

# Performance of Chaos-Based Encryption Algorithm for Digital Image

Suryadi MT, Eva Nurpeti, Dhian Widya

Department of Mathematics, Universitas Indonesia, Depok, 16424, Indonesia

\*Corresponding author, email: {yadi.mt, eva.nurpeti}@sci.ui.ac.id

## Abstract

Presentation of information in digital form is highly vulnerable against information abusing. Digital image is one of digital information which is frequently becomes a target of crime. Therefore, reliable, secure, and fast security techniques are required in digital image information. In this study, chaos-based encryption algorithm for digital image is built to improve endurance from brute force and known plaintext attack. The algorithm use logistic map as a random number generator for key stream. According to test and analysis, this algorithm has key space of  $10^{30}$ , key sensitivity up to  $10^{-16}$ , the key stream is proved random, and the distribution of pixels value from encrypted image is proved uniform. So, it can be concluded that, the algorithm is very difficult to be cracked by brute force attack and also known plaintext attack.

**Keywords:** chaos, logistic map, encryption algorithm, digital image

## 1. Introduction

Performance of an algorithm can be seen from the algorithm endurance security against attacks and computation time. The traditional cipher like *Data Encryption Standard* (DES), *International Data Encryption Algorithm* (IDEA), *Advanced Encryption Standard* (AES), and *Rivest-Shamir-Adleman Algorithm* (RSA) require a large computational time and high computing power. However, the image encryption ciphers are preferable which take lesser amount of time and at the same time without compromising security [1],[2]

To provide a better solution for the security problem of digital image, a number of image encryption techniques have been proposed including the chaos -based image encryption. These techniques provide a good combination of speed, high security, complexity, and computational power, etc [3]-[6] Chaos-based encryption also been extensively studied by researchers because of its superior in safety and complexity [2],[3],[6]-[12].

Chaos is the type of behavior of a system or function that is random, sensitive to initial values, and ergodicity. Function that has chaos properties was called chaos function. Chaos function have been proved very suitable to design facilities for data protection [4],[5],[13]. With these properties, chaos function can be used as a random number generator. One of the simple function that shows the chaos properties is the logistic equation or commonly called the logistic map. Logistic map function is defined as a function  $L_\lambda: \mathbb{R} \rightarrow \mathbb{R}$ ,  $L_\lambda(x) = \lambda x(1 - x)$  which is a function of one variable  $x$  and  $\lambda$  is a fixed parameter. The value of variable  $x$  in the interval  $(0,1)$  and  $\lambda$  in the interval  $(0,4)$ . Meanwhile, the presentation of logistic map function is in the form of iterative. It is :

$$x_{n+1} = \lambda x_n(1 - x_n) \quad (1)$$

where  $n = 0, 1, 2, 3, \dots$  and  $x_0$  is the initial value of iteration [2],[3].

In this paper, we will discuss about security of digital image using chaos -based encryption method, by using the logistic map as a chaos function. Testing of algorithm was done based on the encryption and decryption average time, size of the key space, and key sensitivity analysis. Beside that, we conducted a randomness analysis of key stream which generated by these algorithm, and uniform distribution analysis of pixel values in the image that has been encrypted. The analysis was carried out to see the resistance against brute force attack and known plaintext attack.

## 2. Encryption Algorithm

Encryption algorithm for digital image in this paper uses logistic map as a chaos function. The sequence of the process of securing the digital image can be seen in Figure 1 and process to regain access to the original digital image can be seen in Figure 2:

