

## PENGACAKAN CITRA DIGITAL DENGAN METODE TRANSFORMASI FIBONACCI DAN LUCAS

Riah Ukur Ginting, Dini M Hutagalung

Program Studi Sistem Informasi Universitas Sari Mutiara Indonesia, Medan, Indonesia  
Email: <sup>1</sup>riahukur@gmail.com <sup>2</sup>mhdini@gmail.com

### Abstrak

Citra adalah fungsi 2 (dua) dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat yang menunjukkan ruang, dan nilai dari  $f(x,y)$  disebut intensitas citra pada koordinat tersebut. Citra digital adalah citra yang memiliki nilai  $x, y, f(x,y)$  yang terbatas. Citra digital terdiri dari sejumlah elemen yang terbatas, dimana setiap elemen tersebut memiliki koordinat dan nilai. Elemen – elemen tersebut disebut piksel (Gonzalez and Woods., 2007). Salah satu metode pengacakan pada citra adalah *Arnold Transformation* dimana citra ditransformasikan dengan menggunakan *cat map*, yaitu matriks berukuran  $2 \times 2$ . Setelah dilakukan transformasi maka nilai setiap piksel mendapatkan koordinat baru sampai gambar tidak mempunyai bentuk aslinya lagi. Akibat dengan adanya permasalahan di atas, maka penulis menggunakan dasar *Arnold Transformation*, untuk mengubah nilai *cat map* dengan deret *Fibonacci* dan *Lucas* untuk perancangan aplikasi ini agar hasil pengacakan memiliki tingkat keamanan yang lebih tinggi. Tujuan dari penelitian ini adalah (1) melakukan pengacakan citra digital dengan menggunakan transformasi *Fibonacci* dan *Lucas* sehingga citra tidak memiliki bentuk aslinya lagi dan (2) menyimpan citra hasil pengacakan ke dalam memori penyimpanan. Manfaat dari penelitian ini adalah (1) Perancangan aplikasi pengacakan citra digital, sehingga keamanan citra meningkat dan (2) Keefektifan metode transformasi *fibonacci* dan *lucas* dalam melakukan pengacakan citra digital.

**Kata kunci:** *Citra Digital, Fibonacci dan Lucas*

### Abstract

The image is a function of 2 (two) dimensions  $f(x, y)$ , where  $x$  and  $y$  are the coordinates that show the space, and the value of  $f(x, y)$  is called the image intensity at the coordinates. Digital image is an image that has a limited value of  $x, y, f(x, y)$ . Digital images consist of a limited number of elements, where each element has coordinates and values. These elements are called pixels (Gonzalez and Woods., 2007). One method of randomization in the image is *Arnold Transformation* where the image is transformed using a *cat map*, which is a  $2 \times 2$  matrix. After transformation, the value of each pixel gets new coordinates until the image does not have its original shape. As a result of the above problems, the author uses the basis of *Arnold Transformation*, to change the value of the map paint with *Fibonacci* and *Lucas* series for designing this application so that the randomization results have a higher level of security. The purpose of this study is (1) to do digital image randomization using *Fibonacci* and *Lucas* transforms so that the image does not have its original shape anymore and (2) save the randomized images into storage memory. The benefits of this study are (1) the design of digital image randomization applications, so that image security increases and (2) the effectiveness of the *Fibonacci* and *lucas* transformation methods in digital image randomization.

**Keywords:** *Image Processing, Fibonacci and Lucas*

## 1. PENDAHULUAN

Citra adalah fungsi 2 dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat yang menunjukkan ruang, dan nilai dari  $f(x,y)$  disebut intensitas citra pada koordinat tersebut. Citra digital adalah citra yang memiliki nilai  $x, y, f(x,y)$  yang terbatas. Citra digital terdiri dari sejumlah elemen yang terbatas, dimana setiap elemen tersebut memiliki koordinat dan nilai. Elemen – elemen tersebut disebut piksel (Gonzalez and Woods., 2007).

Li, Min et.al (2013) mengatakan bahwa pengacakan citra digital dapat merubah citra menjadi citra yang transformasi yang dapat menyembunyikan informasi, dimana pada tranformasi citra digital dikenal juga dengan penyamaran informasi. Yang dikemukakan oleh Mishra, Minati et.al (2012) mengatakan dengan perkembangan yang cepat pada teknologi informasi, khususnya internet dan jaringan yang dapat menjadikan pertukaran data menjadi semakin mudah dan cepat. Salah satu metode pengacakan pada citra adalah *Arnold Transformation* dimana citra ditransformasikan dengan menggunakan *cat map*, yaitu matriks berukuran  $2 \times 2$ . Setelah dilakukan transformasi maka nilai setiap piksel mendapatkan koordinat baru sampai gambar tidak mempunyai bentuk aslinya lagi.

