

# IMPLEMENTASI ALGORITMA FRAKTAL UNTUK KOMPRESI CITRA DENGAN METODE PENCARIAN LOKAL

Jatmika<sup>1</sup>, Tresia F Randongkir<sup>2</sup>

## Abstraksi

*The nature of image compression methods always fall between lossy and lossless compression. Lossy compression eliminates insignificant information while retaining the perception, while lossless compression retains the original data completely. These recent years saw the rise of Fractal Image Compression (FIC), a new lossy image compression algorithm. This algorithm features a self-similarity, which in other word it regards an image as an arrangement of copied parts of the image itself, thus we only need a composition of transformation to code an image.*

*This paper discuss about how fractal algorithm can be applied for image compression, how Fractal Image Compression works, and how to implement it using local search where comparison is done to the nearest area (segments) only.*

*Searching in progress often involves great amount of data which takes a considerable time. Local search can reduce the time by comparing only the nearest area within the neighbourhood of the current block, which in turn shortened the overall processing time. However, the sharply reduced processing time achieved by localizing the search does not drastically reduce the quality of the output time.*

**Kata Kunci** : algoritma FIC, Iterated Function System (IFS). Citra, Kompresi

## 1. PENDAHULUAN

Pada jaman ini aplikasi multimedia sangat berkembang, hampir sebagian besar menggunakan file-file citra yang berukuran sangat besar. Metode kompresi citra yang sudah kita kenal antara lain *JPEG (Joint Photographic Experts Group)*. Namun mulai dikembangkan metode *FIC (Fractal Image Compression)* yaitu kompresi citra menggunakan metode fraktal. *FIC* ini bersifat *lossy* yaitu menghilangkan beberapa informasi citra yang tidak berpengaruh besar terhadap penglihatan manusia, ada juga sifat kompresi citra yang lain yaitu *lossless* yang tidak menghilangkan informasi citra sehingga setelah proses dekompresi menghasilkan citra yang sama dengan aslinya. Pada umumnya kompresi bersifat *lossy* dapat menghasilkan rasio kompresi lebih besar dari pada yang bersifat *lossless*. Meskipun masih dalam tahap pengembangan namun *FIC (Fractal Image Compression)* memiliki potensi rasio kompresi yang relatif besar.

## 2. LANDASAN TEORI

### 2.1. Citra

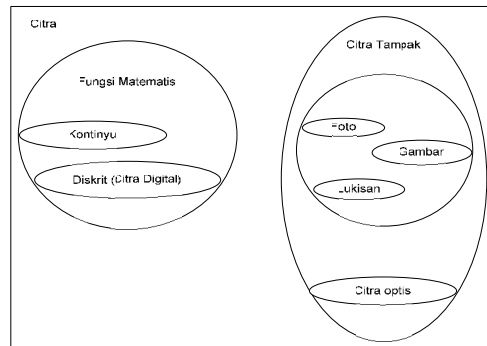
Definisi citra menurut kamus Webster adalah “ suatu representasi, kemiripan, atau imitasi dari objek”. Foto seseorang mewakili keadaan bagian tubuh seorang dan data dalam suatu berkascitra mewakili apa yang digambarkannya. Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak (lihat Gambar 1.). Banyak contoh citra tampak dalam kehidupan sehari-hari: foto teman-teman mahasiswa, foto pemandangan, apa yang tampak dilayar monitor dan televisi, serta hologram (citra optis). Sedangkan citra tak tampak, misalnya citra distribusi panas dikulit manusia, serta pada densitas dalam suatu material. Untuk dapat dilihat mata manusia, citra tak tampak ini harus diubah menjadi citra tampak misalnya: dengan menampilkannya di layar monitor, dicetak di atas kertas, dan sebagainya.

---

<sup>1</sup>FMIPA UKRIM Yogyakarta

<sup>2</sup>FMIPA UKRIM Yogyakarta

Diantara jenis-jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain, jika hendak diolah dengan komputer, harus diubah dulu menjadi citra digital, misalnya: foto dipindah (*scan*) dengan scanner, persebaran panas tubuh manusia ditangkap dengan kamera inframerah dan diubah menjadi informasi numeris, informasi densitas dan komposisi bagian dalam tubuh manusia ditangkap dengan bantuan pesawat sinar-X, serta sistem deteksi radius menjadi informasi digital disebut sebagai pencitraan (*imaging*).



Gambar 1. Pengelompokan jenis-jenis citra (Castleman, 1996)

### 2.2. Transformasi Affine pada Bidang Datar

Transformasi affine dapat dideskripsikan sebagai kombinasi dari rotasi, penyekalan dan translasi dari titik koordinat dalam ruang n dimensi. Transformasi affine pada bidang 2 dimensi  $W : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  dapat dinyatakan dalam bentuk

$$W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad [1]$$

Dimensi a,b,c,d,e, dan f adalah bilangan riil sebagai contoh transformasi.

