Hiding multiple secret information using dynamic image bit manipulation

Ahmad Fashiha Hastawan[1*], Risma Septiana[2]

[1]Dept. of Electrical Engineering, Universitas Negeri Semarang, Semarang, Indonesia
[2]Dept. of Computer Engineering, Universitas Diponegoro, Semarang, Indonesia

*Corresponding author's email: ahmad.fashiha@mail.unnes.ac.id

Abstract. In the New Paradigm of Industry 4.0, the main concern of high flow information transmission is security disruption. The transmission process should secure the information transmitted. Especially for secret information found out by nobody. One of the methods used to improved information security is the information hiding process. The hiding technique often applied is Steganography. In this technique, a multimedia data format such as the digital image embeds information. The problem occurs when the secret information is a digital image. Stored an image into another image will cause data distortion and the information will be damaged. These problems can be solved by improving the steganography method for digital images. Improvement of the steganography method in this study uses a bit manipulation technique on the Least Significant Bit (LSB) method. The secret information represented by the digital image will be hidden in a container image. The hiding technique method worked by manipulating the intensity value of the image in the sequence of binary bit format, so the container image intensity value can be used to hide many digital images. The result of this research is a steganography technique allowed to store three images in one image container at one time. The results of the improvement method can protect the intensity values of the container image from some failure. The results of this study also show that data integrity is maintained even though the value of image intensity changed by the bits of hidden images.

1. Introduction

Evolution of the technology in era Industry 4.0 stipulates communication between two or more hardware devices. The communication is the process of data transmission from one device to the others. The data format transmitted is not only text, but also in a multimedia format such as video, audio, and image. The most popular multimedia data currently used is the image. This proof is shown that image processing is commonly used as a supporting technique in Industrial Era 4.0[1]. There is no guarantee of the security of data transmission. The preventive action for the multimedia data theft is using steganography method to hide the secret data.

Steganography technique is storing secret data into multimedia data, including images. The concept of the technique is to substitute data into data storage. The substitution data called data embedded and the data storage called container image. The result of the substitution is a stego image. If this concept uses two or more images embedded in one container image, then one container image can represent more than two images. This combination method causes two or more images can be hidden in one image which means there are multiple images represented by one image [2]. The problem occurs in the images combining process when the image size will be larger and there is a broken pixel value. Bit manipulation is one of the most popular techniques used to overcome the problem. This research

adopted bit manipulation based on the Least Significant Bit (LSB). The proper bit location had to be the main concern, so the stego image obtained is the optimal result. The goals of this research focused on hidden multiple images in one container image, so the pixels in the container image can use optimally. This research applied the proposed method using a color image that has three layers called Red (R), Green (G), and Blue (B) as the container image and the binary images as the embedded image [3].

Previous studies [4] [5] explain that in the digital transformation era as known as industry 4.0, the security of data transmission must be considered. One security method uses the steganography concept, so the secret data will be hidden and unreadable. Numerous research about steganography shows that the technique can combine data multimedia and other data using the principle of hidden images technique. But, previous research can only hide one image to one container image [6]. The LSB substitution method is the most method improved in the steganography technique. The [7] combines LSB with the method to determine the right location to substitute the bit. The [6] uses a sliding value algorithm to optimize the stego image. [8] and [9] mixed the LSB with the various algorithms of coding to improve the result. The most evaluation of the success steganography studies containers some discussions about parameters such as Robustness, Imperceptibility, Payload Capacity, and PSNR & MSE. The value of compression evaluates imperceptibility and payload capacity. The integrity of data is evaluated by robustness and PSNR & MSE [10]. The success parameter of the hidden process in the steganography is the invisible intensity changing [11]. This research provides an overview of the hidden three images at one container image without distortion of the images.

2. Research Methodology

This research uses bit manipulation in an image to hide multiple images. The basic steganography method referenced is the Least Significant Bit. The proposed method applies the embedding process and extraction process.