Akibat dengan adanya permasalahan di atas, maka penulis menggunakan dasar *Arnold Transformation*, untuk mengubah nilai *cat map* dengan deret *Fibonacci* dan *Lucas* untuk perancangan aplikasi ini agar hasil pengacakan

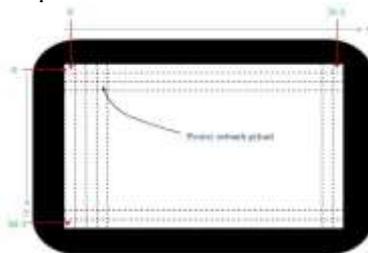
memiliki tingkat keamanan yang lebih tinggi. Berdasarkan hal tersebut, penulis mengangkat judul “Aplikasi Pengacakan Citra Digital Dengan Metode Transformasi *Fibonacci* dan *Lucas*”.

## 2. TEORITIS

### 2.1 Citra

Citra adalah kumpulan piksel-piksel yang disusun dalam larik dua-dimensi yang dikemukakan oleh Ahmad Usman (2005). Dimana indeks baris dan kolom (x,y) dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas cira, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Menurut Petrou Maria dan Petrou Costas (2010) citra digital adalah citra  $f(x,y)$  dimana telah dipetakan menjadi *array* 2 dimensi. Pada masing-masing elemen *array* memiliki nilai yang mewakili intensitas warna masing-masing piksel.

Menurut Abdul Kadir (2013) citra digital dibentuk oleh kumpulan titik yang dinamakan piksel. Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan pada Gambar 1.



Gambar 1. Sistem koordinat citra berukuran  $M \times N$  (M baris dan N kolom)

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut adalah sebagai berikut :

#### 1. Citra Berwarna ( Citra RGB)

Citra berwarna atau citra RGB, adalah citra yang terdiri dari komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan bit (nilainya berkisar 0 – 255). Dengan kemungkinan tersebut maka warna yang dapat disajikan mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna.

#### 2. Citra Berskala Keabuan ( Citra Grayscale)

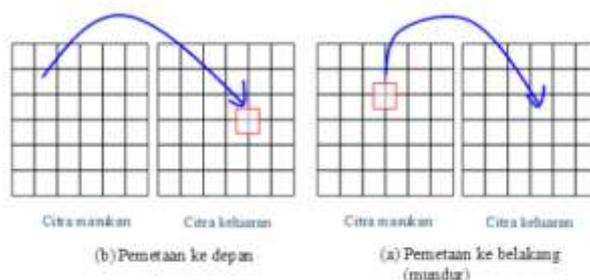
Sesuai dengan namanya, citra ini memiliki rentang warna hitam dan putih, dimana akan menghasilkan efek warna abu-abu. Pada jenis gambar ini dinyatakan dengan intensitas dengan rentang nilai 0-255, dimana nilai 0 menyatakan warna hitam, dan nilai 255 menyatakan warna putih.

#### 3. Citra Biner

Citra biner merupakan citra yang hanya memiliki dua nilai saja, yaitu 0 dan 1, dimana nilai 0 menyatakan warna hitam, dan nilai 1 menyatakan warna putih. Citra biner banyak digunakan untuk pemrosesan citra digital, misalnya untuk memperoleh tepi suatu objek.

### 2.2 Operasi Geomatrik Pada Citra

Menurut Kadir Abdul (2013) operasi geometrik adalah operasi pada citra yang dilakukan secara geometris seperti translasi, rotasi, dan penyekalaan. Pada operasi seperti ini terdapat pemetaan geometric, yang menyatakan hubungan pemetaan antara piksel pada citra masukan dan citra keluaran. Secara prinsip, terdapat dua cara yang dapat dipakai. Pertama yaitu pemetaan ke depan dan kedua berupa pemetaan ke belakang yang terdapat pada Gambar 2.



Gambar 2. Pemetaan Geomatrik

### 2.3 Pengacakan Citra

Xiangdong, LIU et.al (2008) mengatakan bahwa tujuan utama pengacakan citra adalah untuk mengubah suatu citra yang bermakna menjadi citra tidak bermakna atau acak dengan tujuan meningkatkan keamanan.

