KRIPTOSISTEM MENGGUNAKAN ALGORITMA GENETIKA PADA DATA CITRA

Magdalena Ariance Ineke Pakereng
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga, 50711
Email: inekep200472@yahoo.com

***ABSTRACT**: Data and information security either those to be sent through the communication network or those to be kept in a device has brought into attention of the user of this information and data. Information and data security is being kept by making the data and information are not able to be read or known by unauthorized users. This can be done by using the cryptography technique. There are many cryptography techniques has been created to meet the above purpose. This research dealt with a new method in symmetric cryptosystem to encrypt digital image data using genetic algorithm. This method uses some components of genetic algorithm, the crossover operation and mutation operation, crossover rate and mutation rate, and fitness function. The process includes creating the encryption and decryption process using the crossover and mutation process. The crossover process illustrates the transposition technique, while the mutation process illustrates the substitution technique. This cryptosystem is applied to the 8-bit grayscale image using 2 (two) keys, random seed and number of generation. The result of the research shows that cryptosystem using the genetic algorithm is possible to be applied to the digital image. With 7000 or more as number of generation, plain image is possible to be encrypted into unrecognized cipher image, and by measuring the similarity of the image based on the pixel (pixel based similarity), the result of decipher image is similar with the plain image. The keys of the random seed affect the durability of the result of encryption process, while the sum of the generation affects the randomization of the cipher image and the running time (the time needed for the process).*

***Keywords**: Genetic Algorithm, Digital Image, Cryptosystem, Symmetric.*

PENDAHULUAN

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Keamanan data dan informasi dilakukan dengan cara membuat data dan informasi tersebut tidak terbaca atau diketahui oleh orang yang tidak berwenang. Oleh karena itu, informasi atau data rahasia yang akan dikirim harus disandikan agar tidak dapat dibaca oleh orang lain. Hal ini dapat dicapai dengan menggunakan implementasi teknik kriptografi.

Keamanan data tidak hanya diperuntukkan bagi data atau informasi yang akan dikirim melalui jaringan komunikasi, tetapi bisa juga merupakan data atau informasi yang akan disimpan dalam media penyimpanan. Data atau infomasi tidak hanya berupa data teks, tetapi juga dapat berupa data citra (*image*), data suara/bunyi (*audio*) dan video.

Algoritma Genetika adalah suatu metode pencarian (*search*) acak yang didasarkan atas prinsip evolusi yang terjadi di alam, individu-individu yang mampu beradaptasi dengan lingkungan di mana ia berada akan tetap hidup sedangkan yang tidak, akan mati. Algoritma genetika diperkenalkan oleh John Holland dari Universitas Michigan, Amerika Serikat dan termasuk salah satu metode terbaru dalam bidang kecerdasan buatan.

Penelitian dilakukan melalui percobaan laboratorium dengan menggunakan data citra digital dan komputer. Data citra digital dikumpulkan dengan menerapkan teknik *sampling*.

KRIPTOGRAFI

Kriptografi berasal dari kata Yunani *kripto* (tersembunyi) dan *grafia* (tulisan). Secara harfiah, kriptografi dapat diartikan sebagai tulisan yang tersembunyi atau tulisan yang dirahasiakan. Tujuannya adalah supaya tulisan tersebut tidak dapat dibaca oleh setiap orang. Hanya orang-orang tertentu, yaitu orang yang mengetahui cara menyembunyikan tulisan tersebut yang dapat membacanya.

Dalam perkembangannya, kriptografi didefinisikan sebagai ilmu yang berhubungan dengan prinsip-prinsip atau metode-metode mentransformasikan pesan ke dalam bentuk yang tidak dimengerti, kemudian ditransformasikan kembali ke dalam bentuk pesan asli yang dimengerti. Kriptografi adalah ilmu yang mempelajari bagaimana membuat suatu pesan yang dikirim dapat disampaikan kepada penerima

dengan aman [5]. Pesan asli yang dimengerti isinya/ maknanya ini dinamakan *plaintext*. Pesan yang tidak dimengerti, yang merupakan hasil transformasi dari *plaintext*, disebut *ciphertext.*

Beberapa istilah yang berhubungan dengan transformasi dari *plaintext* ke *ciphertext* di antaranya adalah *cipher*, kunci, *encipher*, *decipher*, kriptanalisis, dan kriptologi. *Cipher* adalah algoritma yang digunakan untuk melakukan transformasi *plaintext* ke *ciphertext.* Kunci adalah beberapa informasi kritis yang diperlukan *cipher*, di mana kunci ini hanya diketahui oleh pengirim dan penerima pesan. *Encipher* adalah proses mengkonversikan *plaintext* ke *ciphertext* dengan menggunakan *cipher* dan kunci. *Decipher* adalah mengkonversikan *ciphertext* kembali ke *plaintext* dengan menggunakan *cipher* dan kunci. Kriptanalisis (*cryptanalysis)* adalah studi yang berkenaan dengan prinsip-prinsip atau metode-metode mentransformasikan *ciphertext* kembali ke *plaintext* tanpa mengetahui kunci untuk *decipher*. Sedangkan kriptologi (*cryptologi*) adalah bidang ilmu yang berhubungan dengan kriptografi dan kriptanalisis.

Sebuah kriptosistem terdiri dari lima komponen (P, C, K, ε, D), yang memenuhi kondisi sebagai berikut [7]:

(1) P adalah himpunan berhingga *plaintext*
(2) C adalah himpunan berhingga *ciphertext*
(3) K adalah himpunan kunci
(4) K adalah himpunan kunci yang memenuhi aturan bahwa untuk setiap $p \in P$, dan untuk setiap $k \in K$, maka ada $\varepsilon_K \in \varepsilon$ dan $c \in C$ sehingga $c=\varepsilon_K(p)$. Selanjutnya untuk setiap $\varepsilon_K \in \varepsilon$ dan $d_K \in D$ sedemikian sehingga $d_K(\varepsilon_K(p))=p$, untuk setiap $p \in P$.

Sementara itu sistem kriptografi dapat diklasifikasikan ke dalam 3 dimensi yang independen, yaitu [6]:

(1) Tipe operasi yang digunakan untuk mentransformasikan *plaintext* ke *ciphertext*. Semua algoritma enkripsi berdasarkan pada dua prinsip utama, yaitu substitusi dan transposisi. Substitusi adalah di mana tiap-tiap elemen dari *plaintext* (bit, huruf) dipetakan dengan elemen yang lain. Sedangkan transposisi adalah di mana tiap-tiap elemen dari *plaintext* diatur kembali posisinya atau urutannya. Beberapa sistem kriptografi menggunakan prinsip substitusi dan transposisi sekaligus.
(2) Kunci yang digunakan. Jika pengirim dan penerima pesan menggunakan kunci yang sama, sistem dinamakan simetrik (*symmetric*), kunci tunggal (*singel key*), kunci rahasia (*secret key*), atau enkripsi konvensional (*conventional encryption*). Jika antara pengirim dan penerima pesan menggunakan kunci yang berbeda, sistem dinamakan asimetrik (*asymmetric*), kunci dobel (*double key*), atau enkripsi kunci publik (*public key encryption*).
(3) Cara memproses *plaintext*. *Block cipher* memproses *plaintext* dalam input blok-blok bit dan output juga dalam blok-blok bit. *Stream cipher* memproses *plaintext* dalam input bit demi bit secara kontinyu.

