PERANCANGAN APLIKASI PENGACAKAN CITRA MENGGUNAKAN *M-SEQUENCE* BERDASARKAN PARAMETER

Kristian Telaumbanua[1], Susanto[2]

Program Studi Teknik Informatika, STMIK Mikroskil
Jl. Thamrin No. 122, 124, 140 Medan 20212
kristian@mikroskil.ac.id[1], 091111799@students.mikroskil.ac.id[2]

Abstrak

Pengiriman data citra melalui jaringan komunikasi menghadapi masalah sekuritas dimana data citra kemungkinan besar dapat diakses dan diperoleh pihak lain yang memiliki koneksi ke jaringan, sehingga diperlukan algoritma pengacakan citra. Pengacakan citra (*image scrambling*) adalah sebuah alat bagus untuk membuat citra teracak tidak dapat diidentifikasi secara visual dan sulit untuk didekripsi oleh pemakai yang tidak berhak. Salah satu algoritma yang dapat digunakan untuk melakukan pengacakan citra adalah algoritma *M-Sequence*. Proses kerja dari perangkat lunak dimulai dari pemilihan citra yang akan diacak dan nilai parameter p dan r. Setelah itu, proses dilanjutkan dengan mengacak citra input berdasarkan nilai parameter sehingga diperoleh citra teracak. Citra yang diperoleh tersebut dapat dikembalikan menjadi citra semula dengan menggunakan algoritma *un-scrambling* dengan mengisi kunci yang sama dengan yang digunakan pada saat pengacakan. Aplikasi yang dihasilkan dapat melakukan pengacakan terhadap citra input dan mengembalikan citra semula dengan menggunakan proses *un-scrambling*. Selain itu, aplikasi juga menyediakan fitur untuk menampilkan laporan detail proses perhitungan dan fitur perbandingan untuk membandingkan citra asli dan citra hasil *un-scrambling.*

Kata kunci: *citra digital, scrambling, un-scrambling, algoritma m-sequence*

1. Pendahuluan

Pengiriman data citra melalui jaringan komunikasi menghadapi masalah sekuritas dimana data citra kemungkinan besar dapat diakses dan diperoleh pihak lain yang memiliki koneksi ke jaringan, sehingga diperlukan algoritma pengacakan citra. Pengacakan citra (image scrambling) adalah sebuah alat bagus untuk membuat citra teracak tidak dapat diidentifikasi secara visual dan sulit untuk didekripsi oleh pemakai yang tidak berhak.

M-Sequence, juga disebut sebagai deretan dengan panjang maksimum (*maximum length sequence*), adalah sebuah tipe dari deretan rekursif biner semi acak (*pseudorandom binary recursive sequence*) yang dapat dibangkitkan dengan maksimal *linear feedback shift register*. *M-sequence* digunakan secara luas pada komunikasi digital seperti komuninkasi spektrum tersebar, gangguan semi acak (*pseudo random noise*). Beberapa penerapan dari *M-sequence* telah diperkenalkan terutama yang fokus pada sifat *pseudo random*-nya. *M-sequence* adalah sebuah deretan biner periodik dengan karakteristik korelasi otomatis, sehingga deretan ini dapat dikembangkan menjadi area aplikasi baru dari pengacakan citra [3]. Yicong Zhou, Karen Panetta dan Sos Agaian memperkenalkan sebuah parameter baru berdasarkan pada *M-sequence* dan sebuah algoritma pengacakan citra baru menggunakan *M-sequence* ini. Citra teracak sulit untuk di-*decode* karena kunci sekuritas, parameter pergeseran r dan parameter

jarak p memiliki banyak kemungkinan. Algoritma Zhou dan partner ini dapat digunakan untuk mengacak citra 2D seperti citra biner, citra *grayscale* dan citra berwarna dengan 3 komponen [3]

Berdasarkan uraian latar belakang diatas, maka yang menjadi permasalahan dalam penyusunan penelitian ini adalah:

a. Pengiriman data citra melalui jaringan komunikasi menghadapi masalah sekuritas dimana data citra kemungkinan besar dapat diakses dan diperoleh pihak lain yang memiliki koneksi ke jaringan, sehingga diperlukan algoritma pengacakan citra.
b. Bagaimana merancang sebuah aplikasi yang menyediakan *interface* untuk pengisian nilai parameter sehingga dapat dilakukan pengujian terhadap berbagai nilai parameter berbeda.

