Focus **ACT**ion **O**f **R**esearch **M**athematic
Volume 6, No. 02, July – December, 2023
Page 109-121

ISSN: 2655-3511 (print)
ISSN: 2656-307X (online)
DOI:10.30762/f_m.v6i2. 1678

Implementation of Coding Theory with The Extended Hamming Code in Steganography

Nur Hamid[1], Nurul Jannah[2*]
[1,2] Math Education, Nurul Jadid University, Probolinggo, Indonesia
**Corresponding author. Kotaanyar, 67293, Probolinggo City, Indonesia.*
E-mail: *nurhamid@unuja.ac.id* [1)]
nuruljannah05102001@gmail.com [2*)]

Keywords

Kode Linier, Kode Hamming yang diperluas, steganografi

Linear code, extended Hamming code, steganography

ABSTRACT

Keamanan data perlu diperhatikan agar terhindar dari kebisingan saat pengiriman data atau pembobolan data oleh peretas. Dalam artikel ini, diimplementasikan pengamanan data dengan menggunakan gabungan teori koding dan steganografi. Kode yang digunakan adalah kode Hamming yang diperluas [8,4]. Media yang digunakan adalah gambar *grayscale* berukuran 360×640 dengan panjang karakter yang disisipkan sebanyak 50,100,150,200,250 dan 300 karakter dengan 6 kali percobaan. Sebagai tambahan, ditampilkan nilai PSNR yang diperoleh dan juga waktu komputasi.

Data security needs to be considered to avoid noise when sending data or data breaches by hackers. In this article, we implement data security using a combination of coding theory and steganography. The code used is an extended Hamming code [8.4]. The media used is a grayscale image of size 360×640 *of lengths 50,100,150,200,250 and 300 characters with 6 attempts. In addition, we show the PSNR result obtained and also the computation time.*

INTRODUCTION

Technological developments make it easier for today's people to carry out various activities at one time. People can exchange messages and data without boundaries. Of course, exchanging messages and transferring data can be done easily.

Even so, it cannot be denied that there is a possibility that errors may occur when sending messages. This is caused by

noise or data breaches by irresponsible parties.

Based on Tempo records from January to September 2022, there have been seven major cases of suspected personal data leakage in Indonesia. An example of this case is the Bank Indonesia data leak case which occurred in January 2022. There were approximately 16 computers at the Bank Indonesia Branch Office in Bengkulu that experienced a leak which was resolved by the National Cyber and Crypto Agency (BSSN). In fact, this not only happened at the Bengkulu Branch Office, but also in 20 other cities with a total of more than 52 thousand documents with a size of 74.82 GB and originating from 200 computers (Nurhadi, 2022).

One way to correct errors when sending messages is with coding theory. The message scheme with coding theory can be seen in Figure 1.

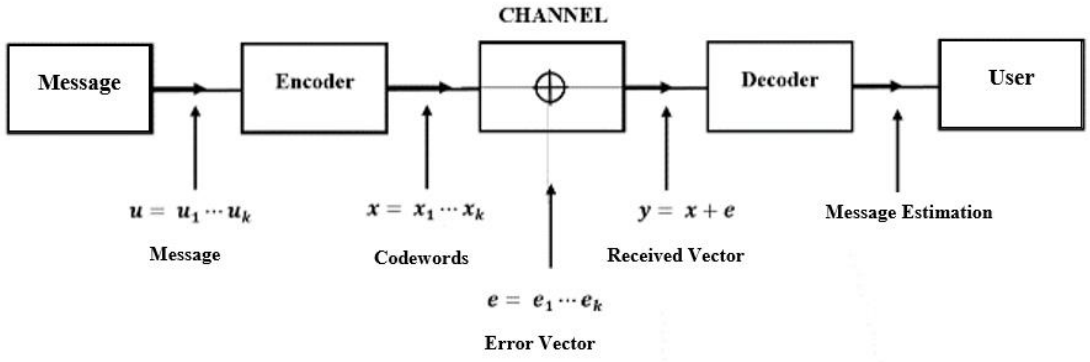

Figure 1. Communication Scheme

Coding theory began with Shannon's research in 1948. Shannon explained the initial concept of coding theory to detect errors and make it possible in correcting, so that the fundamental problem in coding theory was determining what message was sent based on what it was received. So far Coding theory has developed rapidly. It gives various types of code, one of them is linear code (Riyanto, 2020).