2.1. Least Significant Bit Method (LSB) for Steganography Process

Steganography is a technique that used to hide a secret message in the multimedia data format such as image, audio, text, video, etc. Steganography image uses two key component values of an image to hide various data formats. The components called pixel and intensity value. One of steganography image methods, LSB is the basic technique for embed value in an intensity pixel. The advantage of this technique is less distortion of the container image, but the lack comes from the image embedded as the secret image. The secret image will be damaged because the embed process will manipulate the sequence of a bit to establish a stego image.

LSB is one of the steganography methods. LSB worked in the binary representation. This method worked based on embedding binary value to the container image. There are rules in the LSB method [12]:

1. The message saved should be a sequence of binary values.
2. The embedding message starts from the right sequence called MSB (Most Significant Bit). The followed rules:
   - If message_bit=1 and intensity of image container=odd or If message_bit=0 and intensity of image container =even, intensity of stego-image=intensity of image container.
   - If message_bit=1 and intensity of image container=even, intensity of stego-image=intensity of image container + 1
   - If message_bit=0 and intensity of image container=odd, intensity of stego-image=intensity of image container-1
3. Save the Stego image

2.2. Purposed Method

The purposed method worked in the embedding image and extracting image. This method still uses binary representation. Every process utilizes all of the pixels optimally. Data saved in an image not only the image intensity but also the size of the pixels. The goal of saving all data is to ease the extracting process, so the final result shows restoration of the hidden image that will be the same as before.

2.2.1. Embedding Image Algorithm

The embedding process is the algorithm to embed the secret image to a container image. There are some steps for this algorithm:

Step 1, the first process for starting this algorithm initializes the pixel size both embedded image and container image. The process of the initialization shows below:

1. Pixel size = width x height
2. If pixel size embedded image < pixel size container image
3. Then, continue the process
4. Else, stop the process

Step 2, Convert the size of the embedded image to the binary representation. E.g
Pixel Size = 1024 x 800. The binary representation = 10000000000 and 01100100000

Step 3, Separate the container image based on three components of the color layer Red (R), Green (G) and Blue (B)

Step 4, Take the 24^{th} first pixel on each color layer to store the size of the embedded image. The storage consists of width and height value for each image. Twelve pixels are used to the width (N) and 12 remaining pixels will be used to the height (M). Assume that the maximum image size for this experiment is 4095 x 4095 pixels. The user can adjust this size.

Step 5, Insert the value size of the width and the height of embedded images into the determining pixel. This process uses LSB method. The first 12^{th} is for the M value and the next 12^{th} is for the N value.

Step 6, Change the whole sequences of the pixel R G B layer from the matrix size N x M to the 1 x MN format, where the MN is the length (M x N)

Step 7, Insert the first embedded image to the R layer using the 25^{th} -pixel position

Step 8, Insert the second embedded image to the G layer using the 25^{th} -pixel position

Step 9, Insert the third embedded image to the B layer using the 25^{th} -pixel position

2.2.2. Extraction Image Algorithm

Extraction is the restoring image from the container. This method will extract multiple embedded images saved in each colour layers

Step 1, Split the RGB image container based on the color layer Red (R), Green (G), and Blue (B)

Step 2, Get the LSB value in the 24^{th} first pixel for each layer (R, G, and B)

Step 3, Convert the 12^{th} first bit binary value to the width of image M, and the next 12^{th} bit as the hight (N). So, the results provide three sizes of the embedded images

Step 4, Change the format of layer pixel from matrix N x M to the array 1 x MN, where the MN is the length of M x N

Step 5, Take the LSB value from layer R in the 25^{th} bit sequence and then change the format to array 1 x MN appropriate with M x N size, so the first embedded image is found

Step 6, Take the LSB value from layer G in the 25^{th} bit sequence and then change the format to array 1 x MN appropriate with M x N size, so the first embedded image is found

Step 7, Take the LSB value from layer B in the 25^{th} bit sequence and then change the format to array 1 x MN appropriate with M x N size, so the first embedded image is found.