Tujuan penelitian ini adalah untuk membuat sebuah aplikasi pengacakan citra dengan menggunakan *M-Sequence* berdasarkan parameter.

Manfaat dari penelitian ini adalah:

a. Aplikasi dapat digunakan untuk melakukan pengamanan citra digital dengan menerapkan algoritma *M-Sequence* berdasarkan parameter.
b. Sebagai referensi tambahan dalam mempelajari mengenai algoritma pengacakan citra

2. Kajian Teoritis

a. Pengacakan Citra

Inti dari pengacakan citra adalah untuk mengurangi korelasi dari posisi piksel dan korelasi dari nilai piksel hingga menjadi tidak relevan. Proses pengacakan citra dapat dianggap sebagai proses perbaikan ketidakpastian, dan juga sebagai proses penambahan jumlah informasi citra. Korelasi dari piksel citra alami memiliki jumlah yang terbesar dalam blok dan ketidakpastian [2]. Untuk mengenkripsi data citra digital, banyak teknik enkripsi yang telah dikemukakan oleh para ahli. Salah satunya adalah algoritma pengacakan citra *M-Sequence* berdasarkan parameter.

b. M-Sequence Berdasarkan Parameter.

M-Sequence klasik adalah sebuah deretan biner periodik yang dapat dihasilkan dengan sederetan *shift register* dengan operasi modulo 2.

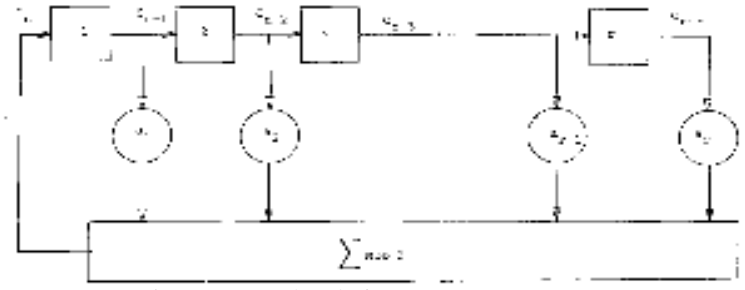
Definisi 1., *M-Sequence* Biner Klasik:

M-Sequence biner klasik $\{m_k\}$ dipenuhi dengan operasi berikut:

$$m_k = \sum_{i=1}^{n} a_i m_{k-i} \pmod 2$$

dimana n adalah jumlah *shift register*, $m_k = 0, 1$ dan $a_i = 0, 1$ adalah koefisien dari *shift register* ke-i. Sirkuit yang mengimplementasikan operasi diatas disebut sebagai *M-Sequence generator*.

Gambaran dari sebuah *shift register* dapat dilihat pada gambar berikut:



Gambar 1. *Simple Shift Register Generator* [1]

Shift register untuk gambar 1 diatas memiliki rumusan matematis sebagai berikut:

$$C_n = \sum_{k=1}^{r} a_k C_{n-k} \pmod 2; \qquad a_r = 1.$$

(Raymond L. Pickholtz, Donald L. Schilling dan Laurence B. Milstein, 1982)

Output dari *M-Sequence generator* tergantung pada koefisien (a_k) dan nilai awal dari *register*. Output dari *M-Sequence* adalah deretan biner dengan panjang periode maksimum $T = 2^n - 1$. Anggap output dari *M-Sequence* adalah $\{m_k\} = \{m_1, m_2, \ldots, m_T\}$, maka $m_k = m_{k+T} = m_{k+2T} = \ldots$.