Dalam sistem kriptografi (*cryptographic system*), yang selanjutnya disebut kriptosistem (*cryptosystem*), tahap *encipher* dikenal dengan enkripsi dan tahap *decipher* dikenal dengan dekripsi. Suatu kriptosistem umumnya dapat memenuhi keutuhan (*integrity*), autentikasi (*authenticity*), dan nonrepudiasi (*nonrepudiation*) [4,5,6]. Kerahasiaan dijamin karena pesan yang ditransmisikan dienkripsi terlebih dahulu. Sementara itu keutuhan, autentikasi, dan nonrepudiasi, dapat diverifikasi oleh penerima pesan dengan memanfaatkan tanda tangan digital (*digital signature*).

Dalam perkembangannya ada dua jenis algoritma kriptosistem, yaitu algoritma enkripsi kunci simetrik (*symmetric-key encryption algorithm*) dan algoritma enkripsi kunci publik (*public-key encryption algorithm*). Algoritma enkripsi kunci simetris menggunakan kunci yang sama, atau disebut juga kunci rahasia, baik untuk enkripsi maupun dekripsi. Karenanya kunci ini harus dirahasiakan oleh pengirim dan penerima pesan, supaya *ciphertext* tetap aman dari gangguan serangan penyusup. Algoritma enkripsi ini sering disebut dengan enkripsi konvensional. Pengembangan algoritma enkripsi konvensional menggunakan metode substitusi dan/atau metode transposisi, sehingga dikenal adanya *cipher* substitusi, *cipher* transposisi, dan *product cipher*.

Algoritma enkripsi kunci publik menggunakan dua kunci yang berpasangan tetapi berbeda. Satu kunci dipakai untuk enkripsi, satu kunci lainnya dipakai untuk dekripsi. Pengembangan algoritma enkripsi kunci publik, yang selanjutnya disebut kriptosistem kunci publik, lebih banyak menggunakan pendekatan matematis untuk memetakan *plaintext* ke *ciphertext* dan sebaliknya.

ALGORITMA GENETIKA

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi aturan "yang kuat adalah yang menang" [2]. Darwin juga menyatakan bahwa kelangsungan hidup suatu makhluk dapat dipertahankan melalui proses reproduksi, *crossover*, dan mutasi. Konsep dalam teori evolusi Darwin

tersebut kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih "alamiah".

Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai kromosom, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai *gen* dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan *generasi*. Dalam tiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (*fungsi_objektif*) menggunakan ukuran yang disebut dengan *fitness*. Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan *seleksi*. Proses seleksi kromosom menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai *fitness* tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya.

Kromosom-kromosom baru yang disebut dengan *offspring*, dibentuk dengan cara melakukan perkawinan antar kromosom-kromosom dalam satu generasi yang disebut sebagai proses *crossover*. Jumlah kromosom dalam populasi yang mengalami *crossover* ditentukan oleh paramater yang disebut dengan *crossover_rate*. Mekanisme perubahan susunan unsur penyusun mahkluk hidup akibat adanya faktor alam yang disebut dengan *mutasi* direpresentasikan sebagai proses berubahnya satu atau lebih nilai *gen* dalam kromosom dengan suatu nilai acak. Jumlah *gen* dalam populasi yang mengalami mutasi ditentukan oleh parameter yang dinamakan *mutation_rate*. Setelah beberapa generasi akan dihasilkan kromosom-kromosom yang nilai gengennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh algoritma genetika terhadap permasalahan yang ingin diselesaikan.

Algoritma genetika berbeda dengan metode–metode optimasi dan prosedur pencarian konvensional dalam beberapa hal yang sangat fundamental yaitu sebagai berikut [2]:

- Algoritma genetika bekerja dengan sekumpulan kode solusi, bukan solusi itu sendiri.
- Algoritma genetika mencari dari populasi solusi, bukan solusi tunggal.
- Algoritma genetika menggunakan fungsi *fitness*, bukan turunan atau pengetahuan bantu lain.
- Algoritma genetika menggunakan aturan probabilistik, bukan deterministik.

Skema dasar dari algoritma genetika, sebagai berikut [1]:

1. Buat populasi secara random dengan n kromosom (solusi yang pantas terhadap masalah)
2. Evaluasi nilai *fitness f(x)* dari setiap kromosom x di populasi
3. Buat populasi baru dengan mengulangi langkah berikut sampai populasi baru lengkap
   - Pilih dua kromosom *parent* dari populasi berdasarkan nilai *fitness*-nya (nilai *fitness* terbaik biasanya mempunyai kesempatan terbesar untuk dipilih)
   - Dengan kemungkinan *crossover* maka *crossover* kedua *parent* untuk membentuk *offspring* baru (anak). Jika *crossover* tidak terjadi maka anak adalah sama seperti *parent*-nya.
   - Dengan kemungkinan mutasi maka mutasi *offspring* baru pada tiap bagiannya (posisi dalam kromosom).
   - Tempatkan *offspring* baru pada populasi baru.
4. Gunakan populasi baru yang telah digenerasi untuk digunakan selanjutnya dalam algoritma.
5. Jika kondisi telah terpenuhi maka perulangan berhenti dan berikan solusi terbaik dari populasi, jika kondisi tidak terpenuhi kembali ke langkah 2.

CITRA DIGITAL

Citra didefinisikan sebagai suatu fungsi intensitas cahaya dua dimensi $f(x,y)$ dimana x dan y menunjukkan koordinat spasial, dan nilai f pada suatu titik (x,y) sebanding dengan tingkat kecerahan (*gray level*) dari citra di titik tersebut. Citra digital adalah citra dengan $f(x,y)$ yang nilainya didigitalisasikan (dibuat diskrit) baik dalam koordinat spasialnya maupun dalam tingkat kecerahannya.

Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Fungsi $f(x,y)$ merupakan fungsi yang memiliki dua unsur. Unsur pertama adalah kekuatan sumber cahaya yang melingkupi pandangan terhadap objek (*illumination*), sedangkan unsur kedua adalah intensitas cahaya yang direfleksikan oleh objek (*reflectance components*). Kedua unsur tersebut masing-masing dapat dituliskan sebagai fungsi $i(x,y)$ dan $r(x,y)$. Sehingga $f(x,y)$ dapat ditulis sebagai :

$$f(x,y) = i(x,y)\ x\ r(x,y) \qquad (1)$$

yang mana $0 \langle i(x,y) \langle \infty$ dan $0 \langle r(x,y) \langle 1$.

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik

tersebut. Titik-titik tersebut dinamakan sebagai elemen citra atau *pixel (picture element)* [3].

Citra digital dapat direpresentasikan dalam bentuk matriks M × N

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2)$$

yang mana elemen-elemen matriks atau piksel menunjukkan nilai keabuan atau warna.

Intensitas dari gambar hitam-putih pada titik (*x, y*) disebut derajad keabuan (*graylevel*), yang mana derajad keabuannya bergerak dari hitam ke putih, sedangkan citranya disebut citra hitam-putih (*grayscale image*) atau citra monokrom (*monochrome image*).