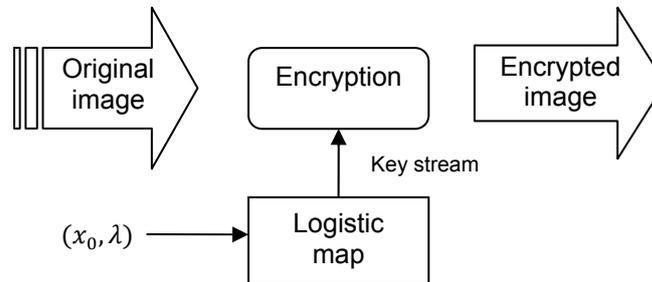


Figure 1. Encryption Process

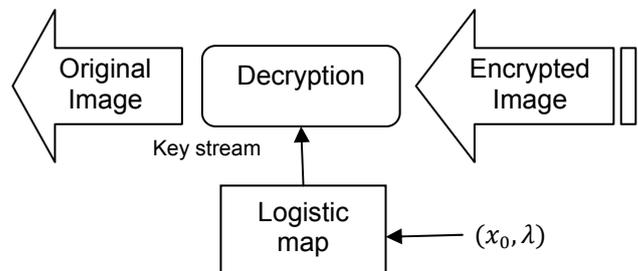


Figure 2. Decryption Process

Figure 1 and Figure 2 shows the flow in securing digital images. Function key stream generator is the logistic map. The input of this algorithm are the original image and the key, the key is  $x_0$  and  $\lambda$ . The output is an image that has been encrypted or image has been safe. To regain access to the original image, then we do the decryption process as shown in Figure 2. Input of the decryption process are the image that has been encrypted and the key. The key used in the decryption process is the same during the encryption process. The output is the original image. Encryption algorithm is described in the step 1 to step 5 [12]:

Step 1 : Insert the key  $(x_0, \lambda)$  and original image with  $n \times m$  size

Step 2 : Do 200 times iteration the logistic map equation (1) and we will get  $x_{200}$  decimal fractions.

Step 3 : Check condition.

Step 3a : If yes, then do 3 times the logistic map iteration and we will obtain the results are decimal fractions, such as  $x_{203}$ .

Step 3b : If not, so the encryption process is done for all part of image and we will obtain encrypted image.

Step 4 : Check whether the iteration is the last or not.

Step 4a : If yes, then do a real transformation to an integer, with procedures:

Select the first 15 number behind the decimal from decimal fraction that has been placed before ( for example  $x_{203}$  ), that are the result of 3 iterations logistic map. Then divide 15 number to  $p$  integer with each integernya has 3 points. Then take as much  $\text{mod}(n.m,p)$  integer. Do operation  $\text{mod} 256$  to each integer, so we get  $\text{mod}(n.m,p)$  byte integer. 1 byte this integer number is called key stream (KS).

Step 4a.1 : Take the pixel value information at each grayscale as much as  $\text{mod}(n.m,p)$ . Each 1 byte information of the image is called P.

Step 4a.2 : Do step 5  $\text{mod}(n.m,p)$  times.

Step 4.b : If not, then do transformation from real to integer, like in the step 4a, but take by  $p$  integer. Then take  $p$  integer. Do operation mod 256 to each integer, so we get  $p$  bytes integer number or  $p$  KS.

Step 4b.1 : Take the pixel information at each pixel grayscale by  $p$ . Each 1 byte information of the image is called  $P$ .

Step 4b.2 : Do the step 5 by  $p$  times.

Step 5: Do bitwise XOR operation on each byte integer number with every byte image data. Otherwise, do:  $KS(j) \oplus P(j)$ . Back to Step 3.

### 3. Results and Analysis

The test data used are cat.jpg digital image grayscale and color, with different sizes are presented in Table 1.

Table 1. Test Data Image

Test Data	Image Show	Image Type	Pixel Size
Data 1.		Cat.jpg	80 × 60
Data 2.			320 × 240
Data 3.			640 × 480
Data 4.			1280 × 960
Data 5.			2560 × 1920
Data 6.		8.jpg	80 × 60
Data 7.			164 × 123
Data 8.			178 × 132
Data 9.			269 × 200
Data 10.			315 × 234
Data 11.		Birthday.jpg	96 × 128
Data 12.			152 × 203
Data 13.			211 × 281
Data 14.			256 × 341
Data 15.			300 × 400

All test data in Table 1 will be used in the encryption process to be shown time encryption and decryption of the algorithm. Then it will be testing the durability of the chaos-based encryption algorithm. The first test is the test of resistance to brute force attacks with key sensitivity analysis and determination of the size of the key space. A second test is the test of resistance to known plaintext attack by randomness of key stream analysis and histogram analysis.