Definisi 2., *M-Sequence* Berdasarkan Parameter:

Anggap deretan biner $\{s_1, s_2, \ldots, s_n\}$ adalah nilai awal dari *shift register* dengan jumlah tahapan sebesar n pada *M-Sequence generator*, dan output dari *M-Sequence* adalah $\{b_{r1}, b_{r2}, \ldots, b_{rT}\}$ setelah *register* ini digeser sebanyak r kali. Deretan biner $\{c_{r1}, c_{r2}, \ldots, c_m\}$ disebut *M-Sequence* berdasarkan parameter yang didefinisikan sebagai:

$$c_{ri} = b_{r(i+p)}$$

$\{c_{r1}, c_{r2}, \ldots, c_{rm}\}$ juga disebut sebagai representasi *M-Sequence* dari $\{s_1, s_2, \ldots, s_n\}$. Dimana i, r, p, T adalah bilangan *integer*, dan $1 \le i \le n$, $T = 2^n - 1$, $1 \le r \le T$, $0 \le p \le T - n$. [3]

c. Transformasi M-Sequence

Berdasarkan pada definisi 2.2 diatas, sebuah bilangan desimal dengan representasi biner dari $S = (s_1, s_2, \ldots, s_n)$ dapat ditransformasikan menjadi representasi *M-Sequence* $C_r = (c_{r1}, c_{r2}, \ldots, c_{rn})_2$, dimana C_r adalah bilangan desimal lainnya.

Hal yang sama, sebuah deretan desimal {1, 2, 3, …, N} dapat ditransformasikan menjadi representasi *M-Sequence* $\{C_{r1}, C_{r2}, C_{r3}, \ldots, C_{rN}\}$ yang merupakan deretan permutasi dari {1, 2, 3, …, N}. Lebih lanjut lagi, deretan permutasi $\{C_{r1}, C_{r2}, C_{r3}, \ldots, C_{rN}\}$ akan berbeda ketika parameter pergeseran r dan parameter jarak p memiliki nilai berbeda.

Transformasi permutasi ini dapat diaplikasikan pada pengacakan citra karena transformasi ini dapat mengubah posisi baris dan kolom dari piksel citra. Parameter pergeseran r dan parameter jarak p dapat dianggap sebagai kunci sekuritas untuk menghasilkan deretan berbeda $\{C_1, C_2, C_3, \ldots, C_N\}$.

Untuk nilai tertentu dari r dan p, representasi dari *M-Sequence* {1, 2, 3, …, N} dapat didefinisikan sebagai:

$$C_r = \{C_{r1}, C_{r2}, C_{r3}, \ldots, C_{rN}\}$$

Data citra 2-D disimpan pada sebuah matriks 2-D seperti citra *grayscale* dan citra biner. Untuk mengacak citra 2-D dalam satu langkah, dapat digunakan transformasi *M-Sequence* 2-D.

Definisi 3., Transformasi *M-Sequence* 2-D:

Anggap D adalah sebuah matriks citra M x N, T adalah matriks koefisien baris, T_c adalah matriks koefisien kolom. Transformasi *M-Sequence* 2-D dapat didefinisikan sebagai:

$$S = T_r D T_c$$

dimana S adalah matriks citra teracak dan:

$$T_r(m,n) = \begin{cases} 1 & (m, C_{rn}) \\ 0 & otherwise \end{cases}, \quad T_c(i,j) = \begin{cases} 1 & (C_{ri}, j) \\ 0 & otherwise \end{cases}$$

dimana $1 \le m, n \le M$, $1 \le i, j \le N$

Definisi 4., Invers Transformasi *M-Sequence* 2-D:

Anggap S adalah sebuah matriks citra teracak M x N, T_r^{-1} dan T_c^{-1} adalah matriks invers dari matriks koefisien baris dan kolom yang didefinisikan pada Definisi 2.3, invers transformasi *M-Sequence* 2-D dapat didefinisikan sebagai:

$$R \quad T_r^{-1} S T_c^{-1}$$

dimana R adalah matriks citra yang direkonstruksi. [3]

d. Algoritma Pengacakan Citra Menggunakan Transformasi M-Sequence Berdasarkan Parameter

Data dari sebuah citra 2-D adalah sebuah matriks 2-D. Setelah itu, transformasi *M-Sequence* 2-D akan digunakan untuk mengacak data citra. Matriks koefisien baris dan kolom harus dihitung dengan memilih kunci sekuritas: parameter pergeseran r dan parameter jarak p. Citra teracak dapat dihasilkan dengan mengaplikasikan transformasi *M-Sequence* 2-D pada citra asli dalam satu langkah.

