

PENERAPAN ALGORITMA GENETIKA UNTUK KOMPRESI CITRA FRAKTAL

**Putu Indah Ciptayani, Wayan Firdaus Mahmudy,
Agus Wahyu Widodo**
Program Studi Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Brawijaya

ABSTRAK

Kompresi citra fraktal merupakan salah satu teknik kompresi yang mampu menghasilkan rasio kompresi yang tinggi dengan kualitas citra hasil kompresi yang baik. Namun memiliki kelemahan lamanya waktu kompresi karena pencocokan antara blok jelajah dan blok ranah yang dilakukan secara brute force. Karena itu dilakukan pendekatan dengan algoritma genetika, di mana algoritma genetika merupakan suatu pendekatan yang tepat untuk permasalahan kombinatorik yang kompleks. Peran algoritma genetika adalah dalam mencocokkan blok jelajah dan blok ranah. Uji coba dilaksanakan dengan menggunakan tiga metode crossover dan tiga metode mutasi, ukuran blok jelajah 4, probabilitas mutasi 0.1, probabilitas crossover 0.5, 0.6, 0.7, 0.8, 0.9 dan 1.0. Maksimal generasi adalah 500. Citra hasil kompresi terbaik menghasilkan rasio 75.01% dengan waktu kompresi 10.7 detik dan MSE sebesar 0.158839.

ABSTRACT

Fractal image compression is one of compression techniques which produce a high compression ratio with good quality of result image. But this method has weakness is the time to compress image is too long because checking domain and range block is done by brute force method. Because of it, necessary to get approach with genetic algorithm which genetic algorithm is an appropriate approach for complex combinatorial problem. Genetic algorithm play role in searching the matching domain and range block. Experiment is done by use three crossover and mutation method, the size of range block is 4, mutation probability is 0.1, crossover probability is 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0. Maximal size of generations are 500. The best result of compression image has ratio 75.01% with compression time is 10.7 second and MSE is 0.158839.

1. Pendahuluan

Dewasa ini ada banyak metode kompresi citra, salah satunya adalah kompresi citra fraktal. Teknik kompresi citra fraktal diperkenalkan oleh Michael Barnsley dan Jacquin, di mana teknik ini memanfaatkan konsep citra fraktal yaitu suatu citra yang

memiliki kesamaan terhadap dirinya sendiri dalam skala yang berbeda. Karena tingginya rasio kompresi dan teknik dekompresi yang sederhana, maka riset mengenai kompresi citra fraktal banyak dilakukan. Hingga saat ini permasalahan yang masih dihadapi adalah waktu kompresi yang lama. Hal ini disebabkan karena waktu pencocokan antara blok jelajah dan blok ranah. Karena itu dilakukan pendekatan dengan algoritma genetika.

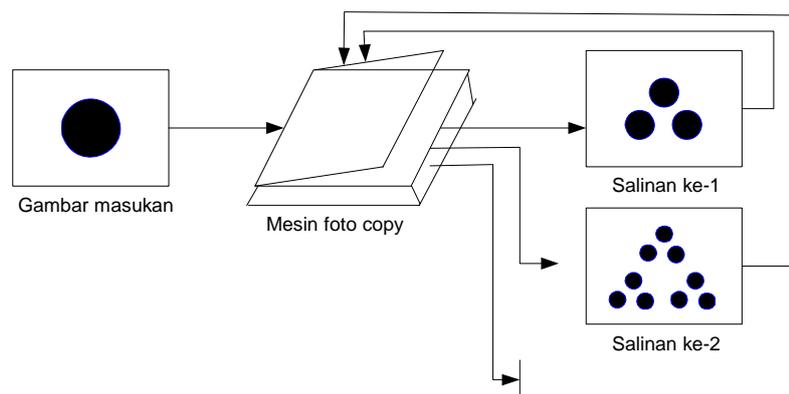
Algoritma genetika merupakan algoritma pencarian heuristik yang sangat cocok sebagai pendekatan untuk permasalahan kombinatorik yang sangat kompleks seperti pencarian blok ranah yang cocok dengan blok jelajah. Algoritma genetika membangkitkan koordinat ujung kiri atas blok ranah secara acak, kemudian *fitness* diperoleh dengan menghitung kemiripan antara blok ranah dengan blok jelajah tertentu. Generasi baru dibentuk dengan *crossover* dan mutasi. Transformasi terbaik diambil dari individu dengan *fitness* terbaik.