Menurut Hamdnaalla, Khalid et.al (2013) pengacakan citra mengacu pada pengubahan suatu citra menjadi citra lain yang sama sekali berbeda dengan citra awal, dimana pengguna hanya mengetahui algoritma dan *key* yang digunakan agar dapat mengembalikan citra.

### 2.4 Transformasi Arnold

Citra digital dapat dilihat sebagai matriks dua dimensi. Ketika citra digital memiliki ukuran  $N$ , maka citra memiliki elemen sebanyak  $N \times N$ . Setiap elemen memiliki koordinat  $x$  dan  $y$ . Setelah citra diproses dengan transformasi maka untuk setiap pasangan  $(x,y)$ , maka koordinatnya menjadi  $(x',y')$  dimana artinya koordinat  $(x,y)$  berpindah ke koordinat baru  $(x',y')$ .

Bentuk transformasi Arnold

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (1)$$

Dimana  $x, y \in \{0, 1, \dots, N-1\}$  dan  $N$  adalah ukuran citra.

#### Transformasi Arnold Umum

Tingkat keamanan dari citra yang telah terenkripsi menjadi rendah ketika menggunakan transformasi perhitungan (1), karena dapat di dekripsi oleh orang lain dengan menggunakan *map* yang sama. Untuk meningkatkan tingkat keamanan dari perhitungan (1), maka *map* yang digunakan diubah menjadi :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k+1 & k \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (2)$$

Atau

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k & k+1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (3)$$

Dimana  $x, y \in \{0, 1, 2, \dots, N-1\}$  dan  $N$  adalah ukuran citra dan  $k \in \{0, 1, 2, 3, \dots\}$

Tidak seperti perhitungan (1), perhitungan (2) dan (3) memiliki variasi *maps* sebanyak  $k$ , sehingga meningkatkan keamanan dari citra yang dienkripsi dari percobaan dekripsi secara paksa oleh orang yang tidak diinginkan.

#### Transformasi Fibonacci Umum

Deret *Fibonacci*  $F_n$  adalah deret bilangan bulat dengan hubungan :

$$F_n = \begin{cases} 0, & \text{untuk } n = 1 \\ 1, & \text{untuk } n = 2 \\ F_{n-1} + F_{n-2} & \text{dan seterusnya} \end{cases} \quad (4)$$

Dapat dilihat bahwa matriks  $2 \times 2$  yang dibentuk dari deret *Fibonacci* merupakan matriks unimodular, dan dapat dianggap sebagai pengacak citra. Secara umum transformasi *Fibonacci* dapat didefinisikan sebagai berikut :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} F_i & F_{i+1} \\ F_{i+2} & F_{i+3} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (5)$$

Dimana  $x, y \in \{0, 1, 2, \dots, N-1\}$  dan  $N$  adalah ukuran citra dan  $F_i$  adalah suku ke  $-i$  dari deret *Fibonacci*.

Dengan cara ini kita dapat membentuk berbagai variasi bentuk transformasi untuk nilai  $i$  yang berbeda. Sama seperti transformasi (2) dan (3), mempunyai tingkat keamanan yang lebih tinggi.

#### Transformasi Fibonacci – Lucas

Diambil dari nama matematikawan Prancis bernama Francois Edouard Anatole Lucas. Bentuk deret *Lucas*:

$$L_n = \begin{cases} 2, & \text{untuk } n = 1 \\ 1, & \text{untuk } n = 2 \\ L_{n-1} + L_{n-2} & \text{dan seterusnya} \end{cases} \quad (6)$$

Deret ini menghasilkan deret angka : 2, 1, 3, 4, 7, 11, 18, 29...

Tidak seperti deret *Fibonacci*, deret *Lucas* tidak menghasilkan matriks unimodular sehingga tidak bisa digunakan untuk pengacakan citra digital, tetapi dengan menggabungkan deret *Fibonacci* dan *Lucas* kita dapat membentuk *maps* yang dapat digunakan untuk pengacakan citra digital.

Dengan menggunakan deret *Fibonacci* di atas dan digabungkan dengan deret *Lucas* maka bentuk transformasi baru dapat didefinisikan sebagai berikut :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} F_i & F_{i+1} \\ L_i & L_{i+1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N} \quad (7)$$

Dimana  $x, y \in \{0, 1, 2 \dots N-1\}$ ,  $L_i$  adalah suku ke  $-i$  dari deret *Lucas*,  $F_i$  adalah suku ke  $-i$  dari deret *Fibonacci*. dan  $N$  adalah ukuran citra.

