Jurnal Ilmiah Sinus (JIS) Vol : 17, No. 1, Januari 2019
ISSN (Print) : 1693-1173, ISSN (Online): 2548-4028

STEGANOGRAFI PADA CITRA BITMAP MENGGUNAKAN METODE LEAST SIGNIFICANT BIT BERSILANG UNTUK TEKS TERENKRIPSI BASE64

Albert Christie Giovani[1)], Yustina Retno Wahyu Utami [2)]; Teguh Susyanto[3)]
[1)] Program Studi Sistem informasi, STMIK Sinar Nusantara
[2)] Program Studi Teknik Informatika, STMIK Sinar Nusantara
[3)] Program Studi Teknik Informatika, STMIK Sinar Nusantara
[1)] albertchristiegiovani@gmail.com; [2)]yustina_retno@sinus.ac.id; [3)]teguhsusyanto@gmail.com

ABSTRACT

The development of internet has become one of the most popular data communication media. The ease of use and complete facilities are the advantages possessed by the internet. However, along with the development of internet media and applications that use the Internet, crime on information system increases as well. With various illegal information-gathering techniques developing, many are trying to access information that is not their right. There are several security techniques for sending messages confidentially and securedly, one of which is known as steganography. This study combined steganography and cryptography. The message was encrypted first using base64 then inserted using the LSB Crossed method. This method was aimed at making the process of extracting messages by unauthorized ones not easy. Embedding message into images was using the last binary number of the RGB value of an image by randomizing the placement of binary numbers by integrating base64 coding so that it combined base64 messages which next the text messages would be encrypted. The measurement results in the stego image using PSNR (Peak Signal to Noise Ratio) showed that the image quality after the insertion process was > 50 db.

Keywords: Base64, Encrypt, Least Significant Bit, Steganography

I. PENDAHULUAN

Berbagai macam teknik yang digunakan untuk melindungi informasi yang dirahasiakan dari orang yang tidak berhak telah banyak dilakukan dalam upaya mengamankan suatu data penting dengan menggunakan sistem kriptografi yang melakukan enkripsi sebelum data penting tersebut ditransmisikan. Tindakan pengamanan menggunakan cara tersebut ternyata dianggap belum cukup dalam mengamankan suatu data karena adanya peningkatan kemampuan komputasi.

Berbeda dengan teknik kriptografi, steganografi menyembunyikan pesan rahasia agar bagi orang awam tidak menyadari keberadaan dari pesan yang disembunyikan. Teknik ini sering digunakan untuk menghindari kecurigaan orang dan menghindari keinginan orang untuk mengetahui isi pesan rahasia tersebut.

Penelitian tentang steganografi yang dikemukakan oleh (Farid, Nurhadiyono, & Rahayu, 2016) tentang "Implementasi Metode Steganografi Least Significant Bit Dengan Algoritma Hill Cipher Pada Citra Bitmap" mengemukakan bahwa Penggabungan metode kriptografi Hill Cipher dan steganografi *Least Significant Bit* (LSB) dapat meningkatkan keamanan pada pesan dalam bertukar informasi. Dalam penelitian ini, akan diusulkan suatu skema untuk menyembunyikan pesan informasi dengan penggabungan kriptografi dengan steganografi dengan metode LSB bersilang, yang mana penyembunyian pesan ini tidak sekedar menyisipkan bit terakhir, namun juga mengacak bit terakhir agar pesan informasi lebih sulit untuk diekstraksi oleh pihak yang tidak berhak.

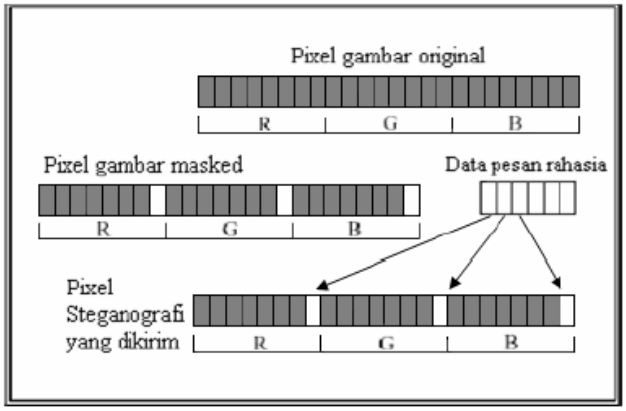
II. TINJAUAN PUSTAKA

2.1 *Steganografi*

Steganografi berasal dari Bahasa Yunani, yaitu steganos yang artinya "tulisan tersembunyi" dan graphos yang berarti tulisan. Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui(Mukharrom, Riza, & Maman, 2013).