Derajad keabuan memiliki rentang nilai dari *lmin* sampai *lmax*, atau dapat ditulis *lmin* < *f* < *lmax* dan selang (*lmin*, *lmax*) disebut skala keabuan. Biasanya selang (*lmin*, *lmax*) sering digeser untuk alasan-alasan praktis menjadi (0, L), yang mana nilai intensitas 0 menyatakan hitam, nilai intensitas L menyatakan putih, sedangkan nilai intensitas antara 0 sampai L bergeser dari hitam ke putih.

Citra *grayscale* disebut juga citra abu-abu. Intensitas nilai *f* pada koordinat titik *f(x,y)*, merupakan derajad keabuan (*gray level*) pada titik tersebut [3].

Pada citra *grayscale* (8 bit) nilai warna primer (merah, hijau, biru) mempunyai nilai yang sama yaitu antara 0 - 255. Citra *gray* merupakan citra dua dimensi, yang direpresentasikan ke dalam sebuah matriks. Posisi baris dan kolom pada matriks menunjukkan posisi piksel pada citra, sedangkan warna piksel adalah nilai yang tersimpan dalam citra.

KRIPTOSISTEM MENGGUNAKAN ALGORITMA GENETIKA PADA DATA CITRA

Metode Kriptosistem

Kriptosistem yang akan dibuat dalam penelitian ini merupakan kriptosistem bersifat simetrik, yang menggunakan pendekatan algoritma genetika lengkap, yaitu mengeksploitasi komponen-komponen generasi, populasi, *crossover*, *crossover rate*, mutasi, *mutation rate* dan *fungsi fitness*, dan diimplementasikan pada citra *grayscale* (8-bit). Untuk proses enkripsi, *plain-image* berformat bitmap (*.bmp) akan dienkripsi dengan kunci yang ditentukan sehingga menghasilkan *cipher-image* yang juga dalam format bitmap (*.bmp). Untuk proses dekripsi, *cipher-image* berformat bitmap (*.bmp) didekripsi dengan kunci yang sama dengan proses enkripsi sehingga diperoleh *decipher-image* yang sama dengan *plain-image.*

Pengkodean Kromosom

Dalam penelitian ini, kriptosistem akan dibangun dengan transposisi blok pada citra yang dilakukan dengan menggunakan algoritma genetika. Dalam metode ini, citra dibagi ke dalam sejumlah blok yang masing-masing bloknya berukuran 8 piksel. Untuk citra dengan ukuran 256x256 piksel akan terdapat 8192 blok, alamat blok ini dikodekan ke dalam 13 digit nilai biner. Tiga belas digit ini yang kemudian dijadikan kode kromosom bersama dengan delapan piksel yang berada dalam blok tersebut. Pengkodean kromosom ditunjukkan pada Gambar 1. yang mana baris pertama kode adalah alamat blok dan baris kedua adalah piksel-piksel yang terdapat dalam alamat tersebut.

1	1	0	1	1	0	1	0	1	1	0	1	1
		10	13	11	12	31	32	33	31			

Gambar 1. Pengkodean kromosom

Inisialisasi Populasi

Jumlah kromosom untuk populasi awal ditentukan berdasarkan formula ukuran populasi (Goldberg, 1989): $1{,}65\ x\ 2^{0.21xL}$, yang mana L menyatakan panjang untai kode kromosom yang disandikan dalam kode biner dan menggunakan nilai acak antara 0 dan 1. Dalam penelitian ini, panjang untai kode kromosom yang disandikan dalam kode biner adalah 13, maka untuk L = 13, jumlah kromosom untuk populasi awal diperoleh sebagai berikut:

$$1{,}65 \text{ x } 2^{0.21\text{x}13} = 1{,}65 \text{ x } 6{,}63 = 10{,}94 \approx 11$$

Kunci Enkripsi dan Dekripsi

Kunci yang digunakan merupakan kunci yang bersifat simetrik, dimana proses enkripsi dan proses dekripsi menggunakan kunci yang sama. Dalam penelitian ini, kunci yang digunakan terdiri dari 2 kunci, yaitu kunci 1 adalah nilai *Random seed* dan kunci 2 adalah jumlah generasi.

Algoritma Enkripsi.

Prosedur enkripsi dengan menggunakan algoritma genetika untuk citra *grayscale* 8-bit berukuran m x n adalah sebagai berikut:

Langkah (1): Data citra I(W x H), dimana W dan H masing-masing adalah lebar dan tinggi citra dibagi ke dalam sejumlah blok dengan ukuran 8 piksel setiap blok. Dalam penelitian ini ukuran citra yang digunakan adalah 256x256 piksel sehingga akan terdapat 8192 blok. Dengan

demikian hanya akan terdapat 8192 kemungkinan kromosom.

Langkah (2): Mendefinisikan semua kemungkinan kromosom dengan kode kromosom seperti pada Gambar 1.

Langkah (3): Menentukan secara acak kromosom untuk populasi awal. Dalam penelitian ini ukuran populasi awal adalah 11 kromosom. Penentuan sebelas kromosom tersebut dilakukan dengan mengambil secara acak dari kemungkinan kromosom yang ada. Bilangan acak dibangkitkan berdasarkan nilai *random seed* yang digunakan sebagai kunci.

Langkah (4): Operasi *crossover*.

- Menghitung nilai *fitness* untuk seluruh kromosom dalam generasi tersebut dengan menggunakan rumus:

$$f(x) = \frac{\sum_{i=1}^{i=13} P(i)}{13} \quad (3)$$

yang mana,
P(i) = jumlah bit bernilai 1 dalam kromosom

- Menentukan *crossover rate* secara random yang diperoleh dengan membangkitkan bilangan acak antara 0 ... 8191, bilangan acak yang diperoleh dibagi dengan 8191.
- Memilih kromosom induk dengan cara membandingkan nilai *fitness* setiap kromosom dengan nilai *crossover rate* yang mana hanya kromosom yang memiliki nilai *fitness* lebih kecil dari *crossover rate* yang akan menjadi induk dalam proses *crossover*.
- Melakukan operasi *crossover* antara dua kromosom terpilih yang nomor urut kromosomnya berdekatan, hal ini dilakukan sebagai penuntun dalam proses dekripsi. Proses *crossover* dilakukan dengan menukar isi blok pada kedua kromosom.

Langkah (5): Operasi mutasi

- Membangkitkan suatu bilangan acak *p* dari 0 . . . 8191.
- Menentukan posisi mutasi yaitu $pm = (p \bmod 13) + 1$.
- Menentukan *mutation rate* dengan rumus $p/8191$
- Menentukan kromosom yang akan mengalami *mutasi* dilakukan berdasarkan nilai fungsi yang didefinisikan sebagai berikut:

$$g(x) = \frac{\sum_{\substack{i=1 \\ i \neq pm}}^{i=13} P(i)}{12}$$

yang mana,
P(i) = jumlah bit bernilai 1 dalam kromosom
pm = posisi gen yang akan dimutasi

- Hanya kromosom yang memiliki nilai $g(x)$ yang lebih kecil dari *mutation rate* yang mengalami mutasi.
- Mutasi dilakukan dengan menegasikan nilai yang ada dalam gen tersebut.

