**Original Article**

e-ISSN: 2774-2016 - https://journal.itera.ac.id/index.php/indojam/
p-ISSN: 2774-2067

# KRIPTOGRAFI DAN KRIPTANALISIS CITRA DIGITAL MENGGUNAKAN ALGORITMA *LOGISTIC MAP*

Received 14th September 2023
Accepted 10th November 2023
Published 13th November 2023

Open Access

**DOI:**
10.35472/indojam.v3i2.1587

Teguh Pasmahendri*[a], Syamsyida Rozi[a], Cut Multahadah[a]

[a] *Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Jambi*

* *Corresponding E-mail: teguhahm@gmail.com*

**Abstract**: Data security is very necessary for companies, institutions, organizations and individuals who have confidential information. The use of data security is intended so that information cannot be stolen by others. The rise of personal data leakage cases in Indonesia has made people worried about the security of their personal data such as identity cards (KTP), emails, and so on. One way to secure data in the form of a digital image is to encrypt it. One of the encryption algorithms is Logistic Map. Logistic map is one of the chaos algorithms that is often used in image cryptography because this algorithm is able to generate a complex array of random numbers with a simple recursive polynomial equation. In this research, the author encrypts a digital ID card image with a size of 3×3 pixels with an RGB image. Based on this explanation, it can be concluded that the encryption and decryption process using the Logistic Map Algorithm using parameter values in the interval range [3.57;4] and initial values in the interval range [0;1] can successfully encrypt a digital image. In this research, changes were made to the initial value (x_0) and different parameter values (r). The results obtained based on experiments conducted by the author using a digital image measuring 1572 × 966 pixels are with the value of x_0 = 0.8672 and r = 3.7541 experiencing a very varied color intensity distribution compared to the histogram produced with the initial value and other parameter values in the research conducted.

**Keywords:** *Citra Digital, Data Security, Logistic Map*

**Abstrak:** Keamanan data sangat diperlukan bagi perusahaan, institusi, organisasi maupun perseorangan yang memiliki informasi rahasia. Penggunaan keamanan data ditujukan agar informasi tidak dapat dicuri oleh orang lain. Maraknya kasus kebocoran data pribadi di Indonesia membuat khawatir dengan keamanan data pribadinya seperti Kartu Tanda Penduduk (KTP), email, dan lain sebagainya. Adapun salah satu cara untuk mengamankan suatu data berbentuk citra digital adalah dengan mengenkripsikannya. Salah satu algoritma enkripsinya adalah *Logistic Map*. *Logistic map* adalah salah satu algoritma *chaos* yang sering digunakan dalam kriptografi citra karena algoritma ini mampu menghasilkan deretan bilangan acak yang kompleks dengan persamaan polinomial rekursif yang sederhana. Dalam penelitian kali ini penulis mengenkripsikan citra digital KTP dengan ukuran $\mathbf{3 \times 3}$ *pixel* dengan citra RGB. Tujuan dari menggunakan ukuran $\mathbf{3 \times 3}$ *pixe* adalah untuk mempermudah serta menyederhankan perhitungan penulis dalam menghitung secara manual apabaila menggunakan ukuran asli maka penulis akan kesulitan dalam proses pengerjaannya. Berdasarkan penjelasan tersebut dapat diambil kesimpulan yaitu proses *enkripsi* dan *dekripsi* menggunakan Algoritma *Logistic Map* dengan menggunakan nilai parameter direntang interval $[\mathbf{3,57;4}]$ dan nilai awal direntang interval $[\mathbf{0;1}]$ berhasil mengenkripsikan suatu citra digital. Pada penelitian kali ini di lakukan perubahan terhadap nilai awal $(\boldsymbol{x_0})$ dan nilai parameter $(\boldsymbol{r})$ yang berbeda-beda. Hasil yang didapatkan berdasarkan percobaan yang telah dilakukan penulis dengan menggunakan citra digital berukuran $\mathbf{1572 \times 966}$ *pixel* adalah dengan nilai $\boldsymbol{x_0 = 0,8672}$ dan $\boldsymbol{r = 3,7541}$ mengalami penyebaran intensitas warna yang sangat bervariasi dibandingkan dengan histogram yang dihasilkan dengan nilai awal dan nilai parameter lainnya dalam penelitian yang dilakukan.

**Original Article**

**Kata Kunci:** *Citra Digital, Keamanan Data, Logistic Map*

## Pendahuluan

Data pribadi menyimpan berbagai privasi seseorang, dan menjadi tabu jika disebarkan tanpa seizin pemiliknya. Maraknya kasus kebocoran data pribadi di Indonesia membuat khawatir dengan keamanan data pribadinya seperti Kartu Tanda Penduduk (KTP), email, dan lain sebagainya. Banyaknya celah pada situs-situs perusahaan atau instansi pemerintah memudahkan seorang peretas atau *hacker* dengan tujuan jahat untuk membobol data pribadi masyarakat (Ibrahim dkk, 2017). Sementara itu, permasalahan sosial dan politik juga menjadi motivasi *hacker* kondang bernama Bjorka yang menyebarkan data pribadi beberapa pejabat dan tokoh publik termasuk Menteri Komunikasi dan Informatika (Menkominfo). Aksi Bjorka ini dinilai sebagai aksi *hacktivism* yang mengaspirasikan lemahnya keamanan data digital dan kurangnya upaya pemerintah dalam melindungi hal tersebut (Handoyo dkk, 2018). Oleh sebab itu pemerintah perlu meningkatkan kembali keamanan data agar tidak dengan mudah dicuri oleh pihak yang tidak bertanggung jawab.