Landasan teori akan dibahas pada bagian 2, metodologi kompresi citra fraktal dengan algoritma genetika akan dibahas pada bagian 3, hasil uji coba dan pembahasan akan dibahas pada bagian 4, sedangkan kesimpulan akan dibahas pada bagian 5.

2. Landasan Teori

2.1 Iterated Function System (IFS)

Michael Barnsley merepresentasikan fraktal ke dalam model matematika yang dinamakan IFS (*Iterated Function System*) [Mun04]. IFS digambarkan sebagai sebuah mesin *foto copy* yang memiliki banyak lensa dan disebut dengan *Multiple Reduction Copy Machine* (MRCM). Setiap lensa dalam MRCM melakukan pengecilan gambar dalam jumlah yang banyak. Setiap lensa dalam MRCM melakukan pengecilan gambar dalam jumlah yang banyak. Gambar yang dihasilkan dari mesin *foto copy* dioperasikan kembali sebagai masukan untuk membuat salinan berikutnya. Ilustrasinya ditunjukkan oleh Gambar 1.



Gambar 1. Mesin foto copy MRCM

Sistem lensa pada MRCM dapat dinyatakan dengan sekumpulan transformasi *affine* w_1, w_2, \dots, w_n . Setiap transformasi w_i melakukan pencondongan, pemutaran, pengecilan dan pergeseran terhadap salinan (*copy*) citra masukan.

Setiap transformasi *affine*[Mun04] dinyatakan sebagai matriks dengan enam buah elemen :

$$w = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \quad (1)$$

Sembarang titik (x,y) pada gambar masukan[Mun04] ditransformasikan oleh w menjadi

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = w \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = Ax + t \quad (2)$$

Setiap transformasi *affine* w_i akan menghasilkan salinan citra yang lebih kecil. Untuk sembarang citra

$$W = w_1 \text{Y} w_2 \text{Y} \dots \text{Y} w_n = \text{Y}_{i=1}^n w_i \quad (3)$$

2.2 Partitioned Iterated Function System (PIFS)

Teknik kompresi yang dipakai oleh kompresi fraktal yaitu dengan menyimpan transformasi *affine*-nya. Walaupun citra alami tidak pernah *self similar* secara keseluruhan, citra alami memiliki kemiripan lokal, yaitu terdapat bagian citra yang mirip dengan bagian lainnya dalam skala yang berbeda. Kemiripan lokal yang banyak terdapat pada citra alami bersifat *self-transformability*, yaitu bagian citra yang lebih kecil dapat diperoleh dengan mentransformasikan bagian citra yang lebih besar namun mirip dengan bagian citra yang lebih kecil itu [Mun04].

Kompresi citra fraktal dilakukan dengan membagi sebuah citra menjadi sejumlah blok dengan ukuran tertentu yang sama dan tidak saling beririsan, disebut sebagai blok jelajah (*range block*). Kemudian untuk citra yang sama, akan dibagi menjadi sejumlah blok dengan ukuran lebih besar daripada blok jelajah dan saling beririsan yang disebut sebagai blok ranah (*domain block*). Setelah menemukan blok jelajah dengan blok ranah yang cocok, kemudian diturunkan transformasi *affine* (IFS lokal) w_i yang memetakan blok ranah ke blok jelajah. Hasil dari semua pemasangan ini adalah *Partitioned Iterated Function System* (PIFS) [Mun04].