### 2.3. Transformasi Bersifat Kontraktif

Suatu transformasi  $w$  dikatakan kontraktif jika untuk sembarang dua titik  $P_1$  dan  $P_2$  terdapat suatu konstanta  $0 \leq S < 1$  sedemikian hingga

$$d(w(P_1), w(P_2)) \leq S d(P_1, P_2) \quad [2]$$

Jarak  $d$  yang akan dibahas disini adalah jarak pada bidang datar dan dirumuskan

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### 2.4. Iterated Function System (IFS)

Ruang lingkup fraktal yang menjadi dasar dalam kompresi fraktal suatu citra adalah *Iterated Function System* (IFS). Dalam Barnsky dan Sloan disebutkan bahwa IFS adalah koleksi dari transformasi affine yang kontraktif [Barnsky dan Sloan, 1988]. Suatu IFS dinyatakan dengan  $W$  dapat dirumuskan :

$$W(f) = \bigcup_{i=1}^n W_i(f) \quad [3]$$

Dimana  $W_i$  merupakan transformasi affine yang kontraktif. IFS direpresentasikan dalam bentuk kode-kode yang disebut kode IFS. Kode inilah yang nantinya dapat digunakan untuk mendekati suatu citra. Table 1 dan 2. merupakan salah satu contoh kode IFS untuk segitiga sierfinske (Gambar 1) dan daun pakis (Gambar 1). Probabilitas  $P$  pada transformasi tersebut menandakan prioritas relative terhadap transformasi yang lain. Jumlah probabilitas dari masing-masing transformasi harus yang diinginkan menggunakan algoritma iterasi random

Table 1.  
Kode IFS untuk segitiga Sierpinski

W	a	b	c	d	e	f	P
1	0.5	0	0	0.5	0	0	0.33
2	0.5	0	0	0.5	1	0	0.33
3	0.5	0	0	0.5	0.5	0.5	0.34

Tabel 2.  
Kode IFS untuk daun Pakis

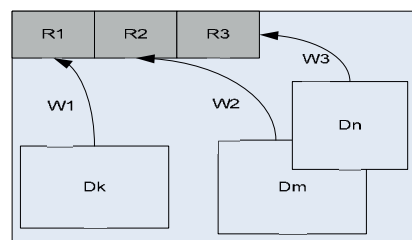
W	a	b	c	d	e	f	P
1	0	0	0	0.16	0	0	0.01
2	0.2	-0.25	0.23	0.22	0	1.6	0.07
3	-0.15	0.28	0.26	0.24	0	0.44	0.07
4	0.85	0.04	-0.04	0.85	0	1.6	0.85

Proses untuk membentuk citra dari kode IFS yang ada dapat dilakukan dengan algoritma iterasi random. Langkah-langkah pembentukan IFS:

1. Inisialisasi  $x = 0, y = 0$
2. Untuk  $n = 1$  sampai iterasi yang diinginkan diselesaikan, lakukan langkah 3 sampai 7
3. Pilih  $k$  dari salah satu bilangan antara  $1, 2, \dots, m$  dengan probabilitas  $pk$
4. Aplikasikan transformasi  $wk$  pada titik  $(x,y)$  untuk mendapatkan  $(x',y')$
5. Set  $(x,y)$  sama dengan titik baru:  $x = x'$  dan  $y = y'$
6. Jika  $n > 10$  plot  $(x,y)$
7. Loop.

### 2.5. Partitioned Iterated Function System (PIFS)

Memodifikasi dari skema IFS agar lebih mudah untuk mengkodekan suatu citra dikemukakan oleh Arnand Jacquin. Skema tersebut dinamakan PIFS. Jika dalam skema IFS citra dipandang sebagai suatu susunan dari hasil transformasi dari citra itu sendiri maka dalam PIFS citra dipandang sebagai suatu susunan dari hasil-hasil transformasi bagian-bagian dari citra itu sendiri. Seperti halnya pada IFS maka proses ini akan di ulang – ulang dan menjadikan hasil output sebagai input pada proses berikutnya untuk mendapatkan hasil yang diinginkan.



Gambar 2. Transformasi domain ke range yang sesuai

### 2.6. Transformasi pada Citra Skala Keabuan

Untuk dapat diaplikasikan dalam citra skala keabuan maka transformasi pada (1) harus ditambahkan dengan satu dimensi lagi yaitu nilai skala keabuan yang dinotasikan sebagai  $Z$  dan  $Z = f(x,y)$ , transformasi (1) disesuaikan menjadi:

$$W \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ai & bi & 0 \\ ci & di & 0 \\ 0 & 0 & si \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} ei \\ fi \\ oi \end{bmatrix}$$

Dimana :

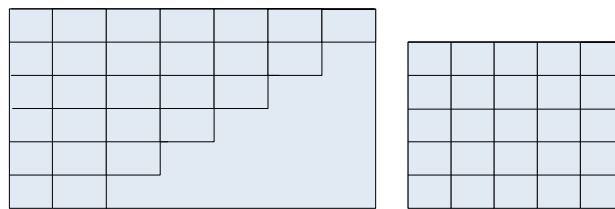
- $si$  = koefisien yang mengatur kecerahan
- $A, b, c$  dan  $d$  = mengatur transformasi posisi
- $S$  dan  $o$  = mengatur transformasi kontras dan kecerahan
- Parameter  $ei$  dan  $fi$  dapat dihitung karena keduanya menyatakan pergeseran sudut kiri blok

## 2.7. Pengkodean Citra

Pengkodean dapat dijelaskan sebagai berikut : andaikan kita mempunyai suatu citra  $f$  yang akan dikodekan, berarti kita ingin mencari koleksi transformasi  $W_1, W_2, \dots, W_n$  dengan  $W = \bigcup_{i=1}^n W_i(f)$  dan  $f = |W|$ , jika kita ingin  $f$  menjadi titik tetap atau attractor dari  $W$ .

