apabila bit-bit *embedded message* disisipkan secara berturutan dalam byte-byte *cover-object*, maka kemungkinan untuk melacak *embedded message* akan sangat mudah. Untuk meningkatkan keamanan data/informasi yang akan dilindungi, beberapa penelitian sebelumnya mengkombinasikan metode LSB dengan teknik enkripsi [10, 11, 14]. Namun dalam penelitian ini, untuk menghindari pelacakan, bit-bit *embedded message* tidak disisipkan secara bertuturan, namun dipilih susunan byte *cover-object* secara acak. Untuk itu diperlukan sebuah metode pembangkit bilangan acak atau *Pseudo Random Number Generator* (PRNG) . Salah satu metode PRNG yang sering digunakan adalah *Linear Congruential Generator* (LCG)[12]. LCG merupakan pembangkit bilangan acak yang berbentuk sebagai berikut.

$X_n = (aX_{n-1} + b) \bmod m$ (1)

dengan:

X_n = bilangan acak ke-*n* dari deretnya
X_{n-1} = bilangan acak sebelumnya
a = faktor pengali
b = *increment*
m = modulus

Kunci pembangkit adalah X_0 yang disebut **umpan** (*seed*). LCG mempunyai periode tidak lebih besar dari m, dan pada kebanyakan kasus periodenya kurang dari itu. LCG mempunyai periode penuh (m – 1) jika memenuhi syarat berikut:

1. b relatif prima terhadap m.
2. a – 1 dapat dibagi dengan semua faktor prima dari m
3. a – 1 adalah kelipatan 4 jika m adalah kelipatan 4
4. $m >$ maks(a, b, x_0)
5. $a > 0$, $b > 0$

3. METODOLOGI PENELITIAN

Luaran penelitian ini berupa sebuah perangkat lunak sistem steganografi. Karena itu, dalam penelitian ini digunakan model proses *waterfall* sebagai tahapan pengembangan perangkat lunaknya. Secara umum model proses waterfall terbagi dalam empat tahap, yakni: tahap analisis, tahap desain, tahap pengkodean, dan tahap pengujian sistem [13].

4. HASIL DAN PEMBAHASAN

Pada tahapan analisis, diagram alir proses *encoding* dan proses *decoding* dibuat dengan mengkombinasikan metode LSB dan LCG. Misalkan *embedded message* adalah "jul" ("01101010 01110101 01101100") dan *cover-object* yang digunakan adalah "test.wav" dengan ukuran 307.200 B. Jika dikonversikan ke dalam biner dari sampel *cover-object* dapat terlihat seperti pada Tabel 1.

Tabel 1. Sampel byte-byte data cover object

00110011	10101100	10110000	11110000	01010101	…
10111001	10000110	10000111	10110000	11110000	…
00110011	10101010	01010101	11111000	01010111	…
11111000	11110000	10101010	10101111	11000000	…
00011111	10001111	10000001	10000000	11000110	…
10011110	00101111	10111111	00110111	10111011	…
00001110	11100000	11001011	10101011	11101111	…
00011111	10111111	10101011	10011111	10010010	…
…	…	…	…	…	…

Setelah file wav di konversikan ke dalam baris bit, maka proses *encoding* dengan teknik LSB tanpa pengacakan posisi bit dilakukan dengan menggantikan bit-bit terakhir dari setiap byte *cover-object* secara berturutan. Hasil proses encoding seperti pada tabel 2.

Tabel 2. Sampel byte-byte data stego-object dengan LSB

00110010	10101101	10110001	11110000	01010101	…
10111000	10000111	10000110	10110000	11110001	
00110011	10101011	01010100	11111001	01010110	…
11111001	11110000	10101011	10101111	11000000	
00011111	10001111	10000000	10000000	11000110	…
10011110	00101111	10111111	00110111	10111011	…
00001110	11100000	11001011	10101011	11101111	
00011111	10111111	10101011	10011111	10010010	…
…	…	…	…	…	…

Untuk melakukan pengacakan posisi byte penyisipan, maka ditambahkan teknik LCG untuk mendapatkan nilai random yang akan dijadikan sebagai penentu posisi byte *cover-object* yang akan digantikan bit terakhirnya. Nilai konstanta a,b, dan m pada persamaan (1) sangat mempengaruhi posisi dan jumlah periode

untuk pembangkitan bilangan. Apabila nilai yang dipilih tidak memenuhi persyaratan maka terdapat kemungkinan posisi byte yang akan berulang sebelum periode m-1. Dimisalkan, nilai konstanta yang didapat yaitu : a = 5.521 dan b = 33.787, dengan m = 307.200 (ukuran byte cover-object), dan nilai umpan (X_0) = 3021. Dengan mengacu pada persamaan (1), maka posisi penyisipan bit (x) dapat dihitung sebagai berikut:

X_1 = (5521 x 3021 + 33787) mod 307200
X_1 = (16712728) mod 307200 = **123928**

X_2 = (5521 x 123928 + 33787) mod 307200
X_2 = 684249275 mod 307200**105875**

X_3 = (5521 x 105875 + 33787) mod 307200
X_3 = 584569662 mod 307200= **275262**

X4 = (5521 x 275262 + 33787) mod 307200
X4 = 1519755289 mod 307200= **36889**

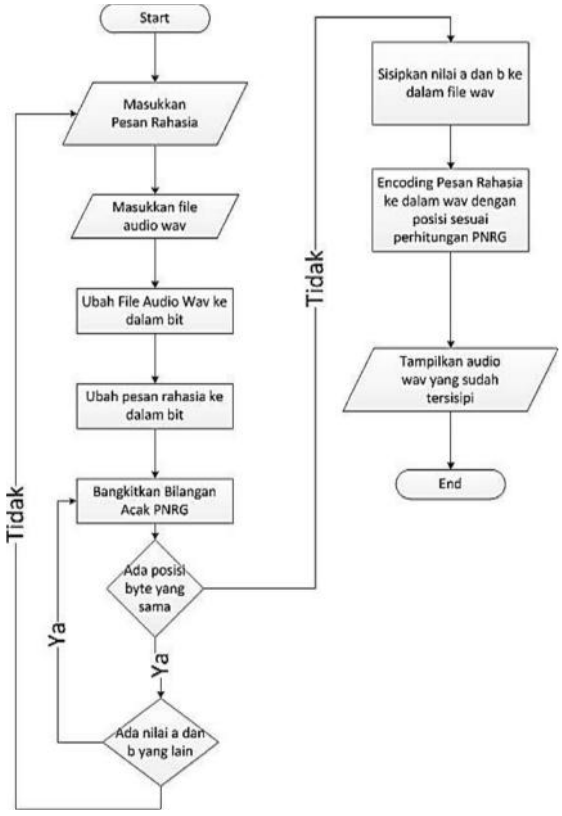
Dan seterusnya untuk X_5, X_6, X_7, dan X_8. Setelah mendapatkan nilai posisi penyisipan bit, kemudian dilakukan proses encoding dengan menggantikan bit-bit terakhir dari setiap byte cover-object berdasarkan posisi byte yang telah dihitung sebelumnya (x). Tabel 3 memperlihatkan posisi byte cover-object dan bitbit yang digantikan.

Tabel 3. Sampel byte-byte data stego-object dengan LSB+LCG

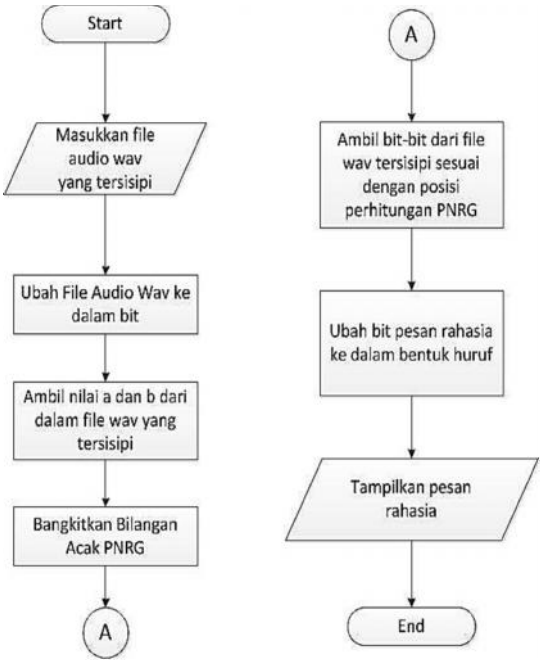
Byte	…	7933	24356	27691	32759
		1110111**1**	1010101**1**	1010101**1**	1010111**1**
Byte	…	33429	36889	57161	94202
		1011100**0**	1010101**0**	1110010**0**	1010011**0**
Byte	…	105875	123928	125268	131215
		1110111**1**	1000011**0**	1010111**1**	1011111**1**
Byte	…	139210	159024	192945	209480
		1010111**1**	1010111**0**	1011100**1**	1010101**0**
Byte	…	214894	220732	237398	256863
		1011101**1**	1100010**0**	1011100**1**	1010111**0**
Byte	…	2626262	272057	275262	304997
		1010111**0**	1011100**1**	1011101**1**	1011100**0**

Berdasarkan ilustrasi tersebut, kemudian dibuatkan diagram alir kombinasi LSB+LCG (Gambar 1), proses encoding (Gambar 2) dan diagram alir proses decoding (Gambar 3).