User yang berhak akan diberikan kunci sekuritas untuk merekonstruksi citra asli dalam proses inversnya. Kunci sekuritas akan digunakan untuk menghitung inverns matriks koefisien baris dan kolom. Kemudian, akan dilakukan invers transformasi *M-Sequence* 2-D untuk men-decode data citra teracak. Citra rekonstruksi dapat diperoleh dengan mengaplikasikan invers transformasi pada citra teracak. [3]

3. Kerangka Penelitian

Adapun langkah-langkah dalam menyelesaikan penelitian ini, mengacu pada pendekatan waterfall, namun tidak semua tahapan ini dilakukan karena menyesuaikan kebutuhan dalam pengembangannya saja, adapun tahapannya sebagai berikut :

a. Analisis
   Memahami kebutuhan perangkat lunak, fungsi-fungsi, unjuk kerja dan antarmuka yang diperlukan. Pada proses ini, akan dideskripsikan proses kerja sistem dengan menggunakan *activity diagram* dan memodelkan sistem menggunakan *use case diagram*. Setelah itu, akan dilakukan proses analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional dari sistem.
b. Desain Sistem
   Tahapan yang berfokus pada empat atribut dari program, yaitu : struktur data, arsitektur perangkat lunak, detil suatu prosedur, dan karakteristik antarmuka. Rancangan *interface* menggunakan Microsoft Visual Basic 2008.
c. Pemrograman (*Coding*)
   Aktivitas yang mengubah hasil rancangan menjadi bentuk yang dapat dimengerti komputer, biasanya dalam bentuk program. Membuat kode program (*source code*) dengan menggunakan Microsoft Visual Basic 2008.
d. Pengujian (*Testing*)
   Setelah pengkodean selesai, maka akan dilakukan pengujian program. Pengujian dilakukan untuk menemukan kesalahan serta memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Melakukan pengujian terhadap sistem dengan menggunakan berbagai input parameter berbeda. Pengujian yang dilakukan mencakup:
   i. Menghitung jumlah titik pasti pada citra teracak. Disebut sebagai titik pasti dari skema pengacakan jika koordinat piksel tidak berubah setelah pengacakan. Semakin sedikit titik pasti dari skema, maka semakin ekfektif skema tersebut, dan sekuritas dari skema semakin tinggi.
   ii. Menghitung rata-rata jarak pergerakan dari pengacakan dengan rumusan berikut:

$$\|D\|_2 = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{(w-i)^2+(v-j)^2}$$

dimana (i, j) merepresentasikan koordinat piksel dari sebuah titik pada citra asli, (w, v) merepresentasikan koordinat piksel dari titik tersebut pada citra teracak. Semakin besar jarak pergerakan rata-rata dari skema pengacakan, maka semakin sedikit hubungan antara citra asli dan citra teracak dan semakin tinggi efisiensi dari skema.

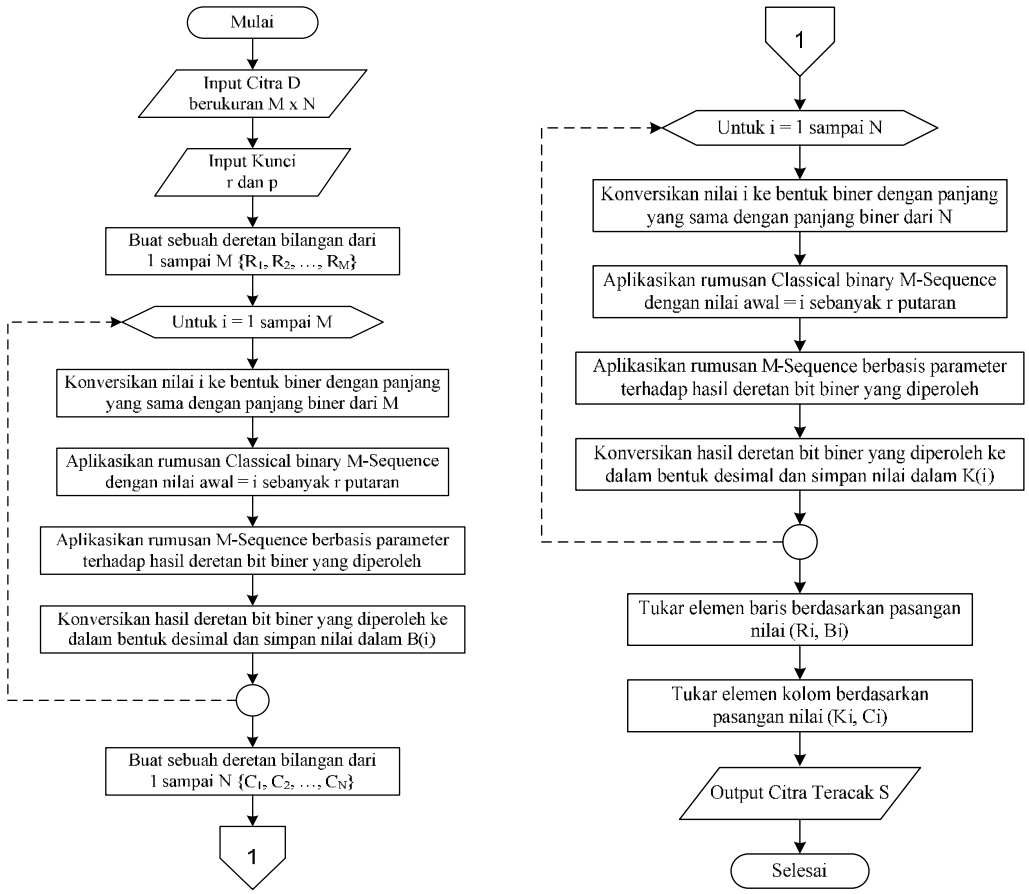
e. Mengambil kesimpulan dari hasil pengujian