Keamanan data sangat diperlukan bagi perusahaan, institusi, organisasi maupun perseorangan yang memiliki informasi rahasia. Penggunaan keamanan tersebut, ditujukan agar informasi tidak dapat dicuri oleh orang lain. Ada beberapa metode yang digunakan untuk mengamanankan pesan dari zaman dahulu, seperti menyembunyikan pesan ke dalam media lain agar orang lain terkecoh dengan tampilannya, ilmu ini disebut *steganography*. Selain pesan yang disembunyikan ke dalam media ada juga ilmu pengamanan dengan menyandikan atau mengubah makna pesan tidak terbaca dengan menggunakan berbagai perhitungan, ilmu itu disebut *cryptography* (Ibrahim dkk, 2017). Citra digital merupakan salah satu bentuk multimedia yang penting. Citra digital menyajikan informasi secara visual dan informasi yang disajikan oleh sebuah citra digital lebih kaya daripada yang disajikan secara tekstual (Chang dkk, 2001).

Dikarenakan meningkatnya permintaan akan keamanan informasi, *enkripsi* dan *dekripsi* citra digital telah menjadi penelitian yang penting dan memiliki prospek aplikasi yang luas. Maka dari itu bidang *enkripsi* menjadi sangat penting di era sekarang. Keamanan citra digital sering sekali memiliki ancaman yang cukup serius dan perlu diperhatikan. *Enkripsi* dan *dekripsi* pada citra digital telah diterapkan pada bidang pemerintahan, komunikasi kemiliteran dan lain sebagainya. Agar data aman dari berbagai serangan dan untuk integritas data penulis harus *mengenkripsi* data tersebut sebelum dikirim atau disimpan (Abdulgader dkk, 2015). Salah satu cara untuk mencegah terjadi kebocoran rahasia yang berupa citra digital dapat dicegah dengan menggunakan kriptografi algoritma *enkripsi* citra digital (Munir,2019). Peneliti pada kasus ini mengambil fungsi chaos yaitu *Logistic Map*. Karena algoritma ini mampu menghasilkan deretan bilangan acak yang kompleks dengan persamaan polynomial rekursif yang sederhana dan algoritma *enkripsi* citra digital berbasis *chaos* mengkombinasikan teknik permutasi dan substitusi. Dua buah fungsi *chaos* dapat dikerjakan dengan *Arnold Cat Map* dan *Logistic Map.* Fungsi *Logistic Map* mempunyai keunggulan dalam kecepatan mengenkripsikan data kemudian mempunyai fungsi sinus berosilasi tinggi akan dikombinasikan dengan fungsi *Logistic Map* sehingga diharapkan tingkat keacakannya lebih tinggi. (Younes, 2008). Peneliti pada kasus ini mengambil fungsi chaos yaitu *Logistic Map*.

*Logistic Map* adalah sistem *chaos* yang paling sederhana yang berbentuk persamaan iteratif. Dimana nilai awal persamaan iterasi adalah $\boldsymbol{x_0}$ yang dimana persamaan satu bersifat deterministik sebab jika dimasukkan nilai $\boldsymbol{x_0}$ yang sama maka dihasilkan barisan nilai *chaotik* ($\boldsymbol{x_i}$) yang sama pula. Oleh karena itu, pembangkit bilangan acak dengan sistem *chaos* disebut *pseudo-random generator*. Sifat algoritma *chaos* yang paling penting adalah sensitivitasnya pada perubahan kecil nilai awal (Stallings dan William, 2004).

Sistem *chaos* terdapat dimana-mana, dari pertimbangan alam yang paling intim untuk seni apapun. Saat ini, masih belum terdapat definisi yang pasti mengenai *chaos* itu sendiri. Meskipun begitu, kebanyakan orang akan setuju mengenai definisi berikut ini. *Chaos* adalah perilaku waktu asimtotik dan

**Original Article**

aperiodik dalam sistem deterministik yang menunjukkan ketergantungan sensitif terhadap kondisi awal (Zeynep, 2009).

Softwere yang digunakan pada penelitian ini adalah *Python*. *Python* adalah bahasa pemograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar (Syahrudin dan Kurniawan, 2018).

## Metode

Metode yang digunakan dalam penelitian ini yaitu menggunakan Algorima *Logistic Map*. *Logistic Map* merupakan salah satu algoritma fungsi *chaos* yang dimana hasilnya bergantung terhadap nilai parameter $(r)$ dan niai awal $(x_{0)}$ . Syarat dari algoritma logistic map ini yaitu nilai $r$ berada di interval [3,57;4] dan nilai $x_0$ berada di interval [0;1] (Lynch,2014). Pada penelitian kali ini penulis menggunakan nilai $r$ dan $x_0$ yang berbeda-beda, karena tujuan pada kali ini adalah untuk melihat di nilai parameter dan nilai awal berapa proses enkripsi menghasilkan citra yang chaos secara signifikan.