Contoh deret dengan perhitungan (6) :

$$F(11)LT_1 = \begin{pmatrix} F_1 & F_2 \\ L_1 & L_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \quad (8)$$

Dengan cara ini dapat dibuat transformasi dengan variasi yang tidak terbatas. Sama dengan transformasi *Arnold* dan transformasi *Fibonacci*, transformasi ini mempunyai keamanan yang lebih tinggi karena banyaknya variasi *maps* yang ada.

### 3. ANALISA DAN PEMBAHASAN

#### 3.1 Analisa Masalah

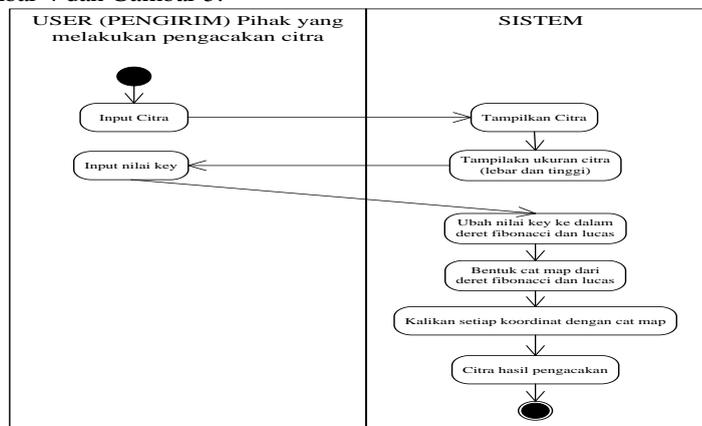
Perancangan sistem menggunakan *Unified Modelling Language(UML)*, yaitu *use case* dan *activity diagram*, dimana UML yang dibuat hanya menjelaskan interaksi sistem terhadap *user* dan tidak menjelaskan detail dari proses perhitungan.

Pada diagram *use case*, menunjukkan interaksi antara pengguna dan system yang terdiri dari pengirim (pihak yang melakukan pengacakan citra) dan penerima (pihak yang melakukan pengembalian citra). Diagram *use case* dapat dilihat pada Gambar 3.

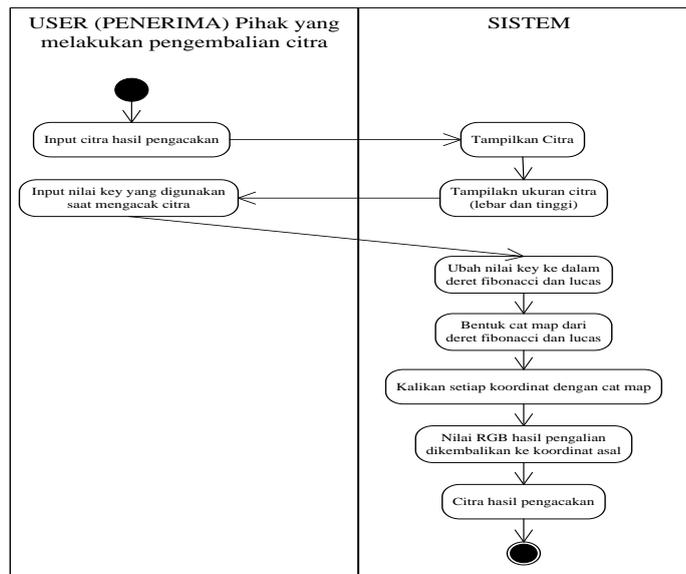


Gambar 3. Use Case Aplikasi Pengacakan Citra

Secara umum, proses yang terjadi di dalam aplikasi dapat digambarkan dalam bentuk *Activity Diagram* yang dapat dilihat pada Gambar 4 dan Gambar 5.



Gambar 4. Activity Diagram Pengirim



Gambar 5. Activity Diagram Penerima

### 3.2 Perancangan Sistem

Perancangan aplikasi pengacakan citra dengan metode transformasi *Fibonacci* dan *Lucas* menggunakan bahasa pemrograman Microsoft Visual Basic .Net 2010. *Form* aplikasi terdiri atas satu *form* utama.