Linear code is a type of code that is very often studied because it is easy to understand and of course has linear properties (MacWilliams & Sloane, 1977). One of the linear codes commonly used is Hamming Code. A Hamming code is a code to correct errors that can occur when computer data is sent, moved, or stored. The Hamming code [7.4] and the extended Hamming code [8.4] are the Hamming codes that are often used.

Hamming codes are also often used in steganography, a branch of the science of embedding messages in images or other media. For example, the Hamming code [7,4] was used in the steganography process involving video media (Mstafa & Elleithy, 2014).

Steganography involving linear code was also discussed by Rahman, Khalil, and Xun Yi (Rahman et al., 2021). They developed a reversible biosignal steganography approach using the method error correction based on extended binary Golay code. This method was used to extract secret messages and reconstruct original biosignals to avoid damage when outsourcing to the cloud and avoid other stakeholders.

A more specific discussion regarding steganography was carried out by Subramanian, Elharrouss, Al-Maadeed, and Bouridane. They explored and discussed steganography with various deep learning methods or deep learning techniques (Subramanian et al., 2021). The techniques used for steganography on image media were divided into three, namely traditional methods, based on Convolutional Neural Network and General Adversarial Network. Steganography involving linear code was also discussed in (Um et al., 2020), (P et al., 2020), (Malathi & Kumar, 2021), (Vien et al., 2021) and (Wu et al., 2020).

Based on the previous description, in this article we implement an inserting of message model on an image involving the extended Hamming code over a binary field. With this code, messages sent can be corrected if an error occurs. In addition, inserting messages in images can protect messages from unwanted things.

METHODS

This research aims to determine the use of the extended Hamming code or Hamming code [8,4] in steganography and to determine the simulation results of inserting messages in images using the expanded Hamming code over binary fields.

This research uses the Least Significant Bit (LSB) method, a steganography algorithm used to hide secret messages in media, such as images, videos or other media. Confidential information is embedded in less significant bits in media files. In principle, the lowest value or Least Significant Bit (LSB) usually has the least contribution in determining the overall value of the data for each bit of digital data (Minarni & Redha, 2020). The media used is a grayscale image with the

length of the character inserted as many times as the number of characters tried.

Linear Code

The linear code used in this article is a subspace of a vector space $\mathbb{F}_2^n$. This linear code has several properties of liniearity and properties related to a subspace in general. (MacWilliams & Sloane, 1977). An $[n, k]$ linear code hase 2^k elements called codewords where k is the dimension of the linear code and n is the length of each codeword.

An $[8,4]$ *Extended Hamming Code*

Definition 1. A binary hamming code H_r of length $n = 2^r - 1$ $(r \geq 2)$ has a parity check matrix H whose columns are nonzero binary column vectors of length r. The code H_r is an $[n, k = 2^r - 1 - r, d = 3]$ coce.

The hamming code is a perfect *single-error-correcting* code (MacWilliams & Sloane, 1977). This code is one of error-correcting code which can detect some errors, but can only correct an error. Based on Correcting and Detecting theorem (MacWilliams & Sloane, 1977), the the hamming code of $d = 3$ can correct $\frac{1}{2}(3 - 1) = 1$ error.

The Hamming code implemented in this article is the $[8, 4]$ extended Hamming code with a generator matrix G, that is

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

First, a message (m_1, m_2, m_3, m_4) of length k bit $(k = 4)$ is encoded by appending 4 parity bit (p_1, p_2, p_3, p_4), the formed codeword is a combination such as $(m_1, m_2, m_3, m_4, p_1, p_2, p_3, p_4)$

The Hamming code is a linear code with two matrices, those are a parity check matrix H and a generator matrix G which work for encoding and decoding. In an encoding process, every message (m_1, m_2, m_3, m_4) will be multiplied by the generator matrix G in modulo 2. The obtained result is a codeword X which consists of 8 bits.

The codeword will be used for inserting the message. The encoding model using the $[8, 4]$ extended hamming code is

$$X = M\,G$$

where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The decoding stage is to detect the received message. The message is multiplied by the transpose of the parity

check matrix H in modulo 2. The decoding model using the $[8,4]$ extended Hamming code is

$$\boldsymbol{Z} = XH^{tr}$$

where

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A result from a vector $Z = (z_1, z_2, z_3)$ should contain (000) if there is no error in the message. Otherwise, we need an error-correcting process.