4. Analisis dan Perancangan Sistem

a. Analisis Proses Kerja

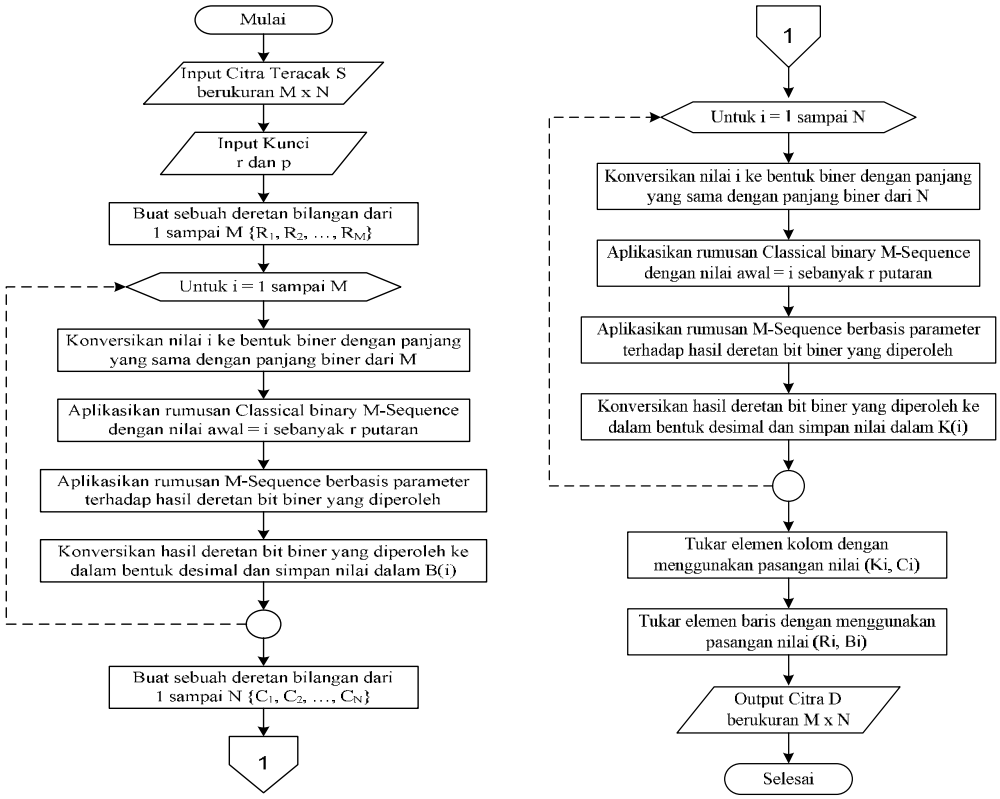
- Proses Pengacakan Citra

Proses ini digunakan untuk melakukan pengacakan terhadap citra input dengan menggunakan kunci yang juga akan digunakan pada saat proses rekonstruksi citra. Kunci input mencakup nilai parameter pergeseran r, nilai parameter jarak p dan nilai a sebagai keadaan awal LFSR. Kunci input ini juga harus diketahui oleh penerima citra teracak agar dapat melakukan proses rekonstruksi untuk memperoleh citra semula. *Flowchart* proses kerja dari proses pengacakan citra dapat dilihat pada gambar 1. berikut:



Gambar 1. *Flowchart* Proses Pengacakan Citra

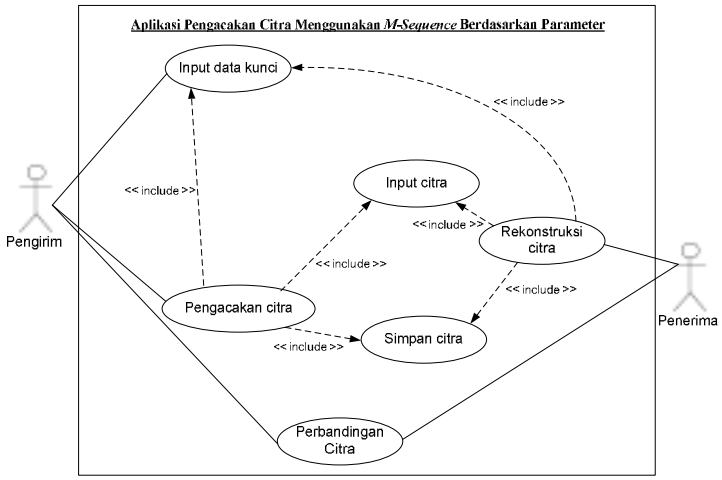
- **Proses Rekonstruksi Citra**
  Proses ini digunakan untuk melakukan proses rekonstruksi citra untuk memperoleh kembali citra semula. Kunci yang digunakan pada proses pengacakan citra harus diketahui agar dapat memperoleh kembali citra semula. *Flowchart* proses kerja dari proses rekonstruksi citra dapat dilihat pada gambar berikut:



Gambar 2. *Flowchart* Proses Rekonstruksi Citra

b. Pemodelan Sistem

Sistem akan dimodelkan dengan menggunakan *use case diagram* seperti terlihat pada gambar 3 berikut:



Gambar 3. Usecase Aplikasi

5. Hasil Program dan Pengujian

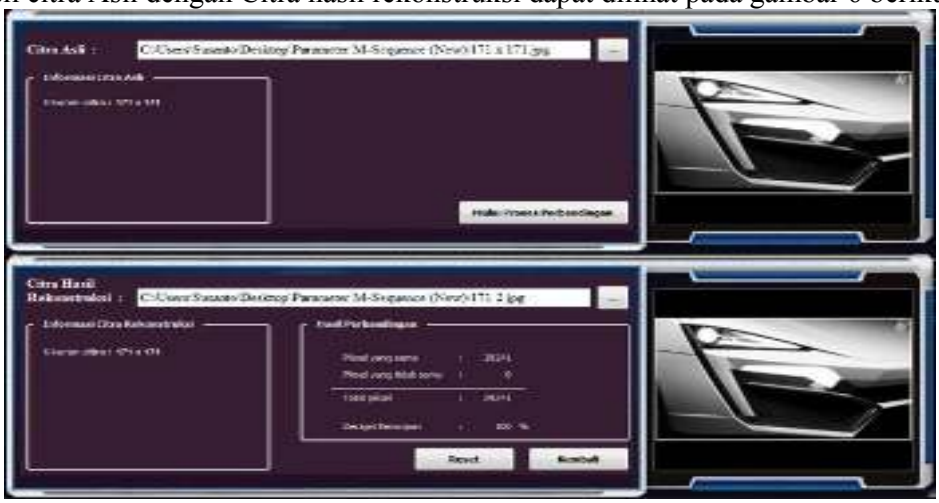
a. Hasil Program

Menu utama dari aplikasi ini dapat dilihat pada gambar 4 berikut ini :

Gambar 4. Menu Utama Aplikasi

Gambar 5. Tampilan *Form Scrambling*

Hasil citra Asli dengan Citra hasil rekonstruksi dapat dilihat pada gambar 6 berikut ini.