#### 1. Form Utama

*Form* utama merupakan tampilan utama aplikasi, di mana *users* dapat melakukan pengacakan dan pengembalian citra. Pada *form* utama *users* dapat melakukan *input* berupa *file* citra yang akan diacak, atau *file* citra yang akan dikembalikan beserta *key* yang digunakan. *Key* yang digunakan pada pengacakan dan pengembalian citra harus sama, karena jika berbeda maka citra tidak bisa dikembalikan. Untuk perancangan *form* utama pengacakan citra dapat dilihat pada Gambar 6 berikut.



Gambar 6. Form Utama(Pengacakan Citra)

Pada proses pengacakan citra, citra dan *key* harus dimasukkan terlebih dahulu agar pengacakan citra dapat dilakukan, dengan batasan nilai *key* antara 1-80. Untuk melakukan pengembalian citra, maka *key* yang dimasukkan juga harus sama dengan nilai *key* yang digunakan pada pengacakan. Rancangan tampilan pengembalian citra dapat dilihat pada Gambar 7 berikut.



Gambar 7. Form Utama(Pengembalian Citra)

Tampilan yang terakhir adalah tampilan *about*. Tampilan ini berisi informasi tentang nama penulis dan nama kampus. Berikut tampilan *about* dapat dilihat pada Gambar 8 berikut.

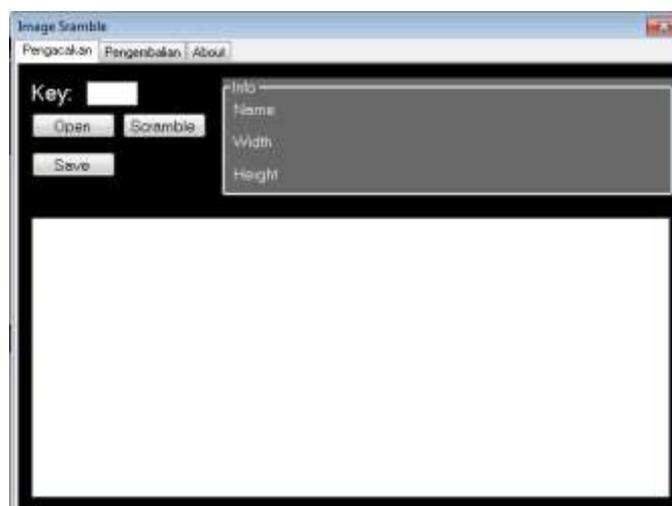


Gambar 8. Form Utama(About)

#### 4. IMPLEMENTASI

##### 4.1 Implementasi Sistem

Saat pertama kali aplikasi dijalankan maka akan muncul tampilan *form* utama seperti Gambar 9.



Gambar 9. Tampilan Form Utama

Pada Gambar 9. terlihat *form* utama terdiri dari 3 *tab*, yaitu *tab* pengacakan, pengembalian dan *about*. Untuk melakukan pengacakan citra tekan tombol “Open”, lalu pilih citra yang hendak diacak, maka citra akan muncul beserta informasi citra. Setelah itu masukkan nilai ”Key” antara 1-124. Tampilan proses awal pengacakan citra dapat dilihat pada Gambar 10.



Gambar 10. Proses Awal Pengacakan Citra

Untuk memulai proses pengacakan, tekan tombol ”Scramble”. Saat pengacakan citra selesai maka tampilan citra akan berubah. Hasil pengacakan citra dapat dilihat pada Gambar 11.



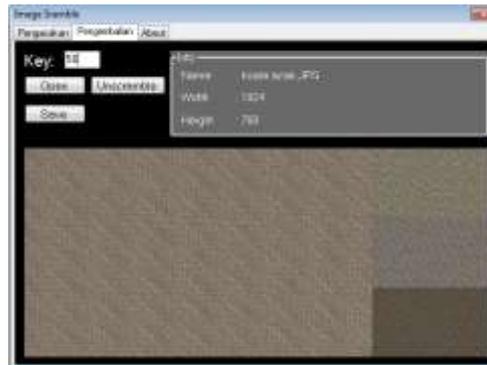
Gambar 11. Hasil Pengacakan Citra

Setelah proses pengacakan selesai, maka *users* dapat menyimpan citra dengan menekan tombol “Save”. Tampilan proses pengacakan citra dapat dilihat pada Gambar 12.