Example 1. Assume that we have a message $M_1\ (1,0,1,1)$ and the extended $[8,4]$ Hammign code. The encoding and decoding process are the following.

1. In the encoding stage, compute $X = M_1\ G$, then we have a codeword $X = 10110100$.
2. In the decoding stage, to obtain the correct message, then the vector Z should be zero. For the first, assume that $X = 10110100$ without any error, then Z will be (000) which means that there is no error in the message.
3. The second assumption is the following. When the message is received, there is an error. However, in this research, we assume that every message received has no error, so we do not need this second assumption.

The Flow of Message Encoding and Insertion Program

The flow of encoding and inserting a message in a digital image is the following.

1. Receive the input message which will be inserted.
2. Convert the message into binary numbers of length 8 bits.
3. In this stage, the encoding process will be done using the $[8,4]$ extended Hamming code with the parity check matrix H and the generator matrix G. This code has the minimum distance $d = 3$, it means that it can detect 2 errors and correct an error with the information rate $\frac{1}{2}$. The binary numbers obtained are divided into 2 parts and we let them be M_1 and M_2. Use the equation $X = M\,G$ to obtain the codeword. In this case, 1 character will be 2 codewords.
4. In this step, we do the process of inserting the message using LSB in a grayscale image. Two codewords will be inserted in each row of the grayscale image side by side. Do XOR the codewords with the element in the image. The result obtained the change the element in the same position.

5. Reconstruct the image with the the message. We say it as a stego image.

In general, the flowchat of the program is the following.

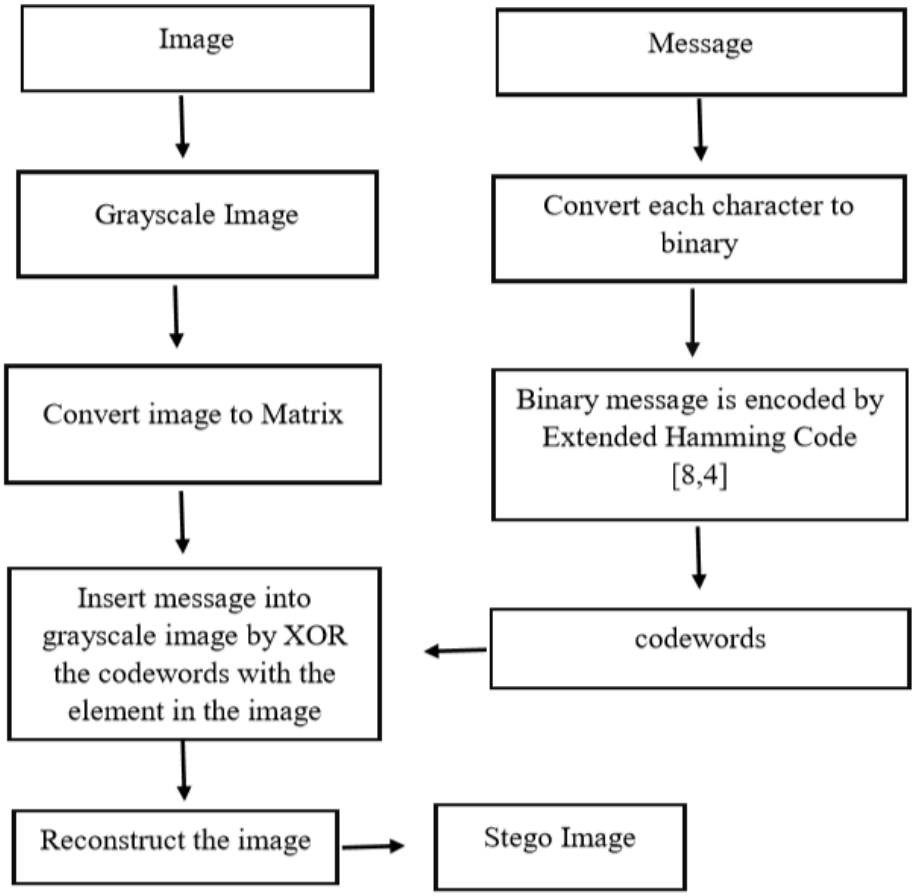

Figure 2. The Flow of Encoding and Inserting Message Program

The Flow of Extracting and Decoding Message

The flow of extacting the message from the image stego and decoding the message are the following.