2.2.3. Evaluation Parameters

Evaluation of the steganography technique is measured based on some factors as follow[12]:

1. Mean Square Error (MSE)

Comparing the stego image with the actual container image is the process of MSE measurement. MSE can be calculated using the equation 1:

$$MSE = \frac{\sum_{X}^{M}\sum_{Y}^{N}[f1(x,y) - f2(x,y)]^2}{MxN} \quad (1)$$

The result of MSE calculation indicates the value of noise for an image. Low MSE value means that stego image have only a bit damage.

2. Peak Signal To Noise Ratio (PSNR)

PSNR is the comparation between intensity image and noise value. PSNR can be count using the equation 2:

$$PSNR = 20\log_{10}\frac{255}{\sqrt{MSE}} \quad (2)$$

The result of PSNR is used to analyze quality of the stego image. Higher value of PSNR shows the good quality of an image.

3. Root Mean Square Error (RMSE)

RMSE is a error measurement for an image. RMSE can be calculated using the equation 3:

$$RMSE = \sqrt{\frac{\sum_{X}^{M}\sum_{Y}^{N}[f1(x,y) - f2(x,y)]^2}{MxN}} \quad (3)$$

The result of RMSE shows an error accuration. Low value of RMSE means that the error does not affected in the image.

3. Result and Discussion

The results of this research evaluate the stego image that is established by container image and multiple embedded images. The discussion focus on the effect of multiple images embedded in an image container. Various embedded images are the sample of the secret image hidden. The image container consists of some color images. The effect of the embedding process will change the intensity value not only in the embedded image but also in the container image. Therefore, the performance analysis must consider intensity value in the image embedded and container image.

3.1. Image Embedded

Image Embedded is an image hidden in the container image. This experiment used various images in two bits of binary representation. There are only two values in the binary representation i.e zero as black color and one as white color. Using binary image representation eased to know the alteration on the intensity of the bits because the embedding process used binary manipulation. This research experiment uses sixteen binary images. Some examples of the embedded images can be shown in figure 1.

(a) img 1.png (b) img 5.png (c) img 6.png (d) img 12.png

Figure 1. Four example of images would be embedded

Figure 1 shows four images as the example of embedded images. Each image has differences in binary value composition. All parameter component of embeded image are resumed in table 1.

Table 1. Parameter Summary of Images Embedded

No	Data	Data Size	Pixel Size
1	img1.png	4.14 KB (4,247 bytes)	626x585
2	img2.png	3.33 KB (3,415 bytes)	612x403
3	img3.png	2.52 KB (2,587 bytes)	593x289
4	img4.png	1.82 KB (1,869 bytes)	420x230
5	img5.png	1.42 KB (1,462 bytes)	420x188
6	img6.png	1.51 KB (1,549 bytes)	420x188
7	img7.png	1.76 KB (1,806 bytes)	198x175
8	img8.png	1.79 KB (1,834 bytes)	198x175
9	img9.png	1.72 KB (1,762 bytes)	198x175
10	img10.png	2.25 KB (2,308 bytes)	325x341
11	img11.png	2.24 KB (2,298 bytes)	325x341
12	img12.png	1.74 KB (1,787 bytes)	285x274
13	img13.png	3.62 KB (3,707 bytes)	422x530
14	img14.png	1.60 KB (1,640 bytes)	260x244
15	img15.png	1.42 KB (1,456 bytes)	198x211
16	img16.png	1.42 KB (1,459 bytes)	198x211

Table 1 shows parameter values of sixteen image embedded. All images have various pixels size that affect in the data size. Larger pixel sizes will result the bigger data size.

3.2. Image Container

Image Container is a carrier image hidding an image embedded. This research uses 24-bit RGB image. The image consists of three layers image, that is R, G, and B. The advantages of using multiple layers image is all of the layers used to save image. So, if one layer can save one image, then using three layers image can be used to save three images. The examples of container image can be shown in figure 2.

(e) img 1.png (f) img 5.png (g) img 6.png

Figure 2. Four example of images would be embedded

Figure 2 shows some examples of the container images. This experiment used eight types of container images that have various components of intensity value. The differences appear from the various colour in each container image. Table 2 resumes the data of eight container images.