Gambar 1. Diagram Alir metode LSB+LCG

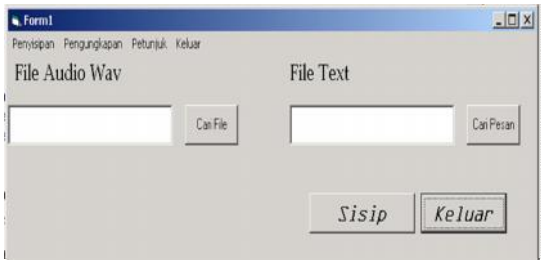


Gambar 2. Diagram Alir Proses Encoding



Gambar 3. Diagram Alir Proses Decoding

Pada tahapan desain, akan dibuatkan desain antarmuka pengguna sistem. Gambar 4 dan Gambar 5 menampilkan contoh antarmuka sistem. Perangkat lunak sistem ini dibangun dengan bahasa pemrograman Visual Basic dan dijalankan pada platform bersistem operasi Windows.

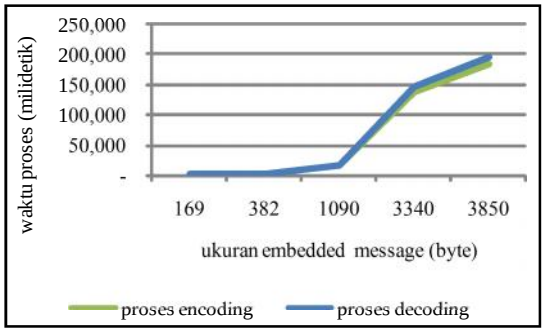


Gambar 4. Antarmuka Proses Endcoding

Untuk memastikan fungsionalitas perangkat lunak telah dilakukan pengujian perangkat lunak. Pengujian dilakukan dengan menggunakan metode *blackbox testing*. Hasil pengujian menunjukkan bahwa semua fungsi perangkat lunak berjalan dengan baik.

Gambar 5. Antarmuka Proses Decoding

Selain pengujian fungsionalitas perangkat lunak, juga dilakukan pengukuran waktu proses encoding dan proses decoding, serta pengukuran aspek steganografi yang meliputi aspek fidelity, robustness, dan aspek recovery.



Gambar 6. Antarmuka Proses Decoding

Hasil pengukuran waktu (Gambar 6) dengan 25 sampel pengukuran menunjukkan bahwa semakin besar embedded message, waktu yang dibutuhkan untuk proses encoding/decoding semakin lama. Apabila dibandingkan waktu encoding dan waktu decoding, terlihat bahwa waktu yang dibutuhkan untuk proses decoding lebih lama dibandingkan dengan proses encoding.

Pengukuran fidelity, dilakukan secara persepsional dan rasio perubahan antara cover-object dan stego-object. Pengukuran persepsional dilakukan dengan memperdengarkan lima cover-obejct dan stego-objectnya kepada 30 responden, kemudian responden menilai apakah ada perbedaan diantara keduanya dengan rentang skor 1 s.d. 5. Hasil pengukuran fidelity secara persepsional (Tabel 4) menunjukkan bahwa rerata penilaian persepsional adalah 4.5. Hal ini berarti secara persepsional aspek fidelity dapat dikatakan baik.

Tabel 4. Pengukuran Aspek Fidelity secara Persepsional

Pernyataan	Mean
Audio "bird_sounds.*wav*" dengan "msg_bird_sounds.*wav*" tidak memiliki perbedaan	4.566667
Audio "Celine Dion – To Love You More.*wav*" dengan "msg Celine Dione – To Love You More.*wav*" tidak memiliki perbedaan?	4.4333
Audio "cat_song.*wav*" dengan "msg_cat_song.*wav*" tidak memiliki perbedaan?	4.5333
Audio "Florida_e_David_gueta….*wav*" dengan "msg_Florida_e_David_gueta….*wav*" tidak memiliki perbedaan?	4.4
Audio "HStone.*wav*" dengan "msg_HStone.*wav*" tidak memiliki perbedaan?	4.56667

Pengukuran perubahan bit antara cover-object dan stego-object, menunjukkan bahwa perubahan bit yang terjadi sekitar 0,0006%. Nilai ini setara dengan rasio embedded message jika dibandingkan dengan cover-objectnya. Hal ini berarti perubahan bit yang terjadi pada cover-object sebanding dengan ukuran embedded message.