Adapun Langkah-langkahnya adalah sebagai berikut:

1. Identifikasi Masalah

   Pada penelitian ini dilakukan penyandian Kartu Tanda Penduduk (KTP) untuk mengamankan data pribadi yang disalahgunakan oleh pihak yang tidak bertanggung jawab.
2. Pengumpulan Data

   Tahapan ini dilakukan dengan menggunakan data berupa citra digital KTP penulis.
3. Pembentukan Matriks dari Citra

   Pada tahap ini dilakukan pembentukan matriks awal dengan menggunakan *pyton* berdasarkan dengan citra digital ukuran $3 \times 3$ *pixel.*
4. Difusi *Logistic Map*

   Tahap ini proses difusi menggunakan *logistic map* dengan mengambil nilai parameter, dan nilai awal yang sudah ditentukan kemudian dilakukan proses perhitungan dengan persamaan 4.
5. Pembangkitan Kunci *(Keystream)*

   Pada tahap ini dibangkitkan kunci rahasia yaitu bilangan bulat dari $(x_i \times 1000)$ mod 256. Dengan $x_i$ adalah nilai yang didapatkan pada proses difusi *logistic map.*
6. Enkripsi Operasi *Exclusive-OR* dengan Pembangkit Kunci

   Pada tahapan ini setelah nilai *keystream* dibangkitkan maka diperoleh matriks baru untuk citra digital melalui operasi XOR.
7. Hasil Enkripsi

   Pada tahap ini didapatkan hasil berupa output perubahan citra digital yang pertamanya berupa *plainmage* menjadi *ciphermage*.
8. Dekripsi Operasi *Exclusive-OR* dengan Pembangkit Kunci

   Pada tahap ini dilakukan pendekripsian dari *ciphermage* menjadi *plainmage* yaitu dengan cara dilakukan operasi XOR dengan nilai pembangkit kunci awal yang digunakan pada saat proses *enkripsi*.
9. Hasil Dekripsi

   Pada tahap ini didaptkan hasil berupa output perubahan citra digital yang pertamanya berupa *ciphermage* menjadi *plainmage*.
10. Uji Analisis Keamanan

    Pada tahap ini dilakukan uji analisis keamanan yaitu dengan menggunakan analisis sensitivitas kunci dan analisis histogram.
11. Intrepretasi Hasil Uji Analisis Keamanan

    Pada tahap ini dilakukan interpretasi yang didapatkan dari penguncian analisis yang dilakukan apakah *enkripsi* citra digital yang menggunakan algoritma *logistic map* layak atau tidak untuk dijadikan topik skripsi.
12. Kesimpulan

    Kesimpulan dapat ditarik setelah proses interpretasi hasil analisis sensitivitas kunci dan histogram didapatkan.

## Hasil dan Diskusi

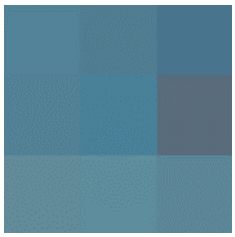
Pada proses *enkripsi* dan *dekripsi* warna direpresentasikan dengan bilangan



*Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)*
*Institut Teknologi Sumatera, Lampung Selatan, Indonesia*

**Original Article**

[0; 255],kemudian proses *enkripsi* akan dilakukan dengan operasi XOR. Supaya proses *enkripsi* sukses, karakter warna dikonversi menjadi bilangan biner. Oleh karena itu perlu dipastikan himpunan bilangan biner, yaitu $G = \{0,1\}$ dengan operasi XOR merupakan suatu grup yang dimana memiliki beberapa aksioma yang harus dipenuhi oleh $G = \{0,1\}$ yaitu: $G$ bukan suatu himpunan kosong,Operasi $\oplus$ bersifat tertutup didalam $G$ ,Operasi $\oplus$ bersifat asosiatif didalam $G$ , $G = \{0,1\}$ memiliki elemen identitas dan Setiap anggota $G$ mempunyai invers.



**Gambar 1.** Citra KTP ukuran $3 \times 3$

Proses *enkripsi* menggunakan *logistic map* dan menerapkan operasi XOR adalah sebagai berikut:

1. Secara sederhana gambaran tentang proses *enkripsi* menggunakan *logistic map* akan ditunjukkan menggunakan citra RGB (Red,Green and Blue) pada **Gambar 6** dengan ukuran $3 \times 3$ *pixel*. Dengan menggunakan *python*, akan ditemukan entri matriks dibalik citra digital **Gambar 6**, yaitu dengan *pseudocode* berikut:

```
import numpy
from PIL import Image
import timeit
start=timeit.default_timer()
p1=Image.open(r'C:\Users\ACER\Documents\SKRIPSI TEGUH\ukuran 3 jpg.jpg')
mp1=numpy.asarray(p1)
row,col,dim=mp1.shape
```