Table 2. Data Summary of Container Images

No	Data	Ukuran Piksel	Ukuran Data
1	Container1.png	1.89 MB (1,985,371 bytes)	1600x901
2	Container2.png	1.62 MB (1,705,291 bytes)	1600x901
3	Container3.png	2.03 MB (2,131,829 bytes)	1600x901
4	Container4.png	1.58 MB (1,660,857 bytes)	1600x901
5	Container5.png	1.28 MB (1,351,013 bytes)	1600x901
6	Container6.png	1.52 MB (1,602,678 bytes)	1600x901
7	Container7.png	1.92 MB (2,024,444 bytes)	1600x901
8	Container8.png	1.26 MB (1,330,971 bytes)	1600x901

Table 2 shows all of data from container images used in the experiment. The same pixel sizes were used in the experiment process. All of the pixel were manipulated by various pixel from image embedded.

3.3. Histogram of Stego Pixel

The main process of this research is bit manipulation process. This process used to insertion the embedded images to a container image. This process will manipulate placement of the bits. Figure 3 shows example explains three embedded images that will be hidden in a container image.

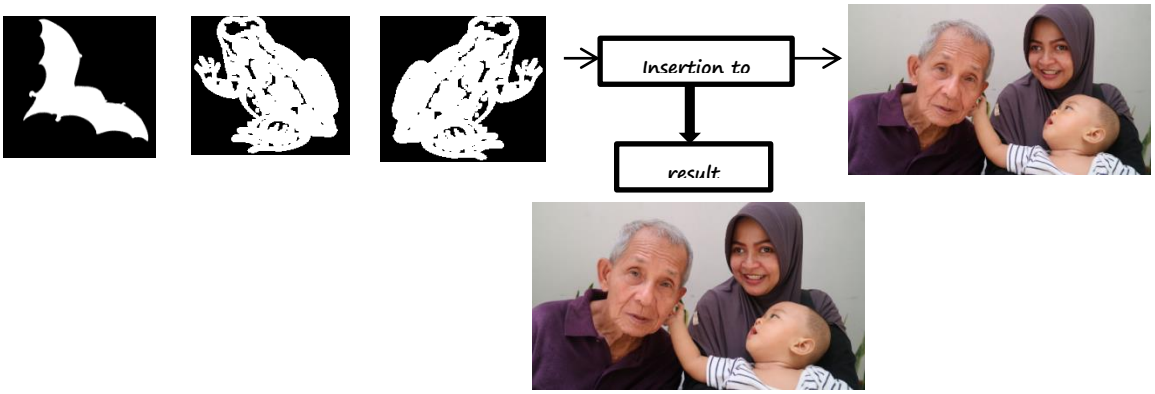


Figure 3. The process of hidden multiple image in one image container

Figure 3 shows the embedding process of multiple images. Three images can be saved in one container image. Bit manipulation worked in the insertion process. Every pixel in the container image saved the embedded image data. Although the insertion bit had changed the intensity of every pixel, the image result still appeared the same image. But, the changing of intensity value can be seen in the image histogram. Each colour layer embed one of embedded image, so the histogram devide by the three colour layer R, G, and B. Comparasion of the histogram from original image and stego image is shown in table 3.

Table 3 Comparasion of sample colour layers original image and stego image

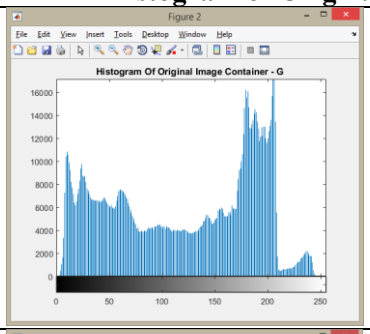
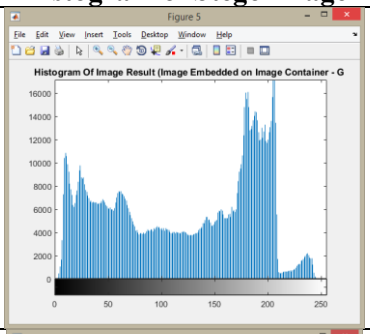
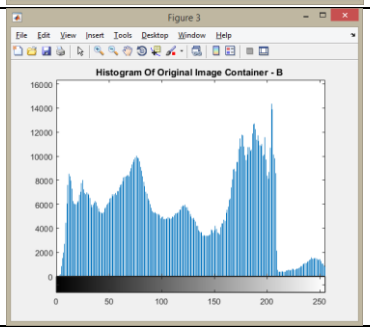
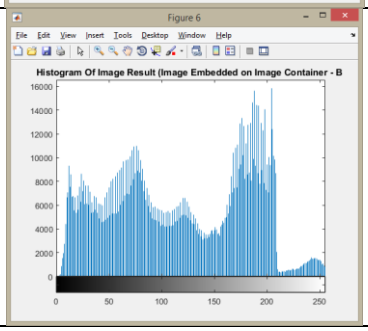
No	Histogram of Original Image	Histogram of Stego Image
1		
2		