Titik (x,y) dengan intensitas z yang termasuk di dalam blok ranah dipetakan oleh w_i menjadi :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (3)$$

Dengan pemetaan w_i di atas, intensitas tiap *pixel* juga diskalakan dan digeser, yaitu

$$z' = s_i z + o_i \quad (4)$$

Parameter e_i dan f_i menyatakan pergeseran sudut kiri blok ranah ke sudut kiri blok jelajah yang bersesuaian. Parameter s_i menyatakan faktor kontras *pixel*, sedangkan o_i menyatakan ofset kecerahan (*brightness*) *pixel*. Karena perbandingan ukuran antara

blok jelajah dengan blok ranah adalah 1:2, maka transformasi *affine* pada persamaan 2.8 akan menjadi lebih sederhana, yaitu

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (5)$$

Parameter s_i dan o_i dihitung dengan menggunakan rumus regresi berikut : Diberikan dua buah blok citra yang mengandung n buah *pixel* dengan intensitas d_1, d_2, \dots, d_n (dari blok ranah D_i) dan r_1, r_2, \dots, r_n (dari R_i). Nilai s dan o diperoleh dengan meminimumkan total kuadrat selisih antara intensitas *pixel* blok D_i yang diskalakan dengan s lalu digeser sejauh o dan intensitas *pixel* blok R_i .

$$s = \frac{\left[n \sum_{i=1}^n d_i r_i - \sum_{i=1}^n d_i \sum_{i=1}^n r_i \right]}{\left[n \sum_{i=1}^n d_i^2 - \left(\sum_{i=1}^n d_i \right)^2 \right]} \quad (6)$$

$$o = \frac{1}{n} \left[\sum_{i=1}^n r_i - s \sum_{i=1}^n d_i \right] \quad (7)$$

Jika $n \sum_{i=1}^n d_i^2 - \left(\sum_{i=1}^n d_i \right)^2 = 0$ maka $s = 0$ dan $o = \frac{1}{n} \sum_{i=1}^n r_i$. Dengan nilai s dan o

yang telah diperoleh, maka kuadrat galat antara blok jelajah dan blok ranah adalah

$$E = \frac{1}{n} \left[\sum_{i=1}^n r_i^2 + s \left(s \sum_{i=1}^n d_i^2 - 2 \sum_{i=1}^n d_i r_i + 2o \sum_{i=1}^n d_i \right) + o \left(no - 2 \sum_{i=1}^n r_i \right) \right] \quad (8)$$

$$Drms = \frac{\sqrt{E}}{n} \quad (9)$$

2.3 Rekonstruksi Citra

Rekonstruksi (*decoding*) citra dilakukan dengan menguraikan PIFS dari citra awal sembarang. Ketika proses dekompresi, setiap IFS lokal mentransformasikan sekumpulan blok ranah menjadi sekumpulan blok jelajah.

2.4 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis [Kus05]. Algoritma ini dimulai dengan sejumlah solusi yang mungkin. Sejumlah solusi yang mungkin ini disebut sebagai populasi. Masing-masing solusi yang mungkin di dalam sebuah populasi disebut dengan kromosom (*Chromosome*). *Chromosome* tersusun atas gen-gen yang merupakan representasi nilai variabel dalam suatu fungsi yang ingin dicari solusinya. Setiap kromosom diberi sebuah

nilai *fitness* berdasarkan pada fungsi *fitness*. Solusi dari sebuah populasi diambil dan dipergunakan untuk membentuk populasi yang baru dengan *crossover* dan mutasi. Hasil yang diharapkan adalah populasi baru yang terbentuk akan memiliki nilai *fitness* yang lebih tinggi dibandingkan dengan populasi sebelumnya.