2.2 Least Significant Bit (LSB)

LSB (*Least Significant Bit*) adalah bit yang mempunyai nilai paling rendah, atau bit yang berada pada posisi paling kanan. Penyisipan LSB dilakukan dengan memodifikasi bit terakhir dalam satu byte data. Bit yang diganti adalah LSB karena perubahan pada LSB hanya menyebabkan perubahan nilai *byte* satu lebih tinggi atau satu lebih rendah(Saptomo, 2008). Ilustrasi pesan yang disisipkan pada LSB ditunjukkan pada Gambar 1.



Gambar 1. Most significant bit (MSB) dan least significant bit (LSB) (Gupta, Gujral, & Aggarwal, 2012)

2.3 BASE64

Base64 adalah istilah umum untuk sejumlah skema serupa encoding yang mengkodekan data biner dengan memperlakukan itu numerik dan menerjemahkannya ke dalam basis 64 representasi.Istilah Base64 berasal dari konten MIME mentransfer encoding tertentu.

Skema Base64 encoding yang umum digunakan ketika ada kebutuhan untuk mengkodekan data biner yang perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani data tekstual. Hal ini untuk memastikan bahwa data tetap utuh tanpa modifikasi selama transportasi. Base64 digunakan umumnya dalam sejumlah aplikasi termasuk email melalui MIME, dan menyimpan data kompleks dalam XML(Nugraha & Gunadhi, 2016).

2.4 Visual Basic Net

Visual basic. NET adalah salah satu bahasa pemograman paling mudah dipelajari dan digunakan dalam waktu yang singkat, selain itu,visual basic.Visual basic (Mukharrom et al., 2013).

NET menyediakan lingkungan pengembangan high-level untuk membangun aplikasi-aplikasi pada NET Framework, dilingkungan inilah akan merasakan teknologi yang mampu menyerdehanakan pembuatan dan penyebaran aplikasi, selain itu visual basic .NET juga menawarkan generasi baru aplikasi berbasis windows dengan fitur-fitur yang tersedia melalui.NET Framework.