Table 3 shows the histogram comparison of original image and stego image.The changing of intensity value can be seen from the differences of histogram composition value. Bit manipulation process worked in each colour layer causes addition value in image intensity value. The histogram shows the addition value using shadow lines as the addition lines of the original histogram. Although the embedded process change the intensity value, the changing is invisible from human eyes.

3.4. Encryption Result

The changing of intensity value affects the quality of stego image as the image result. Quality of stego image can be measured using MSE, PSNR, and RMSE value. Table 5 shows the result of the quality measurement for each stego image

Table 4 The result of measurement of the quality for each stego image

	Image Container	Image Embedded			Stego Image			Measurement Quality		
		Image 1	Image 2	Image 3	Image Name	Image Size	Image Dimension	MSE	PSNR	RMSE
1	Container1.png	Img1.png	Img16.png	Img9.png	Result1.png	1.91 MB (2,008,025 bytes)	1600x901	0.0527	60.9120	0.2296
2	Container1.png	Img2.png	Img15.png	Img10.png	Result2.png	1.91 MB (2,004,993 bytes)	1600x901	0.0465	61.4589	0.2156
3	Container2.png	Img3.png	Img14.png	Img11.png	Result3.png	1.68 MB (1,762,328 bytes)	1600x901	0.0400	62.1094	0.2000
4	Container2.png	Img4.png	Img13.png	Img12.png	Result4.png	1.68 MB (1,771,555 bytes)	1600x901	0.0460	61.5001	0.2146
5	Container3.png	Img5.png	Img12.png	Img13.png	Result5.png	2.05 MB (2,158,352 bytes)	1600x901	0.0440	61.6924	0.2099
6	Container3.png	Img6.png	Img11.png	Img14.png	Result6.png	2.05 MB (2,150,969 bytes)	1600x901	0.0294	63.4522	0.1714
7	Container4.png	Img7.png	Img10.png	Img15.png	Result7.png	1.60 MB (1,681,862 bytes)	1600x901	0.0216	64.7869	0.1470
8	Container4.png	Img8.png	Img9.png	Img16.png	Result8.png	1.59 MB (1,669,329 bytes)	1600x901	0.0128	67.0468	0.1133
9	Container5.png	Img9.png	Img8.png	Img1.png	Result9.png	1.35 MB (1,420,023 bytes)	1600x901	0.0502	61.1219	0.2241
10	Container5.png	Img10.png	Img7.png	Img2.png	Result10.png	1.35 MB (1,416,303 bytes)	1600x901	0.0454	61.5619	0.2130
11	Container6.png	Img11.png	Img6.png	Img3.png	Result11.png	1.54 MB (1,620,133 bytes)	1600x901	0.0418	61.9208	0.2044
12	Container6.png	Img12.png	Img5.png	Img4.png	Result12.png	1.53 MB (1,612,484 bytes)	1600x901	0.0294	63.4482	0.1714
13	Container7.png	Img13.png	Img4.png	Img5.png	Result13.png	1.96 MB (2,056,501 bytes)	1600x901	0.0460	61.5075	0.2144
14	Container7.png	Img14.png	Img3.png	Img6.png	Result14.png	1.95 MB (2,054,560 bytes)	1600x901	0.0365	62.5081	0.1910
15	Container8.png	Img15.png	Img2.png	Img7.png	Result15.png	1.32 MB (1,385,375 bytes)	1600x901	0.0373	62.4103	0.1932
16	Container8.png	Img16.png	Img1.png	Img8.png	Result16.png	1.33 MB (1,401,007 bytes)	1600x901	0.0511	61.0488	0.2260