#### 3.1 Encryption and Decryption Time Analysis

Tests toward all digital image test data, performed using the same key value for both encryption and decryption process. The keys that used are  $x_0 = 0.1$  and  $= 4$ . Based on the test results of the cat.jpg grayscale and color digital image, we obtained an average process time of encryption and decryption which is shown in Figure 3, where each image is done by 5 attempts experiment (Data 1 to Data 5).

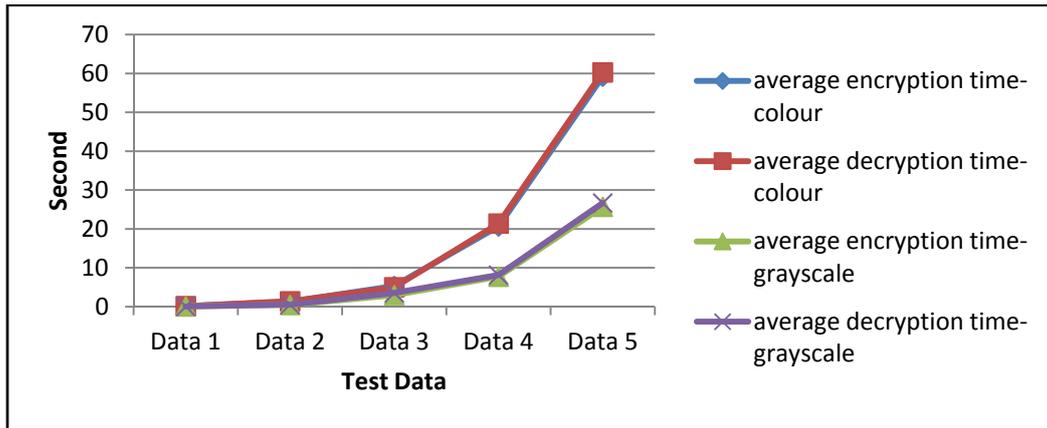


Figure 3. Encryption and Decryption Processing Time for cat.jpg using the proposed algorithm

Shown in Figure 3 that the time between the encryption and decryption process is not much different or relatively similar. For color images takes time encryption and decryption process is longer when compared to the grayscale image. That is because, the encryption process is done for each component of each grayscale red, green and blue, so it takes a longer process than just doing the encryption process on a grayscale image.

Time analysis from this proposed algorithm is better if compare with the algorithm by Gao. et. al [10]. Those were shown in Figure 4, Figure 5, Figure 6, and Figure 7. The test data which was used were Data 6 to Data 15 (Table 1).

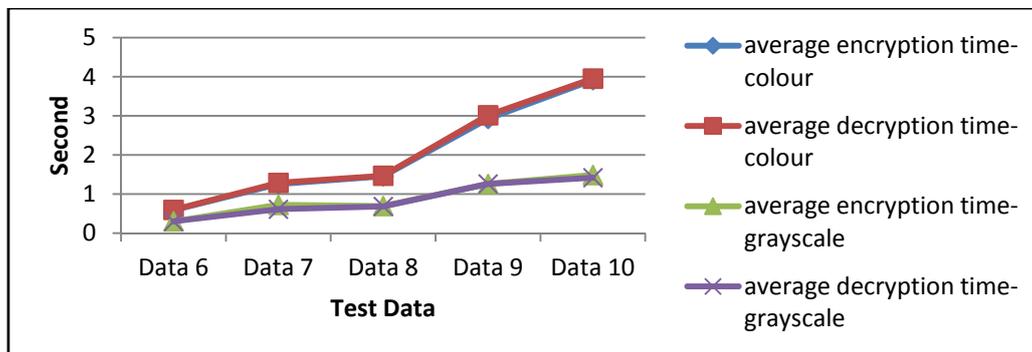


Figure 4. Encryption and Decryption Processing Time for 8.jpg using the algorithm by Gao,et.al.