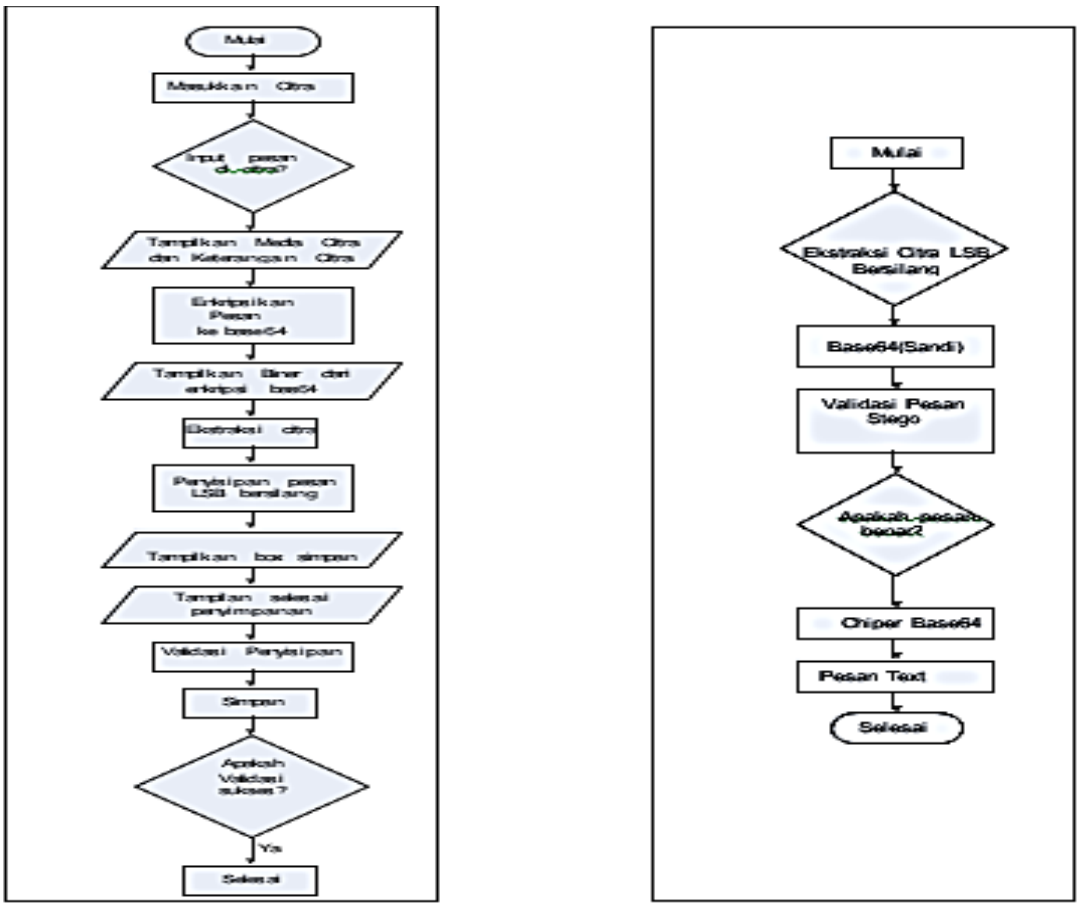
2.5 Penelitian Terkait

 DOI : https://doi.org/10.30646/sinus.v17i1.384

1. Menurut Mukharrom, Riza, dan Maman (2013), steganografi adalah ilmu dan seni menyembunyikan pesan ke dalam suatu wadah. Wadah dapat berupa gambar, berkas audio, atau berkas video. Kedua teknik tersebut dapat digabungkan sehingga menghasilkan sistem keamanan data yang tinggi. Dalam penelitian ini, diimplementasikan teknik kriptografi dan steganografi pada citra berformat bitmap dengan menggunakan metode *End Of File* (EOF).
2. Dalam artikel yang dikemukakan oleh Gupta, Gujral, dan Aggarwal (2012), Algoritma *Least Significant Bit* yang ada dianalisis dan ditemukan memiliki jumlah distorsi yang lebih banyak, jadi sebuah metode baru telah diusulkan "Enhanced Least Significant Bit (ELSB)". Ini meningkatkan kinerja metode LSB karena informasi tersembunyi hanya dalam satu dari tiga warna yaitu warna BLUE dari citra pembawa. Ini meminimalkan tingkat distorsi yang lalai terhadap mata manusia.
3. Menurut Zamani, Manaf, dan Abdullah (2012), studi ini menunjukkan bahwa untuk bit rate tertentu hampir ditentukan hasil PSNR yang dihitung rata-rata di baris terakhir. Dengan kata lain, jika ukuran file pesan hampir sama dengan kapasitas file host, penyisipan PSNR bergantung pada *bit per sample rate*. Host yang berbeda dan file pesan yang berbeda akan memiliki sedikit efek pada PSNR.

III. METODE PENELITIAN

Tahapan penyisipan dan ekstraksi pesan ditunjukkan pada Gambar 2.



(a) Penyisipan Pesan (b) Ekstraksi Pesan

Gambar 2. Alur penyisipan dan ekstraksi pesan

Sistem yang diusulkan dibangun menggunakan Bahasa pemrograman Visual Basic.NET. Pengujian fungsionalitas sistem menggunakan *black box testing*, sedangkan untuk algoritma yang diusulkan diukur dengan PSNR dan menghitung waktu proses, serta diuji dengan memberikan Noise Salt and Pepper.

IV. HASIL DAN PEMBAHASAN

4.1 LSB Bersilang

Secara umum prosses enkripsi dilakukan dengan menggunakan enkripsi BASE64 terhadap pesan yang akan disisipkan kedalam citra bitmap. Sedangkan penyisipan pesan dilakukan dengan menggunakan metode LSB bersilang dengan menggantikan bit-bit LSB pada citra bitmap dengan pesan hasil proses enkripsi.