Langkah (6): Ulangi langkah (4) dan langkah (5) sampai jumlah generasi yang ditentukan dalam kunci terpenuhi.

Algoritma Dekripsi

Prosedur dekripsi dengan menggunakan algoritma genetika untuk citra *grayscale* 8-bit berukuran m x n adalah sebagai berikut:

Langkah (1): Citra hasil enkripsi (*cipher-image*) I'(W x H), dimana W dan H masing-masing adalah lebar dan tinggi citra dibagi ke dalam sejumlah blok dengan ukuran 8 piksel setiap blok.

Langkah (2): Mendefinisikan semua kemungkinan kromosom dengan kode kromosom seperti pada gambar 1.

Langkah (3): Menentukan secara acak 11 kromosom pada populasi awal, berdasarkan nilai *random seed* pada kunci.

Langkah (4): Bangkitkan bilangan acak 0...8191 sebanyak 2 * jumlah generasi, yang mana jumlah generasi diketahui dari kunci yang diberikan. Seluruh bilangan acak yang dibangkitkan disimpan dalam suatu *array* dengan indeks terbalik.

Langkah (5): Mencari seluruh kromosom pada generasi terakhir dengan menggunakan prosedur pada langkah mutasi tetapi dalam prosedur ini tidak dilakukan peng-*update*-an posisi blok pada citra.

Langkah (6): Operasi mutasi.

- Hitung nilai g(x) setiap kromosom dalam generasi tersebut.
- Ambil nilai acak dari *array* yang bersesuaian dengan operasi mutasi dan nomor generasi dalam proses enkripsi.
- Definisikan posisi mutasi yaitu $pm = (acak \bmod 13) + 1$
- Tentukan *mutation rate* yaitu $acak/8191$.
- Pilih kromosom yang akan mengalami mutasi dengan aturan yang sama seperti proses enkripsi.
- Kromosom-kromosom yang terpilih dijamin akan sama seperti dalam proses enkripsi karena nilai $g(x)$ dalam proses enkripsi tidak pernah berubah sebelum dan sesudah mutasi.
- Lakukan operasi mutasi dengan menegasikan posisi gen pada kromosom-kromosom terpilih.

Langkah (7): Operasi *crossover*.
- Hitung nilai *fitness* dari seluruh kromosom dalam generasi ini.
- Pilih bilangan acak yang bersesuaian dengan proses enkripsi. Kemudian bagi bilangan acak yang diperoleh dengan 8191 untuk memperoleh *crossover rate*.
- Pilih kromosom yang akan mengalami *crossover* seperti pada proses enkripsi.
- Lakukan operasi *crossover* seperti pada proses enkripsi dengan urutan terbalik.

Langkah (8): Ulangi langkah (6) dan langkah (7) sebanyak jumlah generasi.

EKSPERIMEN DAN ANALISIS

Untuk eksperimen digunakan dua buah citra seperti yang terlihat pada Gambar 2.

Lena.bmp 256x256 Peppers.bmp 256x256

Gambar 2. Citra *grayscale* untuk eksperimen

Eksperimen Enkripsi – Dekripsi Pada Dua Citra Berbeda

Eksperimen yang dilakukan dengan menggunakan kunci 1 (nilai *Random seed*) = 5 dan kunci 2 (jumlah generasi) = 7000.

Eksperimen dengan nilai kunci 1 adalah 5 dan kunci 2 adalah 7000 yang dilakukan untuk citra lena dan citra peppers menunjukkan bahwa secara visual *cipher-image* tidak sama dengan *plain-image* dan *cipher-image* tidak merepresentasikan suatu bentuk/ objek tertentu yang dapat dikenali. Hasil dekripsi menunjukkan *decipher image* sama dengan *plain-image*. Hasil eksperimen enkripsi dan dekripsi ditampilkan dalam Tabel 1. dan Tabel 2.

Tabel 1. Hasil Eksperimen Enkripsi–Dekripsi Pada Citra Berbeda

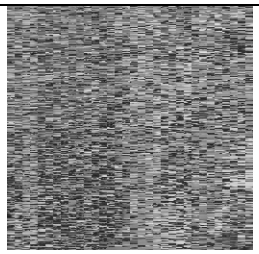
Data Citra	Kunci		Waktu Proses	
	Kunci 1	Kunci 2	Enkripsi	Dekripsi
Lena.bmp	5	7000	625 ms	1047 ms
Peppers.bmp	5	7000	625 ms	1047 ms

Eksperimen Enkripsi Terhadap Sebuah Citra Dengan Kunci 1 Sama dan Kunci 2 Berbeda-beda

Eksperimen yang dilakukan dengan menggunakan kombinasi kunci 1 (nilai *Random seed*) yang sama dan kunci 2 (jumlah generasi) yang berbeda-beda. Eksperimen dengan nilai kunci 1 adalah 10 dan kunci 2 berbeda-beda seperti yang dilakukan untuk citra lena menunjukkan bahwa kunci 2 yaitu jumlah generasi mempengaruhi keter-acak-kan *cipher-image*,

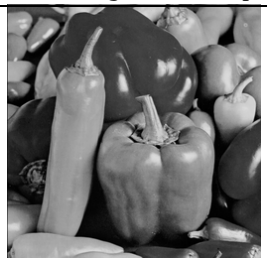
Tabel 2. Citra Hasil Enkripsi – Dekripsi Pada Lena.bmp dan Peppers.bmp

Plain-image – Lena.bmp

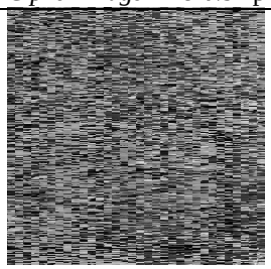


Cipher-image – Lena.bmp

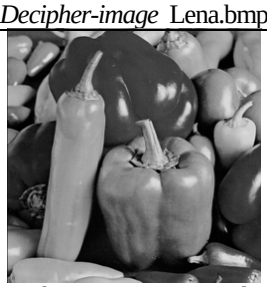
Decipher-image Lena.bmp



Plain-image – Peppers.bmp



Cipher-image – Peppers.bmp



Decipher-image- Peppers.bmp

sekaligus mempengaruhi waktu proses enkripsi, dapat dilihat bahwa semakin tinggi nilai jumlah generasi maka citra semakin acak dan semakin lama waktu proses enkripsi yang dipakai, hasil eksperimen dapat dilihat dalam Tabel 3. dan Tabel 4.