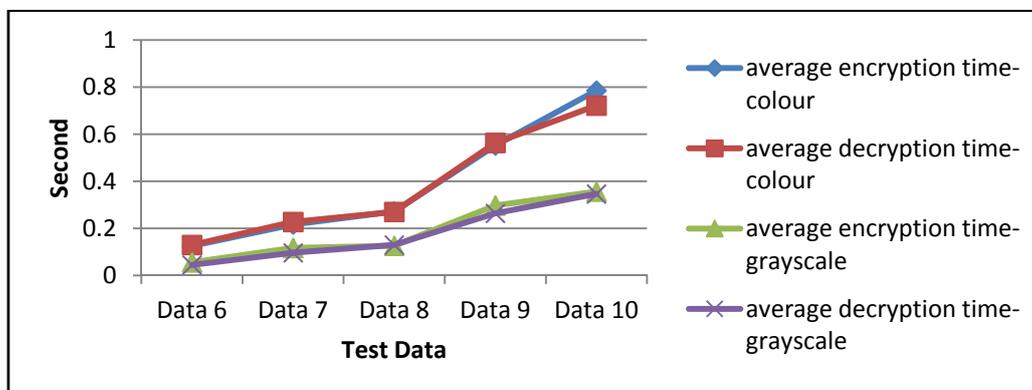


Figure 5. Encryption and Decryption Processing Time for 8.jpg using the proposed algorithm



Figure 6. Encryption and Decryption Processing Time for birthday.jpg using the algorithm by Gao, et. al.

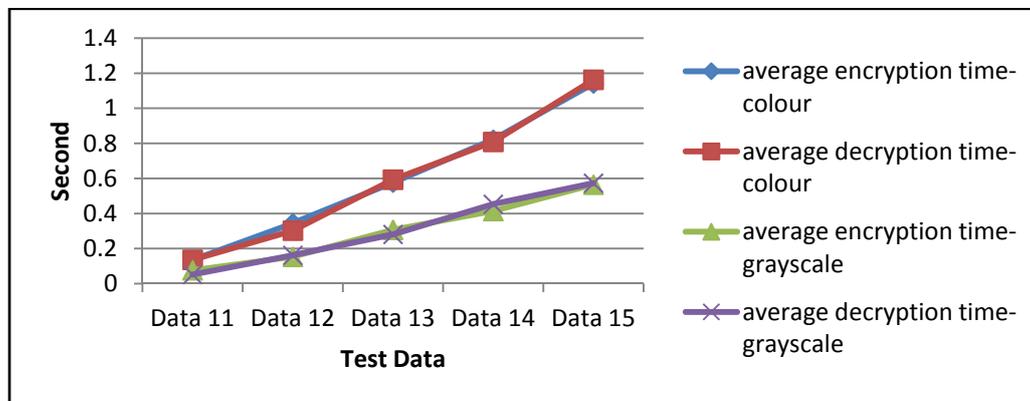


Figure 7. Encryption and Decryption Processing Time for birthday.jpg using the proposed algorithm

Based on the Figure 4 to Figure 7, it is shown that the encryption and decryption processing time on Figure 5 and Figure 7 is better then encryption and decryption processing time on Figure 4 and Figure 6. In terms of encryption and decryption processing time, the algorithm in this proposed algorithm is better than algorithm that was used by Gao H, et. al.[10].

**3.2 Key Sensitivity Analysis**

The value of the key that is used is always same for each digital image test data in this paper. While the decryption process will be tested with various different key value. The results are presented in Figure 8.