1. Find the elements of stego image matrix which includes the message.
2. With the assumption that the receiver also has the grayscale image, do XOR the element by the element in the grayscale image with the same position. We obtain 2 codewords as the result.
3. The decoding process is the same as Example 1. In this simulation, we assume that there is no noisy or error in the message. Then, the decoded result is (000), it means that there is no error.
4. The message character is obtained from the first 4 bits in each codeword in every row. Merge the 4 bits from the

codeword into binary numbers of length 8 in each row.

5. Convert the binary number into a character and arrange into a received message.

In general, the program flow of extracting and decoding the message can be seen in Figure 3.

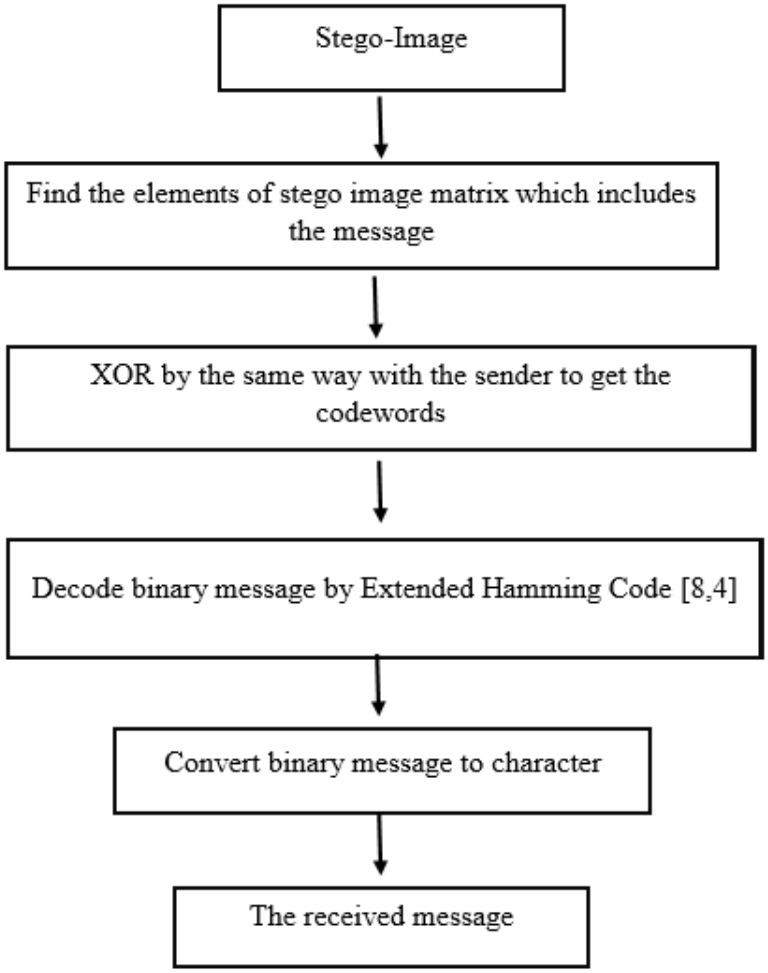

Figure 3. The Flow of Extracting and Decoding Message Program

RESULTS AND DISCUSSION

The data simulation design is in accordance with the program flow in Figures 2 and 3. The code used is the $[8, 4]$ extended Hamming code over $\mathbb{F}_2$ or $GF(2)$. The data used is text to be inserted into an image. The message text is converted into binary numbers using *American Standard Code for Information Interchange* (ASCII) table which is in Figure 4.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 4. American Standard Code for Information Interchange (ASCII)

The image used for inserting the message is a grayscale image. An image in Python is two dimensional list whose elements are integers in $0, \cdots, 255$. To manipulate the image, we use OpenCV (http://opencv.org) which is available in *Google Colab*.

From the simulation program run, we have have the stego image, Pick Signal to Noise Rasio (PSNR), and computation time.