Tabel 3. Hasil Eksperimen Enkripsi Dengan Kunci 1 Sama dan Kunci 2 Berbeda-beda

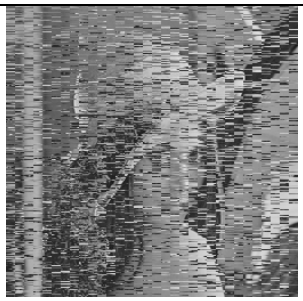
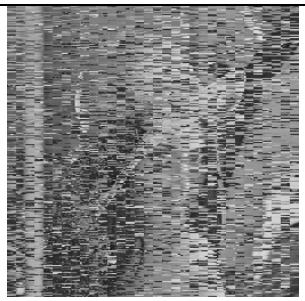
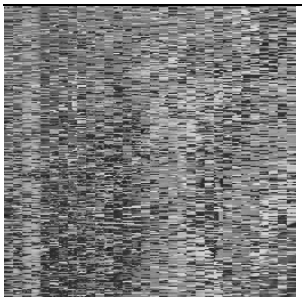
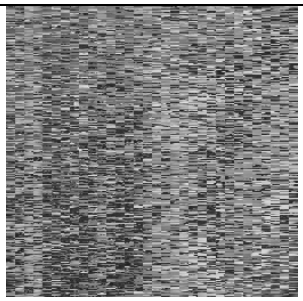
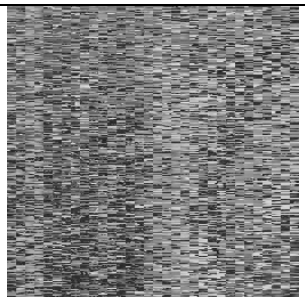
Data ke	Kunci 1	Kunci 2	Waktu Enkripsi (ms)
1	10	1	31
2	10	100	47
3	10	1000	125
4	10	2000	203
5	10	3000	281
6	10	4000	375
7	10	4500	422
8	10	4900	438
9	10	5000	453
10	10	6500	594

Hasil eksperimen dalam Tabel 4., menunjukkan bahwa semakin tinggi jumlah generasi maka *cipher image* yang dihasilkan secara visual semakin tidak dapat dikenali atau dapat dikatakan secara visual *cipher image* berbeda dengan *plain image*. Dapat dilihat juga dari Tabel 4. bahwa dengan jumlah generasi 7000 atau lebih proses enkripsi dapat menghasilkan *cipher image* yang secara visual tidak dapat dikenali.

Eksperimen Enkripsi Terhadap Sebuah Citra Dengan Kunci 1 Berbeda-beda dan Kunci 2 sama

Eksperimen yang dilakukan menggunakan kombinasi kunci 1 (nilai *Random seed*) yang berbeda-beda dan kunci 2 (jumlah generasi) yang sama. Eksperimen dengan nilai kunci 1 berbeda-beda dan kunci 2 sama, seperti yang dilakukan pada citra lena menunjukkan bahwa kunci 1 yaitu nilai *random seed* tidak mempengaruhi waktu proses enkripsi, hasil eksperimen dapat dilihat dalam Tabel 5.

Tabel 4. Citra Hasil Enkripsi dengan Kunci 1 Sama dan Kunci 2 Berbeda-beda

		
Citra asli – lena.bmp 256x256	Kunci 1: 10 Kunci 2: 1 Waktu Proses: 31	Kunci 1: 10 Kunci 2: 100 Waktu Proses: 47
		
Kunci 1: 10 Kunci 2: 1000 Waktu Proses: 125	Kunci 1: 10 Kunci 2: 2000 Waktu Proses: 203	Kunci 1: 10 Kunci 2: 3000 Waktu Proses: 281
		
Kunci 1: 10 Kunci 2: 5000 Waktu Proses: 453	Kunci 1: 10 Kunci 2: 6500 Waktu Proses: 594	Kunci 1: 10 Kunci 2: 7000 Waktu Proses: 625

Tabel 5. Hasil Eksperimen Enkripsi Dengan Kunci 1 Berbeda-beda dan Kunci 2 sama

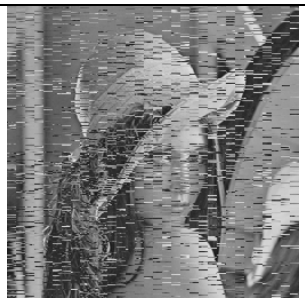
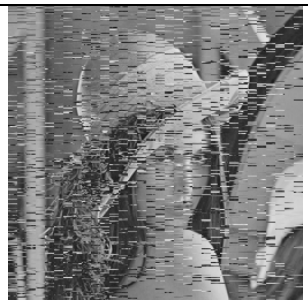
Data ke	Kunci 1	Kunci 2	Waktu Enkripsi (ms)
1	1	1000	125
2	100	1000	125
3	1000	1000	125
4	2000	1000	125
5	3000	1000	125
6	4000	1000	125
7	4500	1000	125
8	4900	1000	125
9	5000	1000	125
10	6500	1000	125

Eksperimen ini juga menunjukkan bahwa besar kecilnya nilai *random seed* dalam kunci 1, tidak mempengaruhi keter-acak-kan *cipher image*, hanya menyebabkan bagian dari citra yang ter-acak berbeda-beda, seperti yang terlihat dalam Tabel 6. Hal ini dikarenakan nilai *random seed* berfungsi sebagai pengendali bilangan random yang dibangkitkan. Nilai *random seed* yang berbeda-beda menyebabkan bilangan random yang diperoleh berbeda-beda pula, tetapi banyaknya bilangan random yang diperoleh sama.

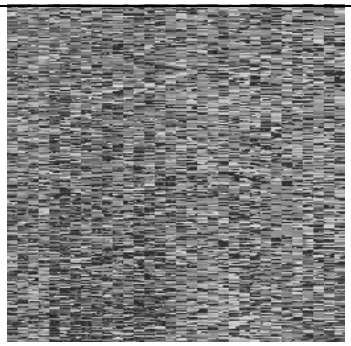
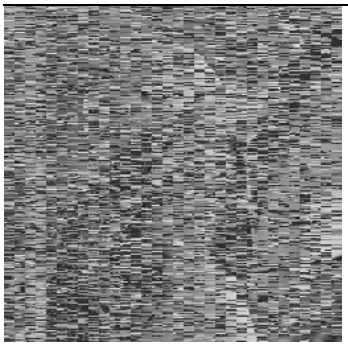
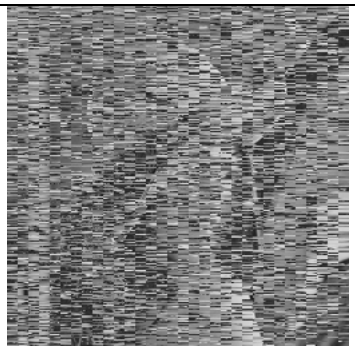
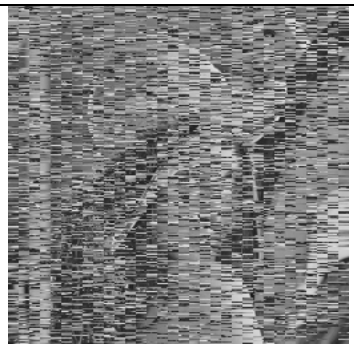
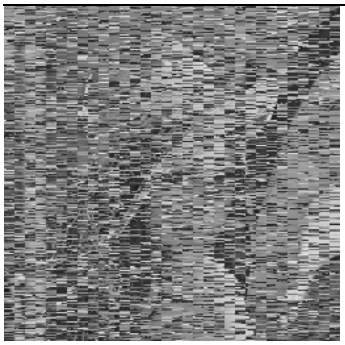
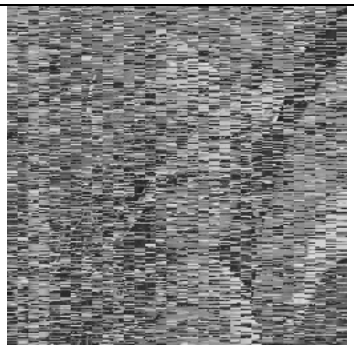
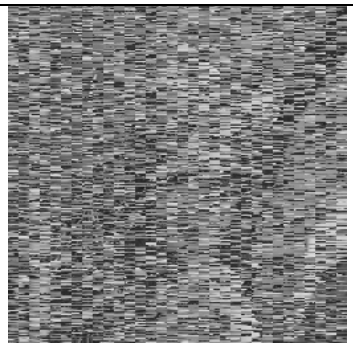
Eksperimen Dekripsi Terhadap Sebuah Citra Dengan Kunci 2 Yang Berbeda Dari Kunci 2 Pada Proses Enkripsi