Dan diperoleh matriks awal dibalik citra **Gambar 1** adalah:

$$A = \begin{bmatrix} [83 & 131 & 153] & [82 & 130 & 152] & [94 & 140 & 155] \\ [80 & 128 & 150] & [79 & 127 & 149] & [96 & 142 & 157] \\ [82 & 113 & 134] & [81 & 112 & 133] & [93 & 134 & 152] \end{bmatrix}$$

2. Karena citra digital berukuran $3 \times 3$ *pixel*, maka didefinisikan vektor $x \in \mathbb{R}^9$ dengan entri 0 dengan $n = 9$.
3. Proses difusi menggunakan *logistic map* dilakukan dengan mengambil nilai parameter dan nilai awal yang dimana pada proses kali ini penulis mengambil nilai parameter $r = 3{,}8$ dan nilai awal $x_0 = 0{,}5$ sehingga dalam hal ini *logistic map* ditulis menjadi $x_{n+1} = 3{,}8x_n(1 - x_n)$. Selanjutnya, dengan menerapkan *logistic map* diperoleh vektor $x \in \mathbb{R}^9$ yang baru pada **Tabel 1.**

**Tabel 1.** Proses Perhitungan Vektor x

| N | $x_{n+1} = rx_n(1 - x_n)$, |
|---|---|
| 1 | $x_1 = 3{,}8 \times 0{,}5 \times (1 - 0{,}5) = 0{,}95$ |
| 2 | $x_2 = 3{,}8 \times 0{,}95 \times (1 - 0{,}95) = 0{,}1805$ |
| 3 | $x_3 = 3{,}8 \times 0{,}1805 \times (1 - 0{,}1805) = 0{,}56209505$ |
| 4 | $x_4 = 3{,}8 \times 0{,}56209505 \times (1 - 0{,}56209505) = 0{,}93534791$ |
| 5 | $x_5 = 3{,}8 \times 0{,}93534791 \times (1 - 0{,}93534791) = 0{,}22979434$ |
| 6 | $x_6 = 3{,}8 \times 0{,}22979434 \times (1 - 0{,}22979434) = 0{,}67255782$ |
| 7 | $x_7 = 3{,}8 \times 0{,}67255782 \times (1 - 0{,}67255782) = 0{,}83685043$ |
| 8 | $x_8 = 3{,}8 \times 0{,}83685043 \times (1 - 0{,}83685043) = 0{,}51882079$ |
| 9 | $x_9 = 3{,}8 \times 0{,}51882079 \times (1 - 0{,}51882079) = 0{,}94865417$ |

Sehingga diperoleh

$$x = \begin{bmatrix} 0{,}95 \\ 0{,}1805 \\ 0{,}56209505 \\ 0{,}93534791 \\ 0{,}22979434 \\ 0{,}67255782 \\ 0{,}83685043 \\ 0{,}51882079 \\ 0{,}94865417 \end{bmatrix}$$

4. Kemudian dibangkitkan kunci rahasia (*keystream*) yaitu bilangan bulat melalui formula

$$((x_i \times 1000)\ mod\ 256)$$

Berdasarkan vektor $x$ pada **Tabel 3**, maka *keystream* (K) yang digunakan pada saat ini adalah:

$$K = \begin{bmatrix} K_{0,0} \\ K_{0,1} \\ K_{0,2} \\ K_{1,0} \\ K_{1,1} \\ K_{1,2} \\ K_{2,0} \\ K_{2,1} \\ K_{2,2} \end{bmatrix} = \begin{bmatrix} \lfloor 0{,}95 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}1805 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}56209505 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}93534791 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}22979434 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}67255782 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}83685043 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}51882079 \times 1000 \rfloor\ mod\ 256 \\ \lfloor 0{,}94865417 \times 1000 \rfloor\ mod\ 256 \end{bmatrix} = \begin{bmatrix} 182 \\ 180 \\ 50 \\ 167 \\ 229 \\ 160 \\ 68 \\ 6 \\ 180 \end{bmatrix}$$

Pada proses perhitungan di atas menggunakan fungsi lantai yang dimana tujuannya untuk

**Original Article**

pembulatan kebawah yang apabila nilai yang dihasilkan tidak integer, karena syarat dari nilai keystream adalah harus bilangan integer.

5. Setelah *keystream* tersebut dibangkitkan maka diperoleh entri matriks baru untuk citra melalui formula berikut:

$$b_{i,j} = a_{i,j} \oplus K_{i,j}$$

Dengan $b_{i,j}$ merupakan elemen pada baris ke $i$ kolom ke $j$ pada matriks baru dan $a_{i,j}$ merupakan elemen pada baris ke $i$ kolom ke $j$ pada matriks awal dimana $i, j = 0, 1, 2$. Matriks baru akan berbentuk:

$$\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \\ b_{2,0} & b_{2,1} & b_{2,2} \end{bmatrix}$$