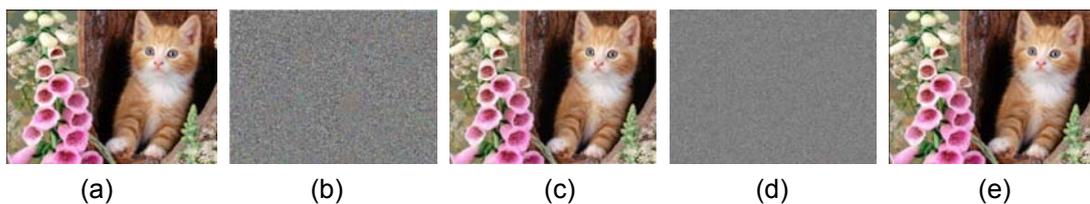


Figure 8. The Results of Cat.jpg. (a) Plain image; (b) Cipher image of (a); (c) Decrypted image; (d) Decrypted image with Difference  $x_0 = 10^{-16}$ ; (e) Decrypted image with Difference  $x_0 = 10^{-17}$

In Figure 8b and Figure 8c are shown the results of the encryption and decryption process simulation using cat image with the same key that is  $x_0 = 0.1, \lambda = 4$ . Thus seen that the decryption process succeeded in opening the original data (Figure 8a).

In Figure 8d are shown that the attempt to decrypt using a key difference between the value  $x_0$  by  $10^{-16}$  did not succeed to get the original image. This is due to one of the properties of the logistic map is sensitive to initial values. Value of 0.1 and 0.1000000000000001 still considered different values by this algorithm. But in Figure 8e, when the difference reaches  $10^{-17}$  the decryption process got the information of original image. It shows that the numbers 0.1 and 0.1000000000000001 is considered to be the same number that is 0.1. Previously, have been tested using the different decryption key for grayscale and color images ranging from  $10^{-2}$  to  $10^{-16}$ . So we get the sensitivity of this algorithm is up to  $10^{-16}$ .

So we obtain that a brute force attack would be very difficult to get the original image information, because these algorithms are very sensitive to changes in the value of the key. Histogram display for each column in a row is just the components Red (R) that shows the distribution of pixel values (Figure 9).

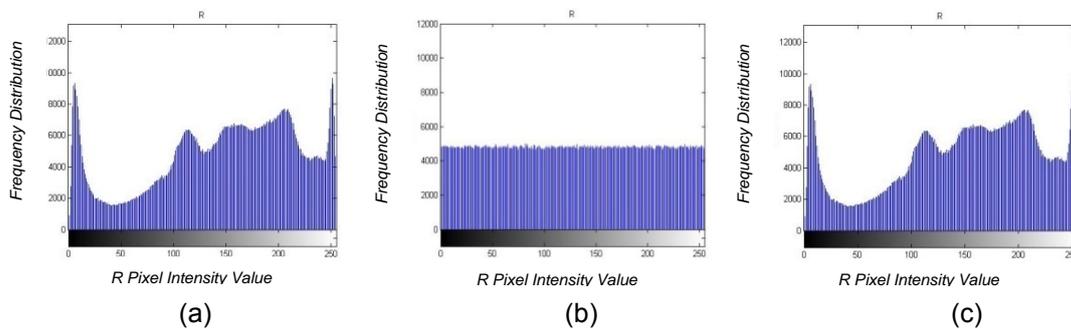


Figure 9. Histogram of Cat.jpg. (a) Histogram of Figure 8a; (b) Histogram of Figure 8b; (c) Histogram of Figure 8c

### 3.3 Size of Key Space

The random number generator which was used to generate key stream is logistic map. Keys that are used on logistic map are  $x_0$  and  $\lambda$ , where  $x_0$  and  $\lambda$  are real number. If we use a higher level of precision, for example 64-bit double precision IEEE standard, the precision level will reach  $10^{-15}$ . So, the total of key space are  $10^{15} \times 10^{15} = 10^{30}$ . Time required to exhaustive key search [14] can be seen in Table 2.