Eksperimen yang dilakukan menggunakan kunci 1 (nilai *Random seed*) yang sama dan kunci 2 (jumlah generasi) yang berbeda dengan kunci yang digunakan pada proses enkripsi. Untuk eksperimen ini, akan diuji beberapa kombinasi kunci, untuk melihat seberapa

Tabel 6. Citra Hasil Enkripsi dengan Kunci 1 Berbeda-beda dan Kunci 2 sama

 *Plain image* – lena.bmp 256x256	 Kunci 1: 1 Kunci 2: 1000 Waktu Proses: 125	 Kunci 1: 100 Kunci 2: 1000 Waktu Proses: 125
 Kunci 1: 1000 Kunci 2: 1000 Waktu Proses: 125	 Kunci 1: 2000 Kunci 2: 1000 Waktu Proses: 125	 Kunci 1: 3000 Kunci 2: 1000 Waktu Proses: 125
 Kunci 1: 8000 Kunci 2: 1000 Waktu Proses: 125	 Kunci 1: 10000 Kunci 2: 1000 Waktu Proses: 125	 Kunci 1: 20000 Kunci 2: 1000 Waktu Proses: 125

Tabel 7. Citra Hasil Dekripsi dengan Kunci Enkripsi 10 & 10000, Kunci 2 Pada Dekripsi Berbeda-beda

 *Plain image* – lena.bmp 256x256	 *Cipher image* Kunci 1: 10 Kunci 2: 10000 Waktu Proses: 906	 *Decipher image* Kunci 1: 10 Kunci 2: 10000 Waktu Proses: 1515
 Kunci 1: 10 Kunci 2: 7000 Waktu Proses: 1063	 Kunci 1: 10 Kunci 2: 7500 Waktu Proses: 1141	 Kunci 1: 10 Kunci 2: 8000 Waktu Proses: 1219
 Kunci 1: 10 Kunci 2: 12250 Waktu Proses: 1812	 Kunci 1: 10 Kunci 2: 12500 Waktu Proses: 1875	 Kunci 1: 10 Kunci 2: 13000 Waktu Proses: 1953

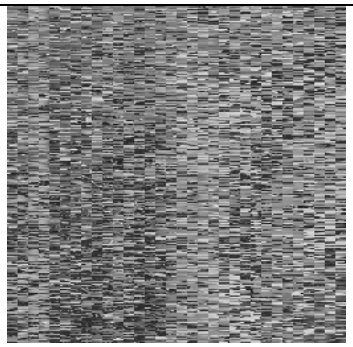
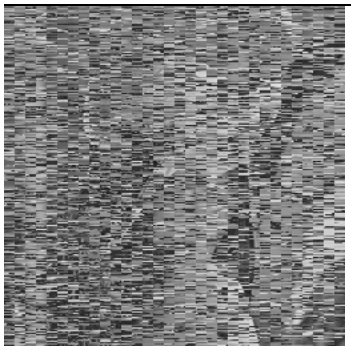
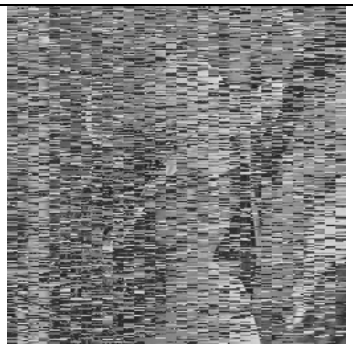
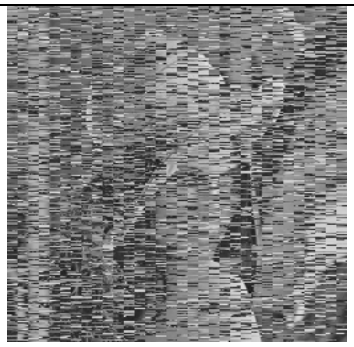
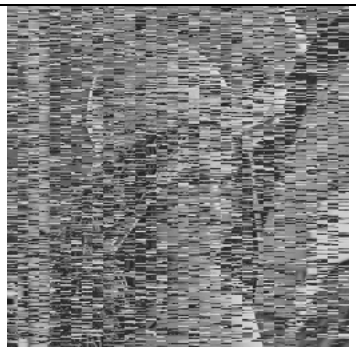
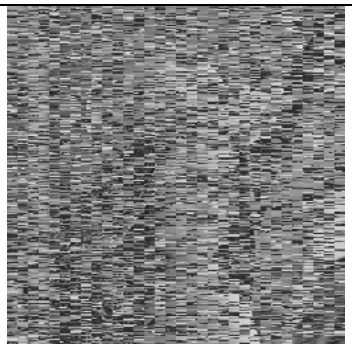
jauh selisih kunci 2 dari kunci 2 pada proses enkripsi, suatu *cipher image* dapat didekripsikan dengan asumsi kunci 1 sama.

Kunci untuk enkripsi, kunci 1 = 10 dan kunci 2 = 10000, *cipher image* akan didekripsi dengan kunci 1 = 10 dan kunci 2 berbeda-beda, hasil dekripsi dapat dilihat pada Tabel 7.

Kunci untuk enkripsi, kunci 1 = 10 dan kunci 2 = 7000, *cipher image* akan didekripsi dengan kunci 1 = 10 dan kunci 2 berbeda-beda, hasil dekripsi dapat dilihat pada Tabel 8.

Berdasarkan data citra hasil dekripsi dalam Tabel 7 dan Tabel 8, hasil eksperimen yang dilakukan pada citra lena menunjukkan bahwa jika kunci 1 pada proses dekripsi sama dengan kunci 1 pada proses enkripsi dan kunci 2 pada proses dekripsi berbeda dengan kunci 2 pada proses enkripsi, maka secara visual sistem menghasilkan *decipher image* yang kurang sempurna, karena kunci 2 yaitu jumlah generasi menentukan banyaknya bilangan random yang dibangkitkan, sehingga jika berbeda, maka banyaknya bilangan random yang dihasilkan akan berbeda, dengan demikian *cipher image* bisa didekripsikan, tetapi tidak sempurna.