Maka perhitungannya sebagai berikut:

a) $b_{0,0} = a_{0,0} \oplus K_{0,0}$

$= [83 \quad 131 \quad 153] \oplus 182$

b) $b_{0,1} = a_{0,1} \oplus K_{0,1}$

$= [82 \quad 130 \quad 152] \oplus 180$

$= [230 \quad 54 \quad 44]$

c) $b_{0,2} = a_{0,2} \oplus K_{0,2}$

$= [94 \quad 140 \quad 155] \oplus 50$

$= [108 \quad 190 \quad 169]$

d) $b_{1,0} = a_{1,0} \oplus K_{1,0}$

$= [80 \quad 128 \quad 150] \oplus 167$

$= [247 \quad 39 \quad 49]$

e) $b_{1,1} = a_{1,1} \oplus K_{1,1}$

$= [79 \quad 127 \quad 149] \oplus 229$

$= [170 \quad 154 \quad 112]$

f) $b_{1,2} = a_{1,2} \oplus K_{1,2}$

$= [96 \quad 142 \quad 157] \oplus 160$

$= [192 \quad 46 \quad 61]$

g) $b_{2,0} = a_{2,0} \oplus K_{2,0}$

$= [82 \quad 113 \quad 134] \oplus 68$

$= [22 \quad 53 \quad 194]$

h) $b_{2,1} = a_{2,1} \oplus K_{2,1}$

$= [81 \quad 112 \quad 133] \oplus 6$

$= [87 \quad 118 \quad 131]$

i) $b_{2,2} = a_{2,2} \oplus K_{2,2}$

$= [93 \quad 134 \quad 152] \oplus 180$

$= [233 \quad 50 \quad 44]$

Sehingga diperoleh matriks baru:

$$B = \begin{bmatrix} [229 \quad 53 \quad 47] & [230 \quad 54 \quad 44] & [108 \quad 190 \quad 169] \\ [247 \quad 39 \quad 49] & [170 \quad 154 \quad 112] & [192 \quad 46 \quad 61] \\ [22 \quad 53 \quad 194] & [87 \quad 118 \quad 131] & [233 \quad 50 \quad 44] \end{bmatrix}$$

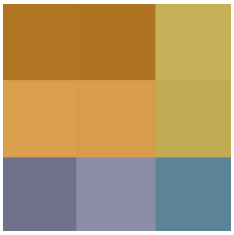
Adapun proses untuk melakukan *enkripsi* dengan menggunakan *softwere python* yaitu dengan *pseudocode* berikut:

```
import numpy
from PIL import Image
import timeit
start=timeit.default_timer()
p1=Image.open(r'C:\Users\ACER\Documents\SKRIPSI TEGUH\ukuran 3 jpg.jpg')
mp1=numpy.asarray(p1)
row,col,dim=mp1.shape
x=numpy.zeros(row*col)
x[0]=0.5
r=3.8
for i in range(row*col-1):
   x[i+1]=r*x[i]*(1-x[i])
mnewp1=numpy.zeros((row,col,dim),dtype=numpy.uint8)
key1=numpy.floor((x*1000)%256)
for i in range(row):
   for j in range(col):
      mnewp1[i,j]=mp1[i,j]^int(key1[i*col+j])
newp1=Image.fromarray(mnewp1)
stop=timeit.default_timer()
newp1.save(r'C:\Users\ACER\Documents\SKRIPSI TEGUH\hasil ukuran 3 jpg.jpg')
print(stop-start)
```

Sehingga didapatkan hasil enkripsi dari citra KTP dengan ukuran $3 \times 3$ *pixel* dapat dilihat pada **Gambar 2.**


*Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)*
*Institut Teknologi Sumatera, Lampung Selatan, Indonesia*

**Original Article**



**Gambar 2.** Ciphermage Hasil Enkripsi

Kemudian untuk mengembalikan hasil enkripisi atau chipermage dapat dilakukan proses sebagai berikut :

1. Dengan menggunakan *python,* akan ditemukan entri matriks dibalik citra digital **Gambar 7**, yaitu dengan *pseudocode* berikut:

```
import numpy
from PIL import Image
import timeit
start=timeit.default_timer()
p1=Image.open(r'C:\Users\ACER\Documents\SKRIPSI TEGUH\hasil ukuran 3 jpg.jpg')
mp1=numpy.asarray(p1)
row,col,dim=mp1.shape
```

Dan diperoleh matriks baru dibalik citra **Gambar 2** adalah:

$$B = \begin{bmatrix} [229 & 53 & 47] & [230 & 54 & 44] & [108 & 190 & 169] \\ [247 & 39 & 49] & [170 & 154 & 112] & [192 & 46 & 61] \\ [22 & 53 & 194] & [87 & 118 & 131] & [233 & 50 & 44] \end{bmatrix}$$

2. Selanjutnya pada proses ini akan dilakukan proses *dekripsi* untuk mengembalikan *chipermage* ke bentuk *plainmage*, yaitu dengan operasi XOR kan nilai matriks baru yang didapatkan pada proeses *enkripsi* sebelumnya dengan *keystream* yang digunakan pada proses *enkripsi,* adapun proses perhitungannya sebagai berikut :