Awalnya kunci yang dimasukkan oleh pengguna di enkripsi dengan menggunakan enkripsi BASE64 untuk memperoleh kunci baru sebesar 64 bit. Kemudian kunci ini digunakan untuk mengenkripsi pesan dengan menggunakan enkripsi BASE64. Proses enkripsi ini menghasilkan chipertext yang siap untuk disisipkan ke dalam citra.

Berikut ini proses Penyisipan dan Ekstraksi:

1. Sebelum proses penyisipan ke dalam citra, plaintext atau pesan diubah dulu chipertext ke dalam bilangan biner dengan menggunakan Tabel 1.

Tabel 1. Tabel konversi teks ke heksadesimal

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Dengan Tabel 1 diatas akan didapatkan nilai heksadesimal dari ***"SGFy"*** bisa diketahui, contoh :

S, koordinatnya 5,3
G, koordinatnya 4,7
F, koordinatnya 4,6
y, koordinatnya 7,9

Setelah mendapatkan angka heksadesimal kemudian konversi ke dalam bilangan biner dengan persamaan dibawah ini:

0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001
A = 1010
B = 1011
C = 1100
D = 1101
E = 1110
F = 1111

 DOI : https://doi.org/10.30646/sinus.v17i1.384

Sehingga didapatkan bilangan biner 01010011, 01000111, 01000110, 01111001.

2. Sebelum menyisipkan bilangan biner tersebut, citra masih berbentuk bilangan desimal dari nilai representasi warna RGB dalam setiap pixelnya. Sebagai contoh:
   R = 234
   G = 241
   B = 244

3. Kemudian ubah bilangan desimal tersebut menjadi bilangan biner,
   R = 234 ⟶ R = 11101010
   G = 241 ⟶ G = 11110001
   B = 244 ⟶ B = 11110100

4. Selanjutnya, proses penyisipan menggunakan citra berukuran 5x5 pixel, dengan mengambil sampel 3 bilangan biner paling awal adalah 111 kemudian disisipkan ke dalam LSB tiap piksel, contoh :

   01010011, 01000111, 01000110, 01111001

   R = 11101010 ⟶ R= 1110101**0**
   G = 11110001 ⟶ G= 1111000**1**
   B = 11110100 ⟶ B= 1111010**0**

5. Dengan algoritma persilangan LSB maka bilangan akhir biner yang disisipi biner chipertext akan bersilang, contoh :

   R = 11101010 ⟶ R = 11101010
   G = 11110001 ⟶ G = 11110000
   B = 11110100 ⟶ B = 11110101

4.2 Implementasi Antar Muka

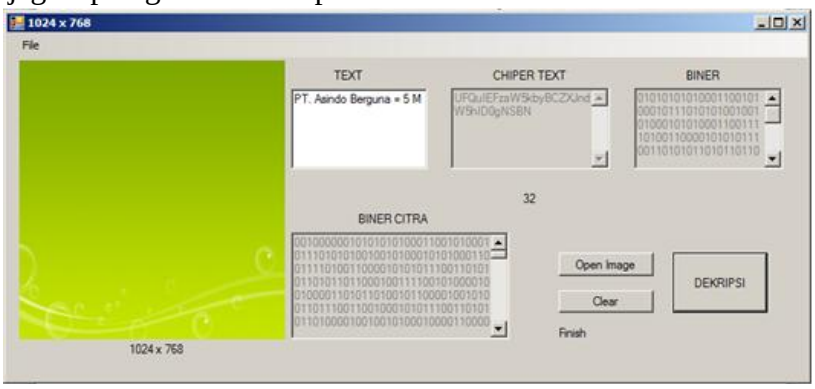
Pada antarmuka terdapat beberapa form yang berupa form ekstraksi, form boxdialog, dan form ekstraksi.

1) Form Penyisipan

Gambar 2. Penyisipan

2) Form Ekstraksi

Form ini berisikan program pengekstraksian dan dekripsi gambar bitmap, yang berisi juga input gambar bitmap



Gambar 3. Ekstraksi

.