Table 2. Time Required to Exhaustive Key Search

Key Space	Experiments/sec	Time Needed		
		Second	Days	Years
$10^{30}$	$10^6$	$10^{24}$	$1,157 \times 10^{19}$	$3,215 \times 10^{16}$
	$10^{12}$	$10^{18}$	$1,157 \times 10^{13}$	$3,215 \times 10^{10}$
	$10^{15}$	$10^{15}$	$1,157 \times 10^{10}$	$3,215 \times 10^7$
	$10^{18}$	$10^{12}$	$1,157 \times 10^7$	32150
	$10^{21}$	$10^9$	11574	32,15

It can be concluded that, the algorithm is very difficult to be cracked by brute force attack.

### 3.4 Randomness Key Stream Analysis

Test of randomness performed using international standard testing of the National Institute of Standards and Technology is monobits frequency test [15]. With the initial value  $x_0 = 0.1$  dan  $\lambda = 4$  testing has been carried out on the key streams generated by the chaos-

based encryption algorithm. Key stream test in the key stream generated by the logistic map are:  $x_{203}, x_{206}, x_{209}, \dots, x_{290}, x_{293}, x_{296}, x_{299}$ , so the length of the binary sequence is 1320 bits.

The testing procedure is [14]:

1.  $n = 1320$
2. With the help of computer calculated  $X_1$  until  $X_{1320}$  and obtained  $S_n = -12$ .
3. Then compute  $S_{obs} = \frac{|S_n|}{\sqrt{n}} = \frac{|-12|}{\sqrt{1320}} = 0.3302891295$
4. After that, get the  $P\text{-value} = \text{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right) = \text{erfc}\left(\frac{0.3302891295}{\sqrt{1320}}\right) = 0.7411815059$
5. It can be concluded with the significance level of 1 % proven true that the sequence is random because  $P\text{-value} > 0.01$ .

Obtained from the randomness analysis of the key stream which generated by this algorithm is completely random. So that, the algorithm is very difficult to be cracked by known plaintext attack that utilizes the statistical properties of the ciphertext.

### 3.5 Histogram Analysis

The keys that we used is  $x_0 = 0.1$  and  $\lambda = 4$ , performed testing using Goodness of fit test [16] on digital image of the encryption process results with various sizes. The results of test statistic values towards grayscale test data digital image cat.jpg with Goodness of fit method are shown in Table 3.

Table 3. Test Statistic Values for Grayscale Image

Test Data	Pixel Size	Test Statistic Value
Data 1.	80 x 60	287.5733333333
Data 2.	320 x 240	255.6800000000
Data 3.	640 x 480	292.2483333333
Data 4.	1280 x 960	265.2695833333
Data 5.	2560 x 1920	260.4456250000

While the test results for the test data of cat.jpg color digital image in various sizes are shown in Table 4.

Table 4. Test Statistic Value for Color Image.

Test Data	Pixel Size	Test Statistic Value for Red (R)	Test Statistic Value for Green (G)	Test Statistic Value for Blue (B)
Data 1.	80 x 60	222.2933333333	233.0666666667	264.2133333333
Data 2.	320 x 240	241.4066666667	271.7600000000	283.6266666667
Data 3.	640 x 480	263.7266666667	236.3666666667	201.4466666667
Data 4.	1280 x 960	226.9208333333	296.3325000000	291.4933333333
Data 5.	2560 x 1920	231.9140625000	225.1015625000	231.3796875000

With degrees of freedom  $256-1=255$ , and 1% significance level, the critical value is 310.4573882199. It was seen from the results of the experiment are shown in Table 3 and Table 4, all the test statistic values less than the critical value. It can be concluded that all the tested data proved uniformly distributed. As seen in Figure 9b for component R, histogram diagram from the results of encrypted image is flat, which shows the distribution of encrypted image pixel value, is uniform.

Based on the test results, the distribution encrypted image pixel value, using this algorithm, is uniform. So this ciphertext is very difficult to be cracked by known plaintext attack that utilizes the statistical properties of the ciphertext.