a) $a_{0,0} = b_{0,0} \oplus K_{0,0}$

$= [229 \quad 53 \quad 47] \oplus 182$

$= [83 \quad 131 \quad 153]$

b) $a_{0,1} = b_{0,1} \oplus K_{0,1}$

$= [230 \quad 54 \quad 44] \oplus 180$

$= [82 \quad 130 \quad 152]$

c) $a_{0,2} = b_{0,2} \oplus K_{0,2}$

$= [108 \quad 190 \quad 169] \oplus 50$

$= [94 \quad 140 \quad 155]$

d) $a_{1,0} = b_{1,0} \oplus K_{1,0}$

$= [247 \quad 39 \quad 49] \oplus 167$

$= [80 \quad 128 \quad 150]$

e) $a_{1,1} = b_{1,1} \oplus K_{1,1}$

$= [170 \quad 154 \quad 112] \oplus 229$

$= [79 \quad 127 \quad 149]$

f) $a_{1,2} = b_{1,2} \oplus K_{1,2}$

$= [192 \quad 46 \quad 61] \oplus 160$

$= [96 \quad 142 \quad 157]$

g) $a_{2,0} = b_{2,0} \oplus K_{2,0}$

$= [22 \quad 53 \quad 194] \oplus 68$

$= [82 \quad 113 \quad 134]$

h) $a_{2,1} = b_{2,1} \oplus K_{2,1}$

$= [87 \quad 118 \quad 131] \oplus 6$

$= [81 \quad 112 \quad 133]$

i) $a_{2,2} = b_{2,2} \oplus K_{2,2}$

$= [233 \quad 50 \quad 44] \oplus 180$

$= [93 \quad 134 \quad 152]$

Sehingga didapatkan hasil dari perhitungan diatas membentuk matriks yang sama seperti matriks awal.

$$A = \begin{bmatrix} [83 & 131 & 153] & [82 & 130 & 152] & [94 & 140 & 155] \\ [80 & 128 & 150] & [79 & 127 & 149] & [96 & 142 & 157] \\ [82 & 113 & 134] & [81 & 112 & 133] & [93 & 134 & 152] \end{bmatrix}$$

Adapun proses untuk melakukan *dekripsi* dengan menggunakan *softwere python* yaitu dengan *pseudocode* berikut:

```
import numpy
from PIL import Image
import timeit
start=timeit.default_timer()
p1=Image.open(r'C:\Users\ACER\Documents\SKRIPSI TEGUH\hasil ukuran 3 jpg.jpg')
mp1=numpy.asarray(p1)
row,col,dim=mp1.shape
x=numpy.zeros(row*col)
x[0]=0.5
r=3.8
```

**Original Article**

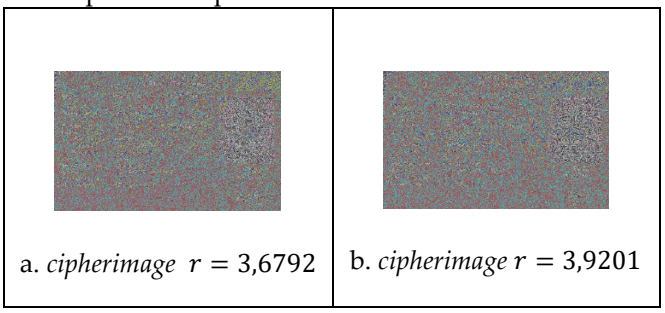
```
for i in range(row*col-1):
   x[i+1]=r*x[i]*(1-x[i])
mnewp1=numpy.zeros((row,col,dim),dtype=numpy.ui
nt8)
key1=numpy.floor((x*1000)%256)
for i in range(row):
   for j in range(col):
      mnewp1[i,j]=mp1[i,j]^int(key1[i*col+j])
newp1=Image.fromarray(mnewp1)
stop=timeit.default_timer()
newp1.save(r'C:\Users\ACER\Documents\SKRIPSI
TEGUH\Dekripsi hasil ukuran 3 jpg.jpg')
print(stop-start)
```

**Gambar 3.** *Plainmage* Hasil Proses Dekripsi

Pada penelitian ini penulis ingin melihat beberapa perubahan yang dihasilkan dengan menggunakan nilai awal dan nilai parameter yang berbeda-beda dan menggunakan ukuran $1572 \times 966$ *pixel*. Adapun tujuan dengan dilakukan perubahan nilai awal dan nilai parameter adalah untuk melihat tingkat sensitivitas yang dihasilkan pada setiap citra dan dengan menggunakan nilai mana histogram yang didapatkan nanti akan mengalami penyebaran warna yang lebih *chaos*.

Pada analisis ini akan dilakukan perubahan nilai awal ($x_0$) pada Algoritma *Logistic Map* dalam melakukan proses *dekripsi* untuk mengetahui sensitivitas *chaos* terhadap nilai awal. Pada algoritma *Logistic Map* suatu citra digital apabila dienkripsi akan mengalami *chaos* apabila nilai $r$ di dalam interval [3,57;4] dan nilai $x_0$ berada di interval $[0; 1]$. Pada penelitian ini akan dilakukan perubahan nilai awal ($x_0$) dan nilai parameter ($r$) dengan membandingkan citra enkripsi yang di hasilkan dan histogramnya. Adapun perubahan-perubahan yang dilakukan adalah sebagai berikut.

a. Analisis sensitivitas terhadap nilai $r$, diasumsikan $x_0 = 0{,}5347$ sedangkan untuk nilai $r = 3{,}6792$ dan $r = 3{,}9201$. Adapun hasil enkripsi yang dapatkan dapat dilihat pada **Gambar 4.**