Figure 5. Grayscale

Result of Stego Image

The grayscale image used here is of size 360×640. So, there is a limitation in determining the length of the message in the simulation. The number of message character which can be applied in the image of size 360×640 should be less than the number of row of the grayscale image matrix, that is 359. The lengths of character used in the simulation program are $50, 100, 150, 200, 250$ and 300 characters with 6 tries. Figure 5 is the grayscale image used in the simulation.

Focus **ACT**ion **O**f **R**esearch **M**athematic
Volume 6, No. 02, July – December, 2023
Page 109-121

ISSN: 2655-3511 (print)
ISSN: 2656-307X (online)
DOI:10.30762/f_m.v6i2. 1678

Based on the encoding and the inserting the message in Figure 4, we have the stego image with the number of characters N which can be seen in Figure 6.

N = 50, PSNR = 56.760 dB

N = 100, PSNR = 55.349 dB

N = 150, PSNR = 54. 218 dB

N = 200, PSNR = 53.307 dB

N = 250, PSNR = 52.750 dB

N = 300, PSNR = 52.022 dB

Figure 6. Steganography Result

Pick Signal to Noise Ratio (PSNR)

Pick Signal to Noise Ratio or PSNR is a measurement showing how much an enhancement can overcome the effects of interference (Roopaei dkk., 2016). PSNR is used to see the comparison of image

values with images that have the embedded messages or noise. The unit of PSNR is decibel (dB). PSNR is defined by

$$PSNR = 10 \log 10 \left(\frac{C_{max}^2}{MSE}\right)$$

where

C_{max} : $maximum\ value\ of\ pixel$

MSE : $Mean\ Square\ Error$

Mean Square Error (MSE) is often used as a measure of the level of accuracy for images and videos (Keleş dkk., 2021). However, the value depends on the dynamic scale or range of the image or video used. MSE is defined as follows. (Sajati, 2018).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{i=0}^{n-1} [I(i,j) - K(i,j)]^2$$

where

m, n : the width, the height of the image

$I(i,j)$: the original pixel value at coordinates (i,j)

$K(i,j)$: the pixel value of the embedded image at the coordinates (i,j)

Based on the simulation done, the value of PSNR of each stego image can be seen in Table 1.

Tabel 1. PSNR of Each Stego Image

Gambar	Banyaknya karakter yang disisipkan (N)	PSNR
Stego 1	50	56.760 dB
Stego 2	100	55.349 dB
Stego 3	150	54. 218 dB
Stego 4	200	53.307 dB
Stego 5	250	52.750 dB
Stego 6	300	52.022 dB

Based on the value of PSNR showed in Table 1, the quality of the image resulted is very good. The value of PSNR which is high shows that the quality of the reconstructed image has a low MSE (Subramanian dkk., 2021)

Computation Time

Computation time is obtained from the entie simulation program being run for each experiment. A diagram of computation time in seconds can be seen in Figure 7.

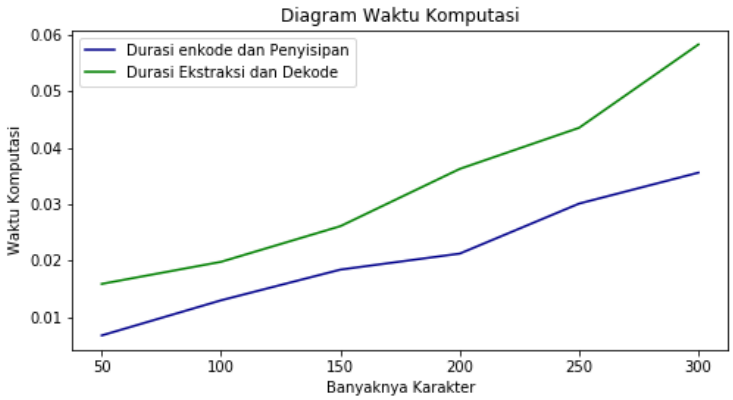

Figure 7. Computation Time

Based on Figure 7, we can conclude that the more characters are inserted, the more time we take to compute the program.

CONCLUSION

Based on the simulation, we obtain some conclusions. The encoding using the [8,4] extended Hamming code can be used for steganography process. The resulting image is not uch different from the grayscale original image, the inserted message can be read correctly with a limit of many characters less than the number of rows of the grayscale image matrix. The number of characters inserted is directly proportional to the image changes and the computation time required.