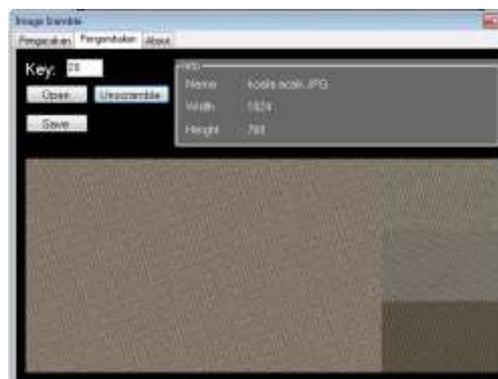
Gambar 12. Hasil Pengacakan Citra

Untuk pengembalian citra, pilih *tab* “Pengembalian”. Setelah itu tekan tombol “Open” untuk memasukkan citra yang hendak dikembalikan. Masukkan nilai *key* yang sama dengan nilai *key* yang digunakan untuk mengacak citra. Tampilan awal pengembalian citra dapat dilihat pada Gambar 13.

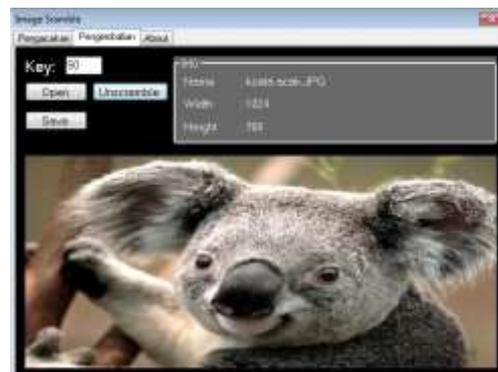


Gambar 13. Proses Awal Pengembalian Citra

Untuk mengembalikan citra, tekan tombol “Unscramble”, setelah itu tunggu sampai proses pengembalian citra selesai. Jika nilai *key* yang digunakan tidak sama dengan nilai *key* yang digunakan pada pengacakan maka citra tidak akan kembali ke citra awal. Tampilan proses pengembalian yang gagal dan berhasil dapat dilihat pada Gambar 14. dan Gambar 15.



Gambar 14. Proses Pengembalian Gagal



Gambar 15. Proses Pengembalian Berhasil

#### 4.2. Pengujian

Pengujian dilakukan pada 5 (lima) *file* citra. Pengujian termasuk pengacakan citra dan pengembalian citra. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

No.	Nama Citra	Ukuran Citra	File Berhasil Diacak	File Berhasil Dikembalikan
1.	Jellyfish.jpg	757 KB	YA	YA
2.	Koala.jpg	762 KB	YA	YA
3.	Desert.jpg	826 KB	YA	YA
4.	Lena128.tif	26.1 KB	YA	YA

---

5.	Penguins.jpg	759 KB	YA	YA
----	--------------	--------	----	----

---

## 5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka diambil beberapa kesimpulan sebagai berikut:

1. Proses pengacakan dan pegembalian citra dengan metode transformasi *Fibonacci* dan *Lucas* tidak mempengaruhi kualitas citra secara kasat mata.
2. Semakin besar nilai *key* yang digunakan pada proses pengacakan, maka semakin acak pula citra yang dihasilkan.

## 6. REFERENSI

- [1] Ahmad, Usman. 2005. Pengolahan Citra Digital & Teknik Pemogramannya. Yogyakarta. Graha Ilmu.
- [2] Gonzalez, Rafael C. and Woods, Richard E. 2007. *Digital Image Processing (3<sup>rd</sup> Edition)*. University of Tennessee.
- [3] Hamdnaalla, Khalid et.al. 2013. "Digital Image Confidentiality Depends upon Arnold Transformation and RC4 Algorithms" in *International Journal of Video & Image Processing and Network Security*, Vol:13 No:04. China.
- [4] Kadir, Abdul and Adhi Susanto. 2013. Teori dan Aplikasi Pengolahan Citra. Yogyakarta. ANDI Yogyakarta.
- [5] Li, Min et.al. 2013."Arnold Transform Based Image Scrambling Method" in *3<sup>rd</sup> International Conference on Multimedia Technology*.
- [6] Mishra, Minati et.al. 2012."Image Encryption Using Fibonacci-Lucas Transformation" in *International Journal on Cryptography and Information Security*, Vol. 2 No.3.India.
- [7] Petrou, Maria and Petrou, Costas. 2010. *Image Processing: The Fundamentals*. WILEY
- [8] Rumbaugh, James et.al. 2004. *The Unified Modeling Language Reference Manual Second Edition*. Addison-Wesley.
- [9] Xiangdong, LIU et.al. 2008. "Image Scrambling Based on Chaos Theory and Sorting Transformation" in *International Journal of Computer Science and Network Security*, VOL.8 No.1. China.