**Gambar 4.** *Ciphermage* Dengan Nilai $r$ Berbeda-beda

Berdasarkan **Gambar 4,** Selanjutnya untuk melihat nilai awal dan nilai parameter mana yang sangat cocok digunakan dapat dilakukan proses memunculkan histogram. Histogram merupakan salah satu fitur citra yang penting, sebab sebuah histogram memperlihatkan distribusi intesitas *pixel-pixel* di dalam cerita tersebut. Dalam melakukan serangan dengan teknik analisis statistik, penyerang menggunakan histogram untuk menganalisis frekuensi kemunculan intensitas *pixel* untuk mendeduksi kunci atau *pixel-pixel* di dalam *plainmage*. Agar serangan dengan analisis statistik tidak dimungkinkan, maka di dalam enkripsi citra penting untuk menghasilkan histogram *ciphermage* yang tidak memiliki kemiripan secara statistik dengan histogram *plainmage*. Oleh karena itu *pixel-pixel* di dalam *ciphermage* seharusnya memiliki distribusi yang (relatif) *uniform* atau ditunjukkan dengan histogram yang terlebih datar (*flat*). Adapun proses untuk melakukan pemunculan histogram dengan menggunakan *softwere python* yaitu dengan *pseudocode* berikut:

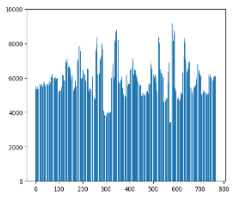
```
import numpy
from PIL import Image
p1=Image.open(r'C:\Users\ACER\Documents\SKRIP
SI TEGUH\(r=2)hasil ukuran 3 jpg.jpg')
mp1=numpy.asarray(p1)
p1_RGB=Image.fromarray(mp1)
histogram_p1_RGB=p1_RGB.histogram()
```



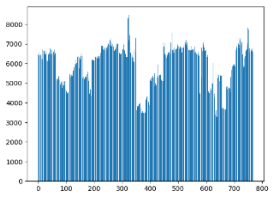
*Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)*
*Institut Teknologi Sumatera, Lampung Selatan, Indonesia*

**Original Article**

```
import matplotlib.pyplot as plt
plt.bar(range(256*3),histogram_p1_RGB,edgecolor='none')
```



a. histogram $r = 3{,}6792$

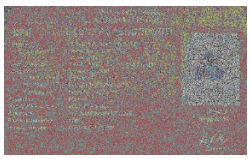


b. histogram $r = 3{,}9201$

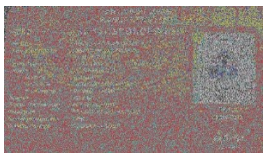
**Gambar 5.** Histogram Dengan Nilai $r$ Berbeda-beda

Berdasarkan **Gambar 5**, dengan mengansumsikan nilai $r$ yang berbeda didapatkan hasil bahwa nilai $r = 3{,}6792$ dan $x_0 = 0{,}5437$ mengalami penyebaran warna yang lebih siginifikan dibandingkan dengan nilai $r = 3{,}9201$ dan $x_0 = 0{,}5437$ . $r$ yang berbeda didapatkan hasil bahwa nilai $r = 3{,}6792$ dan $x_0 = 0{,}5437$ mengalami penyebaran warna yang lebih siginifikan dibandingkan dengan nilai $r = 3{,}9201$ dan $x_0 = 0{,}5437$.

b. Analisis sensitivitas terhadap nilai awal $(x_0)$, di asumsikan nilai $r = 3{,}7541$ sedangkan untuk nilai $x_0 = 0{,}4721$ dan $x_0 = 0{,}8672$. Adapun hasil *enkripsi* yang dapatkan dapat dilihat pada **Gambar 6.**



a. *cipherimage* $x_0 = 0{,}4721$

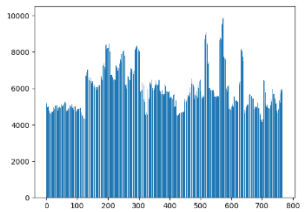


b. *cipherimage* $x_0 = 0{,}8672$

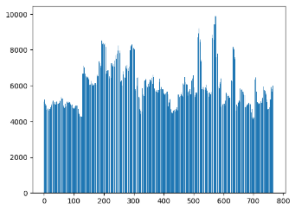
**Gambar 6.** *Ciphermage* Dengan Nilai $x_0$ Berbeda-beda

Berdasarkan **Gambar 6,** Selanjutnya untuk melihat nilai awal dan nilai parameter mana yang sangat cocok digunakan dapat dilakukan proses dengan memunculkan histogramnya.