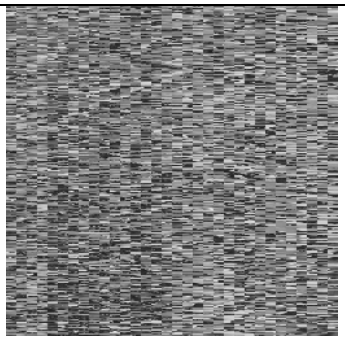
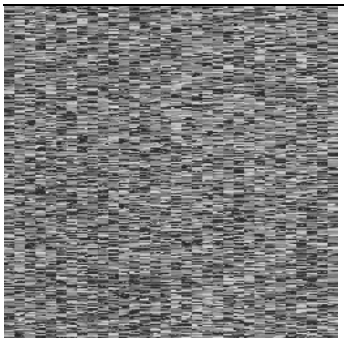
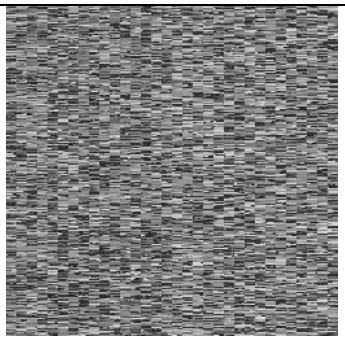
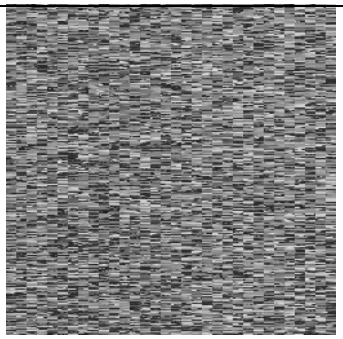
Tabel 8. Citra Hasil Dekripsi dengan Kunci Enkripsi 10 & 7000, Kunci 2 Pada Dekripsi Berbeda-beda

 *Plain image* – lena.bmp 256x256	 *Cipher image* Kunci 1: 10 Kunci 2: 7000 Waktu Proses: 625	 *Decipher image* Kunci 1: 10 Kunci 2: 7000 Waktu Proses: 1047
 Kunci 1: 10 Kunci 2: 4000 Waktu Proses: 656	 Kunci 1: 10 Kunci 2: 4500 Waktu Proses: 750	 Kunci 1: 10 Kunci 2: 5000 Waktu Proses: 828
 Kunci 1: 10 Kunci 2: 7500 Waktu Proses: 1312	 Kunci 1: 10 Kunci 2: 9000 Waktu Proses: 1485	 Kunci 1: 10 Kunci 2: 10000 Waktu Proses: 1593

Eksperimen Dekripsi Terhadap Sebuah Citra Dengan Kunci 1 Yang Berbeda Dari Kunci 1 Pada Proses Enkripsi

Eksperimen yang dilakukan menggunakan kunci 1 (nilai *Random seed*) pada proses dekripsi berbeda dengan kunci 1 pada poses enkripsi sedangkan kunci 2 (jumlah generasi) mempunyai nilai yang sama dengan kunci yang digunakan pada proses enkripsi. Pada proses enkripsi, kunci yang digunakan adalah kunci 1 = 10 dan kunci 2 = 10000, pada eksperimen proses dekripsi, kunci yang digunakan adalah kunci 1 berbeda-beda dan kunci 2 = 10000. Hasil eksperimen yang dilakukan pada citra lena menunjukkan bahwa jika kunci 1 pada proses dekripsi berbeda dengan kunci 1 pada proses enkripsi dan kunci 2 pada proses dekripsi sama dengan kunci 2 pada proses enkripsi, maka secara visual sistem tidak bisa mengembalikan *cipher image* ke *plain image*. Hasil eksperimen ini dapat dilihat dalam Tabel 9.

Tabel 9. Citra Hasil Dekripsi dengan Kunci 1 Berbeda dan Kunci 2 Sama Dengan Kunci Proses Enkripsi

 *Plain image* – lena.bmp 256x256	 *Cipher image* Kunci 1: 10 Kunci 2: 10000 Waktu Proses: 906	 *Decipher image* Kunci 1: 10 Kunci 2: 10000 Waktu Proses: 1515
 Kunci 1: 100 Kunci 2: 10000 Waktu Proses: 1469	 Kunci 1: 10000 Kunci 2: 10000 Waktu Proses: 1453	 Kunci 1: 20000 Kunci 2: 10000 Waktu Proses: 1453

Eksperimen Enkripsi - Dekripsi Terhadap Sebuah Citra Dengan Berbagai Kombinasi Kunci 2

Eksperimen enkripsi dan dekripsi dilakukan terhadap sebuah citra dengan berbagai kombinasi pada kunci 2, untuk melihat pengaruh jumlah generasi terhadap waktu proses enkripsi dan dekripsi. Hasil eksperimen dapat dilihat dalam Tabel 10.

Tabel 10. Hasil Eksperimen Enkripsi – Dekripsi Dan Waktu Proses

Data ke	Kunci		Waktu Proses (ms)	
	Kunci 1 (*Random seed*)	Kunci 2 (Jumlah generasi)	Enkripsi	Dekripsi
1	10	1	31	62
2	10	100	47	63
3	10	1000	125	171
4	10	2000	203	328
5	10	3000	281	468
6	10	4000	375	610
7	10	4500	422	688
8	10	4900	438	750
9	10	5000	453	750
10	10	6500	594	984
11	10	7000	625	1047
12	10	8000	735	1218
13	10	10000	906	1484
14	10	20000	1719	2890

Berdasarkan data dalam Tabel 10., terlihat bahwa semakin besar nilai jumlah generasi, maka semakin lama waktu yang dibutuhkan baik oleh proses enkripsi maupun proses dekripsi. Tabel 10. juga menunjukkan bahwa waktu proses untuk dekripsi lebih lama dari waktu proses enkripsi, hal ini dikarenakan dalam proses dekripsi, terdapat proses pencarian seluruh kromosom dalam populasi generasi terakhir, yang digunakan sebagai panduan untuk menentukan kromosom mana yang mengalami proses mutasi dan *crossover*. Pengaruh kunci, dalam hal ini kunci 2 (jumlah generasi) terhadap waktu proses (*running-time*) untuk enkripsi dan dekripsi digambarkan dalam bentuk grafik seperti terlihat dalam Gambar 4.

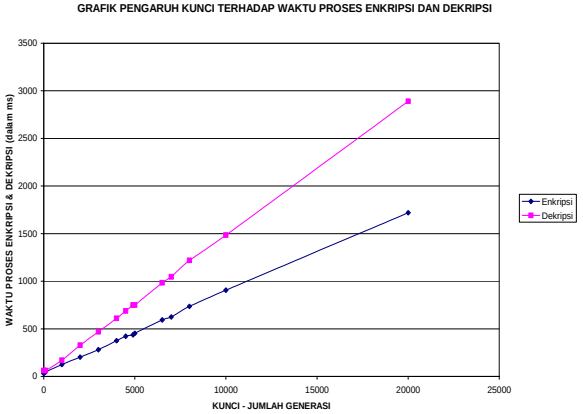
Berdasarkan data grafik yang ditampilkan dalam gambar 3, dapat dikatakan bahwa kunci dalam hal ini jumlah generasi berbanding lurus terhadap waktu proses enkripsi dan dekripsi. Lama tidaknya proses enkripsi dan dekripsi tergantung pada jumlah generasi yang dilakukan.