This research uses an encoding using the [8,4] extended Hamming code and a digital image as message insertion media. For the next research, we hope we can use other media and other codes to implement coding theory in steganography.

ACKNOWLEDGEMENT

We thank to the referees of the modelling competition held by Universitas Mulawarman. This article is an improvement on the work delivered during the competition.

REFERENCES

Keleş, O., Yılmaz, M. A., Tekalp, A. M., Korkmaz, C., & Dogan, Z. (2021). *On the Computation of PSNR for a Set of Images or Video* (arXiv:2104.14868). arXiv. http://arxiv.org/abs/2104.14868

MacWilliams, F. J., & Sloane, N. J. A. (1977). *The Theory of Error-Correcting Codes* (Vol. 16).

Malathi, P., & Kumar, T. G. (2021). An efficient data hiding technique in image using binary Hamming code

along with particle swarm optimisation. *International Journal of Intelligent Systems Technologies and Applications*, *20*(2), 167. https://doi.org/10.1504/IJISTA.2021.119048

Minarni, M., & Redha, R. (2020). Implementasi Least Significant Bit (LSB) dan Algoritma Vigenere Chiper pada Audio Steganografi. *Jurnal Sains dan Teknologi: Jurnal Keilmuan dan Aplikasi Teknologi Industri*, *20*(2), 168. https://doi.org/10.36275/stsp.v20i2.268

Mstafa, R. J., & Elleithy, K. M. (2014). A highly secure video steganography using Hamming code (7, 4). *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*, 1–6. https://doi.org/10.1109/LISAT.2014.6845191

Nurhadi. (2022). Inilah 7 Kaus Dugaan Kebocoran Data Pribadi Sepanjang 2022, (Online), (https://nasional.tempo.co/read/1632043/inilah-7-kasus-dugaan-kebocoran-data-pribadi-sepanjang-2022), diakses 08 September 2022

P, M., M, A. S., Paliwal, A., & T, G. K. (2020). Maximizing the Embedding Efficiency Using Linear Block Codes in Spatial and Transform Domains. *Procedia Computer Science*, *167*, 302–312. https://doi.org/10.1016/j.procs.2020.03.227

Rahman, M. S., Khalil, I., & Yi, X. (2021). Reversible Biosignal Steganography Approach for Authenticating Biosignals Using Extended Binary Golay Code. *IEEE Journal of Biomedical and Health Informatics*, *25*(1), 35–46. https://doi.org/10.1109/JBHI.2020.2988449

Riyanto, M. Z. (2020). Kode Linear untuk Deteksi dan Koreksi Kesalahan Penulisan dalam Huruf Hijaiyah. *JURNAL FOURIER*, *9*, 49–58. https://doi.org/10.14421/fourier.2020.92.49-58

Roopaei, M., Eghbal, M. K., Shadaram, M., & Agaian, S. (2016). Noise-Free Rule-Based Fuzzy Image Enhancement. *Electronic Imaging*, *28*(13), 1–5. https://doi.org/10.2352/ISSN.2470-1173.2016.13.IQSP-225

Sajati, H. (2018). The Effect of Peak Signal to Noise Ratio (PSNR) Values on

Object Detection Accuracy in Viola Jones Method. *Conference SENATIK STT Adisutjipto Yogyakarta*, *4*. https://doi.org/10.28989/senatik.v4i0.139

Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. *IEEE Access*, *9*, 23409–23423. https://doi.org/10.1109/ACCESS.2021.3053998

Um, L. E., Jouhari, H., García-Planas, M. I., & Souidi, E. M. (2020). *Convolutional Codes and Steganography under Linear Systems Theoretical Point of View*. *22*.

Vien, Q.-T., Nguyen, T. T., & Nguyen, H. X. (2021). Deep-NC: A secure image transmission using deep learning and network coding. *Signal Processing: Image Communication*, *99*, 116490. https://doi.org/10.1016/j.image.2021.116490

Wu, X., Yang, C.-N., & Liu, Y.-W. (2020). A general framework for partial reversible data hiding using hamming code. *Signal Processing*, *175*, 107657. https://doi.org/10.1016/j.sigpro.2020.107657