Berdasarkan **Gambar 7**, dengan mengansumsikan nilai $x_0$ yang berbeda didapatkan hasil bahwa nilai $x_0 = 0{,}8672$ dan $r = 3{,}7541$ mengalami penyebaran warna yang lebih siginifikan dibandingkan dengan nilai $x_0 = 0{,}4721$ dan nilai $r = 3{,}7541$.



a. histogram $x_0 = 0{,}4721$



b. histogram $x_0 = 0{,}8672$

**Gambar 7.** Histogram Dengan Nilai $x_0$ Berbeda-beda

Jadi dapat disimpulkan bahwa nilai parameter dan nilai awal yang diasumsikan, citra digital sangat *chaos* ketika di nilai $r = 3{,}7541$ dan nilai $x_0 = 0{,}8672$. Pada nilai $r$ dan $x_0$ tersebut mengalami perubahan yang sangat signifikan yaitu penyebaran warna yang terjadi lebih banyak dibandingkan dengan gambar yang lainnya. Pada histogram, titik sumbu $x$ menyatakan warna yaitu $0 - 255$. Oleh karena pada data yang penulis gunakan yaitu citra RGB, maka banyaknya titik sumbu $x$ pada histogram yang ditampilkan adalah $3 \times 256 = 768$ . Sedangkan sumbu $y$ menyatakan intensitas warna dalam citra tersebut muncul.

## Kesimpulan

Proses *enkripsi* diawali dengan menentukan nilai *keystream* yang dibangkitkan dari suatu persamaan *Logistic Map* sehingga mendapatkan nilai *keystream* yang acak. Selanjutnya, dengan melakukan operasi XOR antara anggota pada matriks awal dengan nilai *keystream*, dapat diporeleh matriks baru yang merupakan representasi dari *chipermage*. Kemudian, untuk mengembalikan *chipermage* menjadi *plainmage* dapat dilakukan proses dekripsi yaitu dengan melakukan operasi XOR antara matriks hasil *chipermage* yang didapatkan dengan nilai *keystream*. Berdasarkan hasil dan pembahasan diatas, dapat diambil kesimpulan yaitu proses *enkripsi* dan *dekripsi* menggunakan Algoritma *Logistic Map* dengan simulasi menggunakan nilai parameter direntang interval $[3{,}57; 4]$ dan nilai awal direntang interval $[0; 1]$ berhasil mengenkripsikan suatu citra digital yang citra awalnya jelas menjadi buram (*chaos*).

Kemudian untuk melihat nilai sensitivitas kunci dan histogramnya ukuran pixel di perbesar menjadi $1572 \times 966$ *pixel* agar hasil enkripsi yang didapatkan lebih jelas. Pada penelitian kali ini penulis melakukan perubahan terhadap nilai awal $(x_0)$ dan nilai parameter $(r)$ yang berbeda-beda. Hasil yang didapatkan berdasarkan percobaan yang telah dilakukan penulis dengan menggunakan citra digital berukuran $1572 \times 966$ *pixel* adalah dengan nilai $x_0 = 0{,}8672$ dan $r = 3{,}7541$ mengalami penyebaran intensitas warna yang sangat bervariasi dibandingkan dengan histogram yang dihasilkan dengan nilai awal dan nilai parameter lainnya dalam penelitian yang dilakukan. Sehingga proses *enkripsi* sudah bisa dikatakan sangat baik memberikan *chipermage* yang sulit dipecahkan.

## Konflik Kepentingan

Penulis menyatakan bahwa artikel ini tidak memiliki konflik kepentingan tentang publiksai.

## Ucapan Terima Kasih

Ucapan terimakasih penlis sampaikan kepada pihak instansi Diskominfo Jambi yang telah memperbolehkan mengolah data untuk penelitian kali ini.

## Referensi

[1] A. Abdulgader, M. Ismail, N. Zainal, and T. Idbeaa, "Enhancement of AES algorithm based on chaotic maps and shift operation for image encryption," *J Theor Appl Inf Technol*, vol. 71, pp. 1–12, Nov. 2015.

[2] C.-C. Chang, M.-S. Hwang, and T.-S. Chen, "A new encryption algorithm for image cryptosystems," *Journal of Systems and Software*, vol. 58, no. 2, pp. 83–91, 2001.

[3] D. R. I. M. Setiadi, A. Handoyo, E. Rachmawanto, A. Sari, and A. Susanto, "Teknik Penyembunyian dan Enkripsi Pesan pada Citra Digital dengan Kombinasi Metode LSB dan RSA," *Jurnal Teknologi dan Sistem Komputer*, vol. 6, p. 37, Nov. 2018, doi: 10.14710/jtsiskom.6.1.2018.37-43.

[4] R. N. Ibrahim and I. M.S., "Perancangan Aplikasi Stegakrip dengan Metode LSB dan Algoritma RSA Berbasis Web", *JCB*, vol. 11, no. 2, pp. 98–109, Dec. 2017.

[5] S. Lynch, "Dynamical Systems with Applications using MATLAB® 2nd Edition," 2014.

[6] R. Munir, "Kriptografi," *Informatika*, Bandung, 2006.

[7] S. William, "Cryptography and Network Security. Principles and Practice," *Prentice-Hall*, New Jersey. 2004

[8] A. Nur Syahrudin and T. Kurniawan, "Input dan Output pada Bahasa Pemrograman Python," *Jurnal Dasar Pemogaraman Python STMIK*, Nov. 2018.

[9] M. Younes and A. Jantan, "Image Encryption using Block-Based Transformation Algorithm," *IAENG Int J Comput Sci*, vol. 35, Nov. 2008.

[10] I. Zeynep, *Chaos and Chaotic Maps*. Belmont: Cankaya University, 2009.