3. Metodologi

3.1 Representasi Kromosom

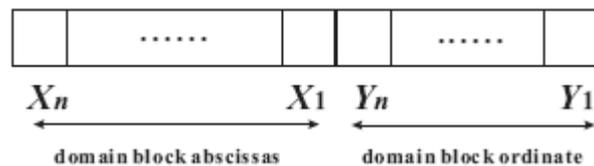
Representasi kromosom untuk kompresi citra fraktal ini terdiri dari 2 bagian yaitu bagian absis (*e*) dan bagian ordinat (*f*) ujung kiri atas blok ranah yang dipetakan ke blok jelajah tertentu. Kromosom dipecah menjadi dua bagian yaitu bagian absis sepanjang jumlah blok jelajah, kemudian diikuti oleh bagian ordinat sepanjang blok jelajah. Ilustrasinya ditunjukkan oleh Gambar 2.

3.2 Perhitungan *Fitness*

Perhitungan nilai *fitness* ini didasarkan pada sebuah fungsi *fitness* yaitu ukuran kedekatan antara blok jelajah dengan blok ranah yang bersesuaian. Untuk menghitung ukuran kedekatan digunakan metrik *rms*, *drms* pada persamaan 9.

3.3 Seleksi

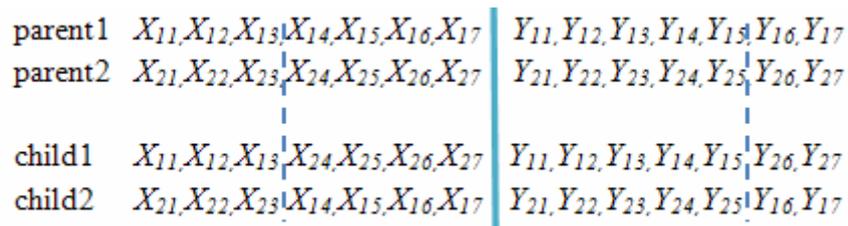
Metode seleksi yang digunakan adalah metode *roulette wheel*. Dalam metode ini, setiap kromosom akan menempati area sesuai dengan besarnya nilai *fitness*. Kemudian akan diacak angka, kromosom yang terpilih adalah kromosom yang areanya terdapat angka yang muncul.



Gambar 2. Representasi kromosom

3.4 Perkawinan Silang

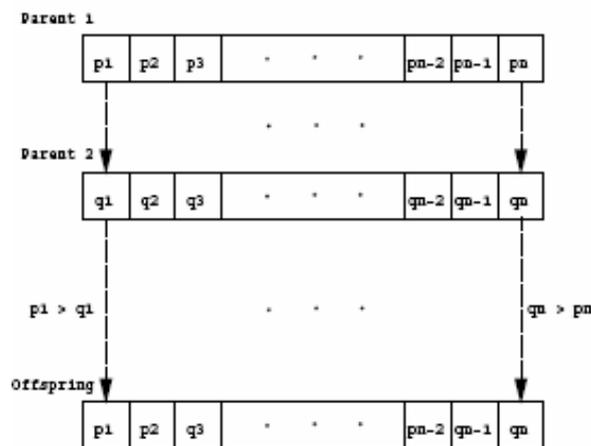
Metode perkawinan silang yang digunakan ada tiga macam yaitu satu titik potong biasa (Gambar 3), satu titik potong dengan pengambilan absis dan ordinat dibalik (Gambar 4) dan perkawinan silang dengan memperhatikan *fitness* induk (Gambar 5).



Gambar 3. Metode *crossover* satu titik potong biasa

parent1	$X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}$	$Y_{11}, Y_{12}, Y_{13}, Y_{14}, Y_{15}, Y_{16}, Y_{17}$
parent2	$X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{27}$	$Y_{21}, Y_{22}, Y_{23}, Y_{24}, Y_{25}, Y_{26}, Y_{27}$
child1	$X_{24}, X_{25}, X_{26}, X_{27}, X_{11}, X_{12}, X_{13}$	$Y_{26}, Y_{27}, Y_{11}, Y_{12}, Y_{13}, Y_{14}, Y_{15}$
child2	$X_{14}, X_{15}, X_{16}, X_{17}, X_{21}, X_{22}, X_{23}$	$Y_{16}, Y_{17}, Y_{21}, Y_{22}, Y_{23}, Y_{24}, Y_{25}$

Gambar 4. Metode *crossover* satu titik potong dengan pengambilan absis dan ordinat dibalik.