4.3 Pengujian

Pengujian ini dilakukan untuk mengetahui kemampuan aplikasi ini dalam mengenkripsi dan mendekripsi pesan. Berikut hasil pengukuran waktu penyisipan dan ekstraksi pesan yang disajikan pada Tabel 2 dan Tabel 3.

Tabel 2. Penyisipan

Panjang Karakter	Waktu Penyisipan(detik)	Ukuran Citra
4	5,089	100x100
12	7,205	100x100
17	10,643	100x100
17	10,754	100x100
466	132,332	100x100
1047	error	100x100

Tabel 3. Ekstraksi

Ukuran Citra	Panjang Karakter	Waktu Ekstraksi (detik)
100x100	4	3,345
100x100	5	4,678
100x100	17	8,933
100x100	17	8,234
100x100	466	112.675

Dalam pengujian kualitas citra, maka akan dibandingkan antara citra asli dengan citra yang telah disisipi pesan dengan menggunakan nilai PNSR. ***Peak Signal to Noise Ratio* (PSNR)** merupakan perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR biasanya diukur dalam satuan desibel (dB).

$PSNR = 10 \log_{10}(C^{2max}/MSE)$ (1)

$MSE = \frac{1}{m \cdot n} \sum_{x=1}^{m} \sum_{y=1}^{n} (S_{xy} - C_{xy})^2$ (2)

- Cmax adalah nilai RGB terbesar pada keseluruhan citra.
- X dan y adalah koordinat suatu titik pada citra.

 DOI : https://doi.org/10.30646/sinus.v17i1.384

- M dan N adalah dimensi dari citra.
- S adalah citra tersisipi*(stego-image)*
- C adalah citra asli*(Cover image)*

Tabel 4. PSNR

Ukuran Citra	Panjang Karakter	Kualitas Citra
10x10	4	67,08
10x10	5	63,28
10x10	17	61,984
10x10	17	58,234
10x10	34	55,37
10x10	36	error

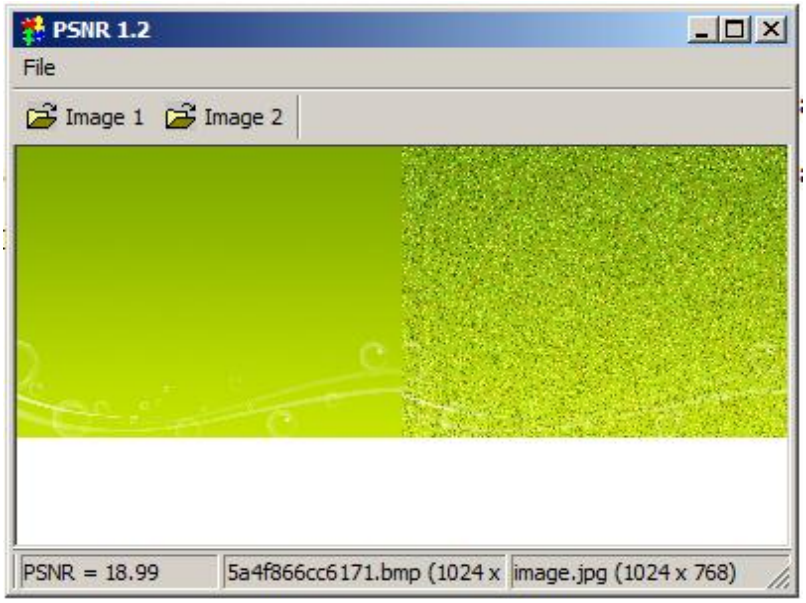
4.4 Daya Tampung

Tabel 5. Daya tampung

No	Ukuran Citra	Jumlah Biner	Daya Tampung Pesan karakter
1.	10x10	300	37,5
2.	20x20	1200	150
3.	100x100	30000	3750
4.	2000x2000	12000000	1500000

Daya tampung pesan didapatkan dari perhitungan RGB tiap pixel yang mempunyai 3 nilai, sehingga jika ukuran citra 10X10 maka jumlah daya tampung 10X10X3 = 300 yang berarti pesan memiliki penampungan sebesar 300 biner, dan memiliki tampungan pesan LSB 300/8=37,5 pesan teks.