Gambar 6. Hasil Citra asli dan citra rekonstruksi

b. Pengujian Aplikasi

Hasil proses pengujian dengan menggunakan beberapa ukuran citra dan kunci yang berbeda dapat dirincikan sebagai berikut:

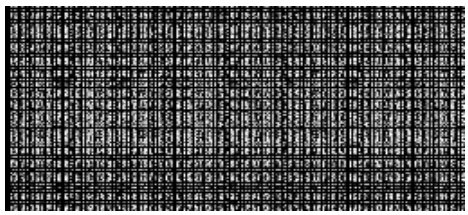
Pengujian menggunakan beberapa kunci berbeda

Dimensi : 180 x 100

Gambar 7 Input Citra 180 x 100.jpg

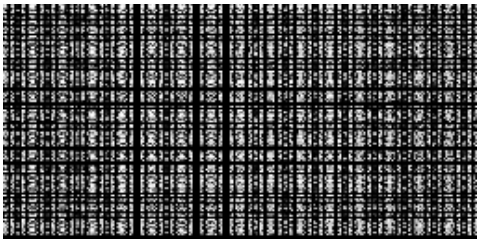
Pengujian akan dilakukan dengan menggunakan beberapa nilai kunci berikut:

a. r = 100, p = 80



Gambar 8 Citra Hasil Pengacakan terhadap Citra 180 x 100 acak.jpg
Lama waktu eksekusi : 1.50 detik
Ukuran citra hasil pengacakan : 255 x 127

b. r = 11, p = 44



Gambar 9. Citra Hasil Pengacakan terhadap Citra 180 x 100 acak 2.jpg
Lama waktu eksekusi : 1.84 detik
Ukuran citra hasil pengacakan : 255 x 127

Dari hasil beberapa pengujian menggunakan beberapa ukuran citra dan kunci berbeda yang dilakukan maka dapat dilihat hasilnya pada tabel 1 berikut ini.

Tabel 1. Hasil Pengujian dengan Menggunakan Beberapa Ukuran Citra dan Kunci Berbeda

Ukuran Citra Input	Kunci yang Digunakan		Jumlah Titik Pasti	Derajat Kemiripan	Rata-rata Jarak Pergerakan
	Nilai r	Nilai p			
127 x 127	5	5	302	1.872 %	1258.6629
127 x 127	5	10	373	2.312 %	1283.0429
127 x 127	5	20	389	2.411 %	1295.3520
127 x 127	5	30	408	2.529 %	1293.8190
127 x 127	5	40	403	2.498 %	1314.3766
127 x 127	10	5	302	1.872 %	1258.6628
127 x 127	20	5	302	1.872 %	1258.6628

Ukuran Citra Input	Kunci yang Digunakan		Jumlah Titik Pasti	Derajat Kemiripan	Rata-rata Jarak Pergerakan
	Nilai r	Nilai p			
127 x 127	30	5	302	1.872 %	1258.6628
127 x 127	40	5	302	1.872 %	1258.6628
127 x 127	10	10	373	2.312 %	1283.0429
127 x 127	20	20	389	2.411 %	1295.3520
127 x 127	30	30	408	2.529 %	1293.8190
127 x 127	40	40	403	2.498 %	1314.3766
255 x 255	5	5	574	0.882 %	3646.8384
255 x 255	5	10	572	0.879 %	3750.4640
255 x 255	5	20	572	0.880 %	3701.0219
255 x 255	5	30	600	0.922 %	3725.6422
255 x 255	5	40	612	0.941 %	3716.9418
255 x 255	10	5	574	0.882 %	3646.8384
255 x 255	20	5	574	0.882 %	3646.8384
255 x 255	30	5	574	0.882 %	3646.8384
255 x 255	40	5	574	0.882 %	3646.8384
255 x 255	10	10	572	0.879 %	3750.4640
255 x 255	20	20	572	0.880 %	3701.0219
255 x 255	30	30	600	0.922 %	3725.6422
255 x 255	40	40	612	0.941 %	3716.9418
511 x 511	5	5	1076	0.412 %	10250.1935
511 x 511	5	10	1105	0.423 %	10468.0665
511 x 511	5	20	1157	0.443 %	10439.5853
511 x 511	5	30	1083	0.415 %	10486.3568
511 x 511	5	40	1146	0.439 %	10408.2139
511 x 511	10	5	1076	0.412 %	10250.1936
511 x 511	20	5	1076	0.412 %	10250.1936
511 x 511	30	5	1076	0.412 %	10250.1936
511 x 511	40	5	1076	0.412 %	10250.1936
511 x 511	10	10	1105	0.423 %	10468.0665
511 x 511	20	20	1157	0.443 %	10439.5853
511 x 511	30	30	1083	0.415 %	10486.3568
511 x 511	40	40	1146	0.439 %	10408.2139