Gambar 5. Metode *crossover* dengan memperhatikan *fitness* induk.

Pada metode *crossover* dengan memperhatikan *fitness* induk, *fitness* masing-masing gen induk dibandingkan nilai *fitness*-nya. Gen dengan nilai *fitness* terbesar akan menjadi gen pembentuk *offspring*.

3.5 Mutasi

Mutasi dilakukan dengan tiga metode yaitu mutasi pertukaran (Gambar 6), penggantian (Gambar 7), dan pencarian optimum local dari gen termutasi (Gambar 8).

sebelum mutasi $X_{11}X_{12}X_{13}X_{14}X_{15}X_{16}X_{17}$ $Y_{11}Y_{12}Y_{13}Y_{14}Y_{15}Y_{16}Y_{17}$
 setelah mutasi $X_{11}X_{12}X_{15}X_{14}X_{13}X_{16}X_{17}$ $Y_{11}Y_{16}Y_{13}Y_{14}Y_{15}Y_{12}Y_{17}$

Gambar 6. Mutasi pertukaran (*swap*)

sebelum mutasi $X_{11}X_{12}X_{13}X_{14}X_{15}X_{16}X_{17}$ $Y_{11}Y_{12}Y_{13}Y_{14}Y_{15}Y_{16}Y_{17}$
 setelah mutasi $X_{11}X_{12}X_{15}X_{14}X_{13}X_{16}X_{17}$ $Y_{11}Y_{16}Y_{13}Y_{14}Y_{15}Y_{13}Y_{17}$

Gambar 7. Mutasi penggantian

	1	2	3	4	5	6
0		T2				
1	T1	M	T3			
2		T4				
3						
4						
5						
6						

Gambar 8. Mutasi pencarian optimum lokal

Pada Gambar 8, gen yang termutasi adalah gen 1,1 (area berwarna merah). Kemudian diperiksa apakah tetangga dari gen tersebut yaitu 0.1 (T1), 1.0 (T2), 2.1 (T3) dan 1.2(T4). Jika *fitness* tertinggi dari keempat tetangga tersebut lebih tinggi daripada *fitness* gen yang termutasi, maka tetangga dengan *fitness* terbaik itulah yang akan menggantikan gen tersebut.

4. Hasil Uji Coba dan Pembahasan

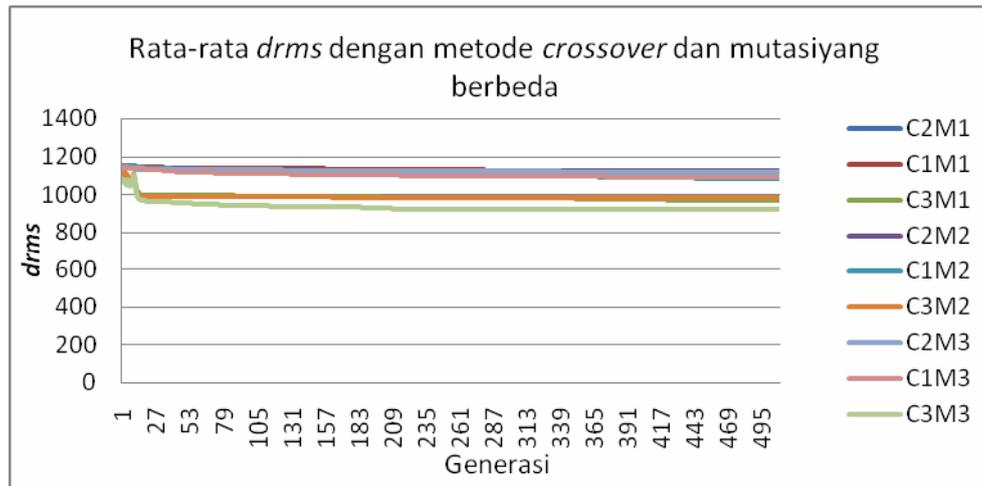
Uji coba dilakukan dengan menggunakan AMD Sempron 1.5 GHz dan RAM 384 MB dengan sistem operasi Windows XP Professional dan dengan menggunakan *compiler* Delphi 7. Citra yang dipakai adalah citra *grayscale* Lena.bmp dengan ukuran 117 x 149 *pixel*, ukuran blok jelajah 4, probabilitas mutasi 0.1 dan probabilitas *crossover* 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. Jumlah individu adalah sebanyak 5 dan generasi maksimal 500. Citra Lena.bmp dapat dilihat pada Gambar 9a, sedangkan citra Lena.dwi hasil kompresi dengan metode *crossover* yang memperhatikan *fitness* induk dan mutasi pencarian optimum lokal dapat dilihat pada Gambar 9b.