## 4. Conclusion

Conclusion of this paper are :

- a. Performance of chaos-based encryption algorithm are:
  - (i). The time of encryption and decryption processes are relatively similar to each grayscale and color image.
  - (ii). Time of color image encryption and decryption process is longer than grayscale image

- because on the color image, the process of encryption and decryption were done for each component grayscale, they are red, green, and blue.
- (iii). Encryption algorithm has key space for  $10^{30}$  and key sensitivity that reaches  $10^{-16}$ , so the algorithm is very difficult to be cracked by brute force attack.
  - (iv). This encryption algorithm is very difficult to be cracked by known plaintext attack, due to the value distribution of the pixels of the encrypted result is proved uniform (all test statistic value less than the critical value) and key streams that were generated, proved to be completely random with  $P_{\text{value}} = 0.74118 > 0.01$ .
- b. So, it can be concluded that, the algorithm is very difficult to be cracked by brute force attack and also known plaintext attack.

### Acknowledgments

This work was supported by the Directorate of Research and Community Engagement Universitas Indonesia (Initial Research Grant PUPT UI, No. 3355/H2.R12/HKP.05.00/2014).

### References

- [1] Stallings W. *Computer and Network Security: Principle and Practice* (5<sup>th</sup> ed.). New York: Prentice hall. 2011.
- [2] Pareek NK, Patidar V, Sud KK. Image encryption using chaotic logistic map. *Journal of Image and Vision Computing*. 2006; 24: 926-934.
- [3] Patidar V, Pareek NK, Sud KK. A new substitution-diffusion based image cipher using chaotic standard and logistic maps. *Journal of Commun Nonlinear Sci Numer Simulat*. 2009; 14: 3056-3075.
- [4] Devaney RL. *An introduction to chaotic dynamical systems* (2<sup>nd</sup> ed.). New York: Addison-Wesley Publishing company, Inc. 1989.
- [5] Hirsch MW, Smale S, Devaney RL. *Differential equations, dynamical systems, and an introduction to chaos* (2<sup>nd</sup> ed.). Elsevier Academic Press. 2004.
- [6] Zhang W, Wong K, Yu H, Zhu Z. An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. *Journal of Commun Nonlinear Sci Numer Simulat*. 2013; 18: 2066-2080.
- [7] Suryadi MT. *New Chaotic Algorithm for Video Encryption*. 4<sup>th</sup> The International Symposium on Chaos Revolution in Science, Technology and Society 2013, Jakarta, August 28-29. 2013.
- [8] Zhang Y. Plaintext Related Image Encryption Scheme Using Chaotic Map. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(1): 635-643.
- [9] Zhang Y, Xia JL, Cai P, Chen B. Plaintext related two-level secret key image encryption scheme. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1254-1262.
- [10] Gao H, Zhang Y, Liang S, Li D. A new chaotic algorithm for image encryption. *Journal of Chaos, Solutons and Fractals*. 2006; 29: 393-399.
- [11] Abu Zaid, Osama M, El-Fishawy, Nawal A, Nigm EM. Cryptosystem Algorithm Based on Chaotic System for Encrypting Colored Image. *International Journal of Computer Science Issues*. 2013; 10(4): 215-224.
- [12] Eva N, Suryadi MT. *Chaos-Based Encryption Algorithm for Digital Image*. The 2<sup>nd</sup> IndoMS International Conference on Mathematics and Its Applications, Yogyakarta, November 6-7. 2013.
- [13] Kocarev L, Lian S. *Chaos-based cryptography*. Berlin Heidelberg: Springer-Verlag. 2011.
- [14] Stallings W. *Data and Computer Communications* (8<sup>th</sup> ed.). Prentice Hall. New Jersey. 2007.
- [15] National Institute of Standard and Technology (NIST). *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (Special Publication 800-222). U.S. Department of Commerce. 2010.
- [16] Walpole RE, Myers, RH, Myers SL, Ye, K. *Probability and Statistics for Engineers and Scientists* (9<sup>th</sup> ed.). Prentice Hall, Boston. 2012.