Dari hasil pengujian diatas, dapat dilihat bahwa:

- Dimensi citra hasil tidak sama dengan citra input karena dimensi citra hasil tergantung pada panjang maksimal LFSR yang digunakan. Panjang maksimal LFSR ini dapat ditentukan dengan menggunakan rumusan $2^n - 1$, dimana n adalah panjang bit dari jumlah baris dan jumlah kolom citra. Hal ini mengakibatkan ukuran citra hasil akan lebih besar daripada ukuran citra input. Oleh karena itu, agar citra semula dapat direkonstruksi kembali, maka perlu dilakukan pengisian ukuran citra semula.
- Citra hasil pengacakan sudah acak karena tidak memiliki informasi apapun mengenai citra input. Hal ini dapat dilihat pada hasil pengujian yang dilakukan di atas.
- Lama proses pengacakan sama dengan lama proses rekonstruksi. Hal ini dapat dilihat pada hasil pengujian di atas. Proses pengacakan dan rekonstruksi memiliki waktu yang

relatif sama karena metode ini menggunakan algoritma yang sama untuk proses pengacakan dan rekonstruksi.

- Citra hasil pengacakan yang dihasilkan cukup teracak. Hal ini dapat dibuktikan dari hasil pengujian yang dilakukan dimana proses penghapusan ataupun penambahan warna pada bagian tertentu pada citra hasil pengacakan akan tersebar pada keseluruhan citra hasil rekonstruksi.
- Berdasarkan hasil dari tabel pengujian diatas, diketahui bahwa kunci dari nilai p lebih berpengaruh terhadap jumlah titik pasti dan rata-rata jarak pergerakan daripada nilai r. Hal ini berarti bahwa perubahan terhadap nilai p akan mengakibatkan terjadinya perubahan jumlah titik pasti dan rata-rata jarak pergerakan antara citra asli dan citra teracak.
- Derajat kemiripan antara citra asli dan citra teracak sangat kecil (di bawah 1 %). Hal ini berarti bahwa citra teracak yang dihasilkan sangat bagus.
- Nilai rata-rata jarak pergerakan dari pengacakan sangat besar. Hal ini berarti bahwa hubungan antara citra asli dan citra teracak sangat kecil dan tingkat efisiensi dari skema sangat tinggi.

6. Kesimpulan

Setelah menyelesaikan pembuatan perangkat lunak ini, penulis dapat menarik beberapa kesimpulan sebagai berikut:

a. Dimensi citra hasil tidak sama dengan citra input karena dimensi citra hasil tergantung pada panjang maksimal LFSR yang digunakan.
b. Citra hasil pengacakan sudah relatif acak karena tidak memiliki informasi apapun mengenai citra input.
c. Lama proses pengacakan dan rekonstruksi relatif sama.
d. Citra hasil pengacakan yang dihasilkan cukup teracak.
e. Perangkat lunak menyediakan *interface* untuk mengisi input nilai parameter p dan r sehingga dapat digunakan untuk menguji algoritma *image scrambling.*
f. Citra hasil pengacakan dapat disimpan ke dalam sebuah *file* citra sehingga dapat dikirimkan melalui jaringan komunikasi kepada penerima.

Referensi

[1] Raymond L. Pickholtz, Donald L. Schilling dan Laurence B. Milstein, (1982). Theory of Spread-Spectrum Communications – A Tutorial, IEEE Transactions on Communications, Vol. Com-30, No, 5.

[2] Xue Yang, Xiaoyang Yu, Qifeng Zou dan Jiaying Jia (2010). *Image Encryption Algorithm Based on Universal Modular Transformation*, Information Technology Journal 9 (4): 680-685.

[3] Yicong Zhou, Karena Panetta dan Sos Agaian (2008). *An Image Scrambling Algorithm Using Parameter Based M-Sequences*, USA.