Gambar 9a

Gambar 9b

Rasio kompresi yang dihasilkan yaitu 75.01% dengan MSE sebesar 0.161584. Grafik kinerja perpaduan metode *crossover* dan mutasi ditunjukkan oleh Gambar 10



Gambar 10. Grafik kinerja perpaduan metode *crossover* dan mutasi

5. Kesimpulan

Untuk mengatasi lamanya waktu kompresi citra fraktal, maka dilakukan pendekatan dengan algoritma genetika. Algoritma genetika berperan dalam pencocokan blok jelajah dengan blok ranah. Metode *crossover* yang dipakai sebanyak tiga, sedangkan metode mutasi yang dipakai ada tiga. Dari Sembilan perpaduan metode, metode yang menghasilkan hasil terbaik adalah metode *crossover* memperhatikan *fitness* induk dan mutasi pencarian optimum lokal. Waktu yang diperlukan adalah selama 10.7 detik dengan rasio 4:1 dan MSE sebesar 0.161584.

Daftar Pustaka

- [Anonym97] Anonymous. 1997. *An Introduction to Fractal Image Compression*. Texas Instruments Europe
- [Her06] Hermawanto, Denny. 2006. *Representasi Kromosom Algoritma Genetika dalam Bentuk Biner*, <http://dennyhermawanto.webhop.org>. Tanggal akses : 24 Mei 2008.
- [Kus07] Kuswadi, Son. 2007. *Kendali Cerdas, Teori dan Praktisnya*. Andi : Yogyakarta
- [Kusu05] Kusumadewi, Sri & Purnomo, Hari. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Graha Ilmu : Yogyakarta
- [Mun04] Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Informatika: Bandung.
- [Nur06]] Nurjaya, Wahyu. 2006. *Analisis Proses Word Matching Problem Menggunakan Algoritma Genetika*. Universitas Komputer Indonesia.

- [Obit98] Obitko, M. 1998. *Genetic Algorithm*, <http://cs.felk.cvut.cz/~xobitko/ga/>. Tanggal akses: 24 Mei 2008.
- [Obit98] Obitko, M. 1998. *Introduction to Genetic Algorithm*, <http://cs.felk.cvut.cz/~xobitko/ga/>. Tanggal akses: 24 Mei 2008.
- [Ven97] Vences, Lucia, Rudomin, Isaac. 1997. *Genetic Algorithms for Fractal Image and Image Sequence Compression*. Instituto Tecnológico de Estudios Superiores de Monterrey, Campus Estado de Mexico: Mexico.
- [Xi07] Xi, Lifeng, Zhang, Liangbin. 2007. *A Study of Fractal Image Compression Based on an Improved Genetic Algorithm*. Institute of Mathematics, Zhejiang Wanli University.