Table 4 shows the result of quality measurement from stego image. Stego image is established from one container image and three embedded image. The experiment used sixteen variation with the varies image using sixteen embedded image. The quality measurement result provides the good value of MSE, PSNR, and RMSE. The good quality is obtained from the lower MSE and RMSE, but the PSNR become higher.

3.5. Decryption Result

The quality measurement is also the extraction of embedded image after the hidden process. The result of image-extracted quality is shown in table 5.

Table 5 Image Extracted Quality Measurement

No	Image Extracted														
	Image 1					Image 2					Image 3				
	Image Size	Pixel Size	MSE	PSNR	RMSE	Image Size	Pixel Size	MSE	PSNR	RMSE	Image Size	Pixel Size	MSE	PSNR	RMSE
1	4.14 KB (4,247 bytes)	626x585	0	Inf	0	1.42 KB (1,459 bytes)	198x211	0	Inf	0	1.72 KB (1,762 bytes)	198x175	0	Inf	0
2	3.33 KB (3,415 bytes)	612x403	0	Inf	0	1.42 KB (1,456 bytes)	198x211	0	Inf	0	2.25 KB (2,308 bytes)	325x341	0	Inf	0
3	2.52 KB (2,587 bytes)	593x289	0	Inf	0	1.60 KB (1,640 bytes)	260x244	0	Inf	0	2.24 KB (2,298 bytes)	325x341	0	Inf	0
4	1.82 KB (1,869 bytes)	420x230	0	Inf	0	3.62 KB (3,707 bytes)	422x530	0	Inf	0	1.74 KB (1,787 bytes)	285x274	0	Inf	0
5	1.42 KB (1,462 bytes)	420x188	0	Inf	0	1.74 KB (1,787 bytes)	285x274	0	Inf	0	3.62 KB (3,707 bytes)	422x530	0	Inf	0
6	1.51 KB (1,549 bytes)	420x188	0	Inf	0	2.24 KB (2,298 bytes)	325x341	0	Inf	0	1.60 KB (1,640 bytes)	260x244	0	Inf	0
7	1.76 KB (1,806 bytes)	198x175	0	Inf	0	2.25 KB (2,308 bytes)	325x341	0	Inf	0	1.42 KB (1,456 bytes)	198x211	0	Inf	0
8	1.79 KB (1,834 bytes)	198x175	0	Inf	0	1.72 KB (1,762 bytes)	198x175	0	Inf	0	1.42 KB (1,459 bytes)	198x211	0	Inf	0
9	1.72 KB (1,762 bytes)	198x175	0	Inf	0	1.79 KB (1,834 bytes)	198x175	0	Inf	0	4.14 KB (4,247 bytes)	626x585	0	Inf	0
10	2.25 KB (2,308 bytes)	325x341	0	Inf	0	1.76 KB (1,806 bytes)	198x175	0	Inf	0	3.33 KB (3,415 bytes)	612x403	0	Inf	0
11	2.24 KB (2,298 bytes)	325x341	0	Inf	0	1.51 KB (1,549 bytes)	420x188	0	Inf	0	2.52 KB (2,587 bytes)	593x289	0	Inf	0
12	1.74 KB (1,787 bytes)	285x274	0	Inf	0	1.42 KB (1,462 bytes)	420x188	0	Inf	0	1.82 KB (1,869 bytes)	420x230	0	Inf	0
13	3.62 KB (3,707 bytes)	422x530	0	Inf	0	1.82 KB (1,869 bytes)	420x230	0	Inf	0	1.42 KB (1,462 bytes)	420x188	0	Inf	0
14	1.60 KB (1,640 bytes)	260x244	0	Inf	0	2.52 KB (2,587 bytes)	593x289	0	Inf	0	1.51 KB (1,549 bytes)	420x188	0	Inf	0
15	1.42 KB (1,456 bytes)	198x211	0	Inf	0	3.33 KB (3,415 bytes)	612x403	0	Inf	0	1.76 KB (1,806 bytes)	198x175	0	Inf	0
16	1.42 KB (1,459 bytes)	198x211	0	Inf	0	4.14 KB (4,247 bytes)	626x585	0	Inf	0	1.79 KB (1,834 bytes)	198x175	0	Inf	0