Analisis Kriptosistem

Berdasarkan hasil eksperimen enkripsi dan dekripsi terhadap citra lena dan peppers, menunjukkan bahwa kriptosistem dengan menggunakan algoritma

genetika dapat mengenkripsi dan mendekripsi data citra dengan baik. *Plain-image* dienkripsi menjadi *cipher image*, secara visual terlihat bahwa *cipher-image* yang dihasilkan tidak sama dengan *plain-image* dan tidak merepresentasikan suatu objek yang dikenali. Demikian juga sebaliknya, kriptosistem dapat mendekripsi *cipher image*, dimana *decipher image* terlihat sama dengan *plain-image*.

Pengukuran yang dilakukan untuk melihat apakah *decipher image* sama dengan *plain image*, dilakukan dengan metode kesamaan citra berbasis piksel (*pixel base similarity*), yaitu membandingkan setiap intensitas nilai piksel yang terdapat pada titik koordinat yang sama pada *decipher image* dengan *plain image*. *Decipher image* dikatakan sama dengan *plain image*, jika setiap nilai piksel pada titik koordinat yang sama pada *decipher image* dengan *plain image* adalah sama. Berdasarkan hasil pengukuran diperoleh hasil bahwa pada titik koordinat yang sama pada *decipher image* dengan *plain image* mempunyai intensitas nilai piksel yang sama.



Gambar 4. Grafik Pengaruh Kunci Terhadap Waktu Proses Enkripsi dan Dekripsi

Analisis Sensitifitas Kunci

Berdasarkan eksperimen yang dilakukan dengan berbagai kombinasi kunci, dapat dikatakan bahwa:

- Kunci 1 pada enkripsi, dalam hal ini nilai *random seed*, tidak mempengaruhi keter-acak-kan *cipher-image*.
- Kunci 2 pada enkripsi, dalam hal ini jumlah generasi, mempengaruhi keter-acak-kan *cipher image* dan waktu yang dibutuhkan untuk proses enkripsi dan dekripsi
- Kunci 1 dalam dekripsi, jika berbeda dengan kunci 1 dalam enkripsi (kunci 2 sama), maka *cipher-image* tidak dapat didekripsikan.
- Kunci 2 dalam dekripsi, jika berbeda dengan kunci 2 dalam enkripsi (kunci 1 sama), maka *cipher-image* dapat didekripsikan, tetapi tidak sempurna, dengan range selisih dengan kunci 2 berada dalam interval berikut:
  $K2 - 3000 \leq n \leq K2 + 3000$
  yang mana :
  K2 = Kunci 2 yang digunakan dalam enkripsi, $K2 \geq 7000$
  n = kunci 2 yang tidak sama dengan kunci 2 pada enkripsi

Analisis Ruang Kunci

Ruang kunci menentukan jumlah *cipher-image* berbeda yang dapat dihasilkan dari sebuah *plain-image* dengan suatu metode. Semakin besar ruang kunci, maka semakin banyak *cipher-image* berbeda yang dihasilkan. Jika semakin banyak *cipher image* yang dihasilkan, hal ini memperkecil kemungkinan terbongkarnya sebuah *cipher-image*.

Ruang kunci pada metode ini ditentukan oleh 2 buah kunci yaitu:

- *Random seed*, bertipe integer 32 bit (*longint-signed bit*), mempunyai kemungkinan nilai dengan interval nilai antara -2147483648 s/d 2147483647, dan jumlah bit –nya adalah 2^{32}.
- Jumlah generasi, bertipe integer 32 bit (*longint-signed bit*), mempunyai kemungkinan nilai dengan interval nilai antara -2147483648 s/d 2147483647. Untuk jumlah generasi, nilai yang valid adalah > 0, sehingga jumlah bit-nya adalah 2^{31}.

Dengan demikian ruang kunci yang dihasilkan untuk metode kriptosistem ini adalah sebesar $2^{32} . 2^{31} = 2^{63}$.

Analisis Waktu Proses (*Running Time*)

Waktu proses (*running time*) dipengaruhi oleh berbagai faktor, salah satunya ditentukan oleh jumlah proses yang dikerjakan. Dalam kriptosistem yang dibuat waktu proses dipengaruhi oleh jumlah generasi, yang mana semakin tinggi nilai jumlah generasi maka waktu yang dibutuhkan dalam proses enkripsi dan dekripsi juga semakin tinggi.

Proses enkripsi dilakukan dengan operasi *crossover* dan mutasi, sedangkan proses dekripsi dilakukan dengan operasi mutasi sebanyak 2 kali dan operasi *crossover*, dengan demikian waktu proses untuk dekripsi lebih lama dibandingkan waktu proses untuk enkripsi, Hal ini juga terlihat dalam hasil eksperimen yang dilakukan yang dinyatakan dalam bentuk grafik.

KESIMPULAN

Berdasarkan perancangan, percobaan, dan analisis yang telah dilakukan, maka dapat diambil kesimpulan, sebagai berikut:

1. Kriptosistem yang dibuat merupakan kriptosistem yang dapat mengenkripsi citra digital berupa citra *grayscale* 8-bit dalam format citra bitmap (*.bmp) menjadi *cipher image* dalam format bitmap (*.bmp) dan dapat didekripsikan kembali menjadi *decipher image* yang sama dengan *plain image*.
2. Jumlah generasi berbanding lurus dengan waktu proses enkripsi dan dekripsi. Semakin besar jumlah generasi semakin lama waktu yang dibutuhkan dalam proses enkripsi dan dekripsi.
3. Dengan jumlah generasi 7000 atau lebih, secara visual kriptosistem sudah menghasilkan *cipher image* yang berbeda dari *plain image*, serta tidak merepresentasikan suatu objek yang dapat dikenali.
4. Ruang kunci yang dihasilkan cukup tinggi yaitu sebesar 2^{63}, sehingga memperkecil kemungkinan terbongkarnya *cipher-image* yang dihasilkan.
5. Waktu yang diperlukan untuk proses dekripsi lebih lama dari waktu untuk proses enkripsi.

DAFTAR PUSTAKA

1. Gen, M. dan Cheng, R. 2000. *Genetic Algorithms and Engineering Optimization*, John Willey and Sons, Inc., Canada.
2. Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison – Wesley, Reading, MA.
3. Gonzalez, R. C. dan Woods, R. E. 1992. *Digital Image Processing*, Addison – Wesley, Publishing Company, Inc.
4. Kurniawan, Y. 2004. *Kriptografi: Keamanan Internet dan Jaringan Komunikasi*, C.V. Informatika, Bandung.
5. Schneier, B. 1996. *Applied Cryptography: protocols, algorithms, and source code in C*, John Wiley & Sons, Inc.
6. Stalling, W. 1999. *Cryptography and Network Security, Principle and Practice 2rd Edition*, Pearson Education, Inc.
7. Stinson, D. R. 1995. *Cryptography: Theory and Practice*, CRC Press LLC, USA.