Tabel 5. Pengukuran Perubahan Bit Cover-object

Ukuran cover-object (byte)	Ukuran embedded message (byte)	perubahan bit (byte)	rasio embdded terhadap cover-object (%)	rasio perubahan bit terhadap cover-object (%)
10.913.818	382	25,25	0,0035001	0,0002314
10.913.818	3340	240,375	0,0306034	0,0022025
41.085.170	1090	65,25	0,0026530	0,0001588
41.085.170	3850	562,75	0,0093708	0,0013697
78.155.776	169	8,5	0,0002162	0,0000109
78.155.776	382	84,375	0,0004888	0,0001080
78.155.776	1090	252,75	0,0013947	0,0003234
78.155.776	3340	501,375	0,0042735	0,0006415
78.155.776	3850	544	0,0049261	0,0006960
		rerata	**0,0063807**	**0,0006380**

Pengukuran secara objektif juga dilakukan dengan menghitung nois/gangguan/derau yang dihasilkan setelah proses encoding. Pengukuran ini menggunakan metode PSNR (*Peak Signal-to-Noise Ratio*) dengan rumus:

$$PSNR = 10 \log 10 \left(\frac{P1^2}{P1^2 + P0^2 - 2\,P1\,P0} \right)$$

dimana P1 adalah kekuatan sinyal berkas audio setelah proses penyembunyian pesan dan P0 adalah kekuatan sinyal awal. Semakin besar nilai PSNR maka noise yang terjadi semakin kecil. Derau yang dihasilkan dikatakan baik jika nilai PSNR >= 30db. Hasil pengukuran (Tabel 6) dengan sembilan data sampel menunjukkan bahwa secara keseluruhan nilai PSNR masih dapat diterima. Dengan demikian aspek fidelity sistem ini masih dapat terpenuhi.

Tabel 6. Pengukuran PSNR

P0 (db)	P1 (db)	PSNR (db)
-15,59	-15,48	42,967
-15,59	-15,57	57,825
-15,59	-15,587	74,3
-10,32	-10,36	48,2
-10,32	-10,29	50,7
-10,32	-10,4	42,22
-15,01	-15,22	37,2
-15,01	-14,58	30,6
-15,01	-15,00	63,5
rerata		**49,723556**

Pengukuran aspek robustness dilakukan dengan memotong (*cropping*) stego-object, kemudian dilakukan proses decoding. Seluruh hasil pengukuran dengan lima data sampel menunjukkan bahwa proses decoding tidak menghasilkan embedded message yang diharapkan. Hal ini berarti bahwa sistem steganografi yang dibangun ini belum memenuhi aspek robustness. Pengukuran aspek recovery dilakukan dengan lima data sampel. Seluruh hasil pengukuran menunjukkan bahwa embedded message bisa didapatkan kembali melalui proses decoding, sepanjang stego-object belum termanipulasi.

5. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, peneliti menarik beberapa kesimpulan sebagai berikut.

1. Perangkat lunak pengamanan pesan rahasia menggunakan algoritma LSB dan *file audio wav* sebagai medianya dapat digunakan dengan baik untuk menyembunyikan pesan rahasia teks, yang bertujuan agar orang lain tidak menyadari ada pesan di dalam *file audio wav* tersebut.
2. Pesan rahasia hanya dapat disisipkan jika ukuran pesan rahasia tidak melebihi ukuran dari *file audio wav* yang akan disisipi.
3. Semakin besar ukuran pesan rahasia dan ukuran *file audio wav*, maka akan semakin lama proses penyisipan dan pengungkapan.
4. Kelemahan dari perangkat lunak ini adalah *file audio wav* tidak dapat mengalami pemotongan ataupun pemanjangan *file audio wav*, karena dapat mengganggu proses perhitungan algoritma PRNG.
5. Berdasarkan pengujian secara subjektif terdengar bahwa tidak ada perbedaan yang dapat dirasakan oleh telinga normal.
6. Semakin banyak pesan rahasia yang terdapat di dalam file media, maka semakin banyak merubah kekuatan sinyal. Semakin banyak perubahan sinyal yang terjadi maka semakin kecil nilai PSNR, jika nilai PSNR dibawah nilai 30, maka *noise* dapat dirasakan.
7. Pesan yang dapat disisipkan hanya berupa file text, sedangkan file lainnya tidak dapat disisipkan karena metode yang digunakan adalah LSB (*Least Significant Bit*).

Adapun beberapa saran yang dapat diberikan oleh penulis dari kesimpulan yang dikemukakan diatas, adalah sebagai berikut.