Table 5 shows the result of the stego image extracted. The measurement of image extracted quality aims to know the distortion effect to the embedded images. The result shows that the distortion does not influence the embedded images. It appears from the zero value of MSE and RMSE. PSNR value give the infinite result. It means the images have the same intensity value as the before embedding process.

4. Conclusion

Three layers colour of image RGB can use for hidden multiple image optimally. This paper measure the success of proposed method through the quality of stego image. Stego images have low MSE and RMSE value, then the PSNR value become higher. It means that the quality of stego image give the good result. Although the intensity values change the composition of bits sequence, the damage of intensity values were rarely found. The hidden process can use all of pixels in container image. All embedded images data saved in the image container. The embedded images hidden were not having

damage. The proof of the statement is proven by MSE and RMSE values that become zero, then the PSNR value become infinite. The size of embedded data was not influence the process of hidden image. The hidden process will be success throughout the size of container image is larger than the embedded image.

References

[1] T. P. Raptis, A. Passarella, and M. Conti, "Data Management in Networked Industrial Environments: State of the Art and Open Challenges," *ArXiv Prepr. ArXiv190206141*, 2019

[2] X.-W. Li, W.-X. Zhao, J. Wang, and Q.-H. Wang, "Multiple-image hiding using super resolution reconstruction in high-frequency domains," *Opt. Commun.*, vol. 404, pp. 147–154, Dec. 2017

[3] R. M. H. Nguyen and M. S. Brown, "RAW Image Reconstruction Using a Self-contained sRGB–JPEG Image with Small Memory Overhead," *Int. J. Comput. Vis.*, vol. 126, no. 6, pp. 637–650, Jun. 2018

[4] A. K. Bairagi, R. Khondoker, and R. Islam, "An efficient steganographic approach for protecting communication in the Internet of Things (IoT) critical infrastructures," *Information Security Journal: A Global Perspective*, vol. 25, no. 4–6, 2016

[5] I. Sittón and S. Rodríguez, "Pattern Extraction for the Design of Predictive Models in Industry 4.0," in *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017*, vol. 619, F. De la Prieta, Z. Vale, L. Antunes, T. Pinto, A. T. Campbell, V. Julián, A. J. R. Neves, and M. N. Moreno, Eds. Cham: Springer International Publishing, 2018, pp. 258–261

[6] S. Singh and V. K. Jain, "Generalized Method for Stego Pixel Optimization Using Slide Value Algorithm," in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–5

[7] R. Shanthakumari, "A SPATIAL DOMAIN BASED IMAGE IN IMAGE HIDING SCHEME USING PARTICLE SWARM OPTIMIZATION."

[8] S. V. T., "Analysis Of The Steganography In Image With Mobile Computing," *INTERNATIONAL SCIENTIFIC JOURNAL "INDUSTRY 4.0,"* vol. II, no. 2, pp. 80–82, 2017

[9] D. Rawat and V. Bhandari, "A steganography technique for hiding image in an image using lsb method for 24 bit color image," *Int. J. Comput. Appl.*, vol. 64, no. 20, 2013

[10] M. Mohit, "An Enhanced Least Significant Bit Steganography Technique," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 5, no. 6, pp. 1721–1725, 2016

[11] R. Roy and S. Changder, "Quality Evaluation of Image Steganography Techniques: A Heuristics based Approach," *Int. J. Secur. Its Appl.*, vol. 10, no. 4, pp. 179–196, Apr. 2016

[12] P. N. Andono, T. Sutojo, and Muljono, *Pengolahan Citra Digital*, 1st ed. Yogyakarta: Andi, 2017