4.5 Pengujian noise Salt and Pepper



Gambar 4. Pengujian citra asli dengan citra stego bernoise

Didapatkan nilai PNSR sebesar 18,99 db yang berarti citra stego tidak bagus karena kurang dari 30 db. Dan karena noise ini citra ini tidak bisa diekstraksi, yang berarti pesan di dalam citra sangat rentan untuk rusak dan tidak bisa diekstraksi

V. PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan, maka dapat ditarik kesimpulan dan saran sebagai berikut:

1. Metode LSB*(Least Significant Bit*) bersilang ini membuat pesan yang akan disisipkan membuat susah untuk dipecahkan dan jika ada analisis pegektsraksian akan juga sulit walaupun dengan metode perbandingan.
2. Dengan metode ini juga biner yang disisipkan akan sangat efisien, karena jika bilangan biner citra sama dengan bilangan biner pesan maka tidak akan diganti.
3. Enkripsi Base64 memperkuat pesan yang akan disisipkan menjadi Chiper text yang kalau di ekstraksi pesan tersebut tidak bisa dibaca dan enkripsi base64 bersilangan. Sehingga penganalisa akan semakin susah, karena akan di curigai isi pesan jika diekstraksi adalah enkripsi jenis baru.
4. Untuk daya tampung pesan sendiri, setiap pixel dapat menampung 3 bilangan biner , yang kalau memakai citra berukuran sangat kecil 5x5 akan bisa menampung 75 bilangan biner.
5. Validasi ditambahkan setelah penyisipan pesan yang berbentuk tanda, tanda ini bersifat *stealth*(siluman) karena hampir mirip ketika diekstraksi dengan citra tanpa pesan yang diekstraksi sehingga pesan jika diedit atau sudah rusak akan diketahui oleh sistem.
6. Untuk kualitas citra setelah disisipi dengan pengujian PSNR didapatkan hasil bahwa citra hasil sisipan pesan mempunyai kualitas rata-rata diatas 30 db atau sekitar 50-90 db, sehingga kualitas citra ini bagus dan tidak dicurigai ada pesan didalamnya.
7. Dalam prakteknya citra sangat mudah rusak sehingga pesan yang didalam citra juga akan rusak, seperti terkena noise, dan kompresi.

5.2 Saran

Berdasarkan penelitian ini, terdapat beberapa saran untuk penelitian lebih lanjut, yaitu:

1. Perlu dikembangkan algoritma yang lebih handal terhadap adanya noise dan manipulasi citra lainnya.
2. Alangkah lebih baik enkripsi base64 diperbaharui dengan enkripsi yang terbaru walau membutuhkan lebih banyak ruang dan menggunakan objek penelitian selain file citra.

DAFTAR PUSTAKA

Farid, N., Nurhadiyono, B., & Rahayu, Y. (2016). Implementasi Metode Steganografi Least Significant Bit Dengan Algoritma Hill Cipher Pada Citra Bitmap. *Techno.COM*, *15*(2), 109–116.

Gupta, S., Gujral, G., & Aggarwal, N. (2012). Enhanced Least Significant Bit algorithm For Image Steganography. *IJCEM International Journal of Computational Engineering & Management ISSN*, *15*(4), 22307893.

Mukharrom, E., Riza, I. R., & Maman, S. (2013). Aplikasi Steganografi Pada Citra

Berformat Bitmap Dengan Menggunakan Metode End of File. *Transien*, *2*(3), 1–9.

Nugraha, A. P., & Gunadhi, E. (2016). Penerapan Kriptografi Base 64 Untuk Keamanan Url (Uniform Resource Locator) Website Dari Serangan Sql Injection, 491–498.

S. Gupta, S., Gujral, G., & Aggarwal, N. (2012). Enhanced Least Significant Bit algorithm For Image Steganography. *IJCEM Int. J. Comput. Eng. Manag.*, *15*(4).

Saptomo, W. L. Y. (2008). Modifikasi Least Significant Bit dalam Steganografi. *Jurnal Ilmiah Sinus*, *6*, 1–8.

Zamani, M., Manaf, A., & Abdullah, S. (2012). Correlation between PSNR and bit per sample rate in audio steganography. *Conference on Signal …*, 163–168.