1. Untuk proses penyisipan pesan yang disisipkan dengan ukuran sangat besar, sangat membutuhkan waktu yang sangat lama. Sehingga diharapkan bahwa adanya algoritma atau metode lain yang dapat mempercepat proses penyisipan dan pengungkapan.
2. Untuk proses steganografi pada *file audio wav*, diharapkan ada metode lain yang dapat meminimalisir perubahan *bit* yang terjadi pada *file* media, salah satunya dengan metode *Chaos*. Metode ini nantinya dapat mencari bit yang sama dengan pesan rahasia, sehingga secara garis besar tidak ada perubahan bit yang terjadi di dalam *file* media.
3. Pesan rahasia yang disisipkan ke dalam *file audio wav*, hanya berupa pesan teks saja, sehingga diharapkan adanya pengembangan dan metode lain yang lebih sempurna untuk proses penyisipan. Untuk penyisipan *file* lainnya dapat menggunakan metode DWT.

4. Metode LSB yang disarankan untuk digunakan tidak hanya terbatas pada perubahan pada 1 bit LSB saja, tetapi ada kemungkinan untuk merubah bit-bit yang lain. Selain itu penelitian ini dapat juga dijadikan bahan penelitian lebih lanjut untuk mengetahui perbandingan kualitas *file audio wav* dengan menggunakan metoda penyisipan lebih dari 1 bit.

DAFTAR PUSTAKA

Alatas, Putri. 2009. Implementasi Teknik Steganografi Dengan Metode LSB Pada Citra Digital. *Skripsi*. Universitas Gunadarma.

Ardhyana, Alfebra S., Asep Juarna. 2008. Aplikasi Steganografi Pada Mp3 Menggunakan Teknik LSB.*Skripsi*. Universitas Gunadarma.

Arubusman, Yusrian R. 2007. Audio Steganografi. *Skripsi*. Universitas Gunadarma.

Asri, Indra. 2008. Audio Steganografi Menggunakan Teknik LSB Dengan Media Audio Wav. *Skripsi*. Universitas Andalas.

Chasanah, Zulfah. 2009. Steganografi Pada File Audio Mp3 Untuk Pengamanan Data Menggunakan Metode Least Significant Bit (LSB). *Skripsi*. Universitas Islam Negeri Malang.

Darwis, Dedi. 2015. Implementasi Stegano-grafi Pada Berkas Audio WAV Untuk Penyisipan Pesan Gambar Menggunakan Metode Low Bit Coding. *Expert*. Vol 5. No 1. 6-11.

Eriyani, Indah M. 2008. Steganografi Pada Gambar Bitmap Menggunakan Metoda LSB (Least Significant Bit). *Skripsi*. Universitas Andalas.

Fahlevi, Muhammad R. 2012. Aplikasi Steganografi Untuk Menjaga Kerahasiaan Informasi Menggunakan Bahasa Pemrograman Java. *Skripsi*.Universitas Gunadarma.

Gunawan, Andy., Nugroho Agus Haryono, Junius Karel T. 2007. Penyembunyian Pesan Text Pada File Wav Dengan Metode Least Significant Bit. *Jurnal Informatika*. Vol 3. No 1. 16-19.

Indriyono, B.V. 2016. Penerapan Keamanan Penyampaian Informasi Melalui Citra dengan Kriptografi Rijndael dan Steganografi LSB. *Citec Journal*. Vol 3, No 3. 228-241.

[11]Lovebbi, D. Z. Sudirman. 2012. Rancang Bangun Aplikasi Steganografi dengan Metode Least Significant Bit di Audio pada Sistem Operasi Android. *Ultimatics*, Vol 4, No 1. 7-16.

Munir, Rinaldi. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Informatika : Bandung.

[13]Pressman, Roger S. 2005. *Rekayasa Perangkat Lunak*. Andi. Yogyakarta

Sukrisno & Utami. 2007. Implementasi Steganografi Teknik EOF Dengan Gabungan Enkripsi Rijndael, Shift Cipher dan Fungsi Hash MD5. *Seminar Nasional Teknologi*. D1-D16.

Utami, Ema. 2009. Pendekatan Metode Least Bit Modification Untuk Merancang Aplikasi Steganograhpy Pada File Audio Digital Tidak Terkompresi. *Jurnal DASI*. Vol 10. No 1.

Wijaya, H.dan Wilianti, K. 2013. Penyisipan Teks dengan Metode Low Bit Coding Pada Media Audio Menggunakan Matlab 7.7.0. *Jurnal TICOM*. No 3. 28-35.