Jadi  $f = W(f) = W_1(f) \cup W_2(f) \cup \dots \cup W_n(f)$

Sebagai ilustrasi pengkodean citra kita mengambil contoh sebuah citra berukuran  $256 * 256$  piksel yang setiap piksel mempunyai 256 tingkat keabuan dan hitam sampai putih. Misal  $R_1, R_2, \dots, R_{1024}$  adalah citra yang berukuran  $8*8$  piksel dan tidak saling tumpang tindih dan  $D$  adalah citra yang berukuran  $16 * 16$  piksel dan saling tumpang tindih.



Gambar 3. Himpunan domain dan partisi range

## 2.8. Pendekodean Citra

Pada proses pengkodean citra di atas kita mendapat koleksi transformasi  $W_1, W_2, \dots, W_{1024}$  dan hasil tersebut kita dapat melakukan proses pendekodean citra dengan memperkecil  $|W|$ . Pendekodean dapat dimulai dari sembarang citra awal  $f_0$  selanjutnya kita mengaplikasikan  $W$  yang merupakan gabungan dari  $W_1, W_2, \dots, W_{1024}$  pada citra awal.

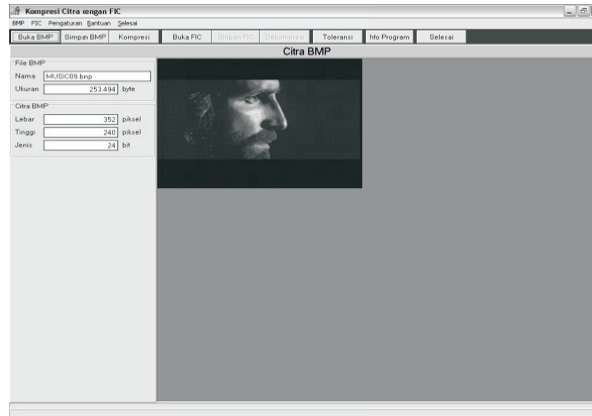
Konvergen disini ditandai adanya kenyataan bahwa pada suatu iterasi tertentu perbedaan antara citra asli dengan citra yang didapatkan dari proses iterasi tersebut tidak mengalami perubahan yang berarti sehingga walaupun iterasi ditambahkan terus akan didapat yang sama

## 2.9. Kompresi pada Citra Berwarna

Untuk citra berwarna, bisa dianggap sebagai perluasan atau pengembangan dari algoritma tersebut [Saupe et.al., 1996]. Citra berwarna terlebih dahulu dipisah menjadi komponen-komponen warna yang nantinya algoritma kompresi diaplikasikan pada tiap-tiap komponen warna tersebut. Pemisahan komponen warna yang dipakai sebaiknya berdasarkan persepsi mata manusia pada warna yaitu berdasarkan kecerahan dan krominansi misalnya komponen YIQ atau YUV dimana komponen Y merupakan komponen luminasi (kecerahan) dan IQ maupun UV adalah komponen krominansi (berhubungan dengan warna).

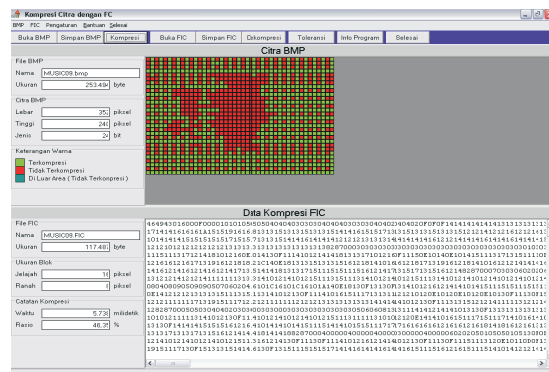
### 3. Implementasi

Pada proses kompresi, sistem menerima masukan dari pemakai berupa sebuah berkas citra digital dengan format BMP (Bitmap). Dari berkas citra BMP yang dibaca oleh sistem, akan ditampilkan kepada pemakai melalui antarmuka yang telah disediakan (Gambar 4).



Gambar 4. Buka File BMP

Setiap citra digital hasil dari kompresi akan tersimpan dengan ekstensi .FIC yang telah ditentukan oleh sistem. Gambar 5 memperlihatkan sebuah contoh hasil kompresi yang ditampilkan pada antarmuka aplikasi. Serupa dengan proses kompresi, sistem juga menyediakan fasilitas untuk dekomposisi terhadap berkas citra hasil kompresi dengan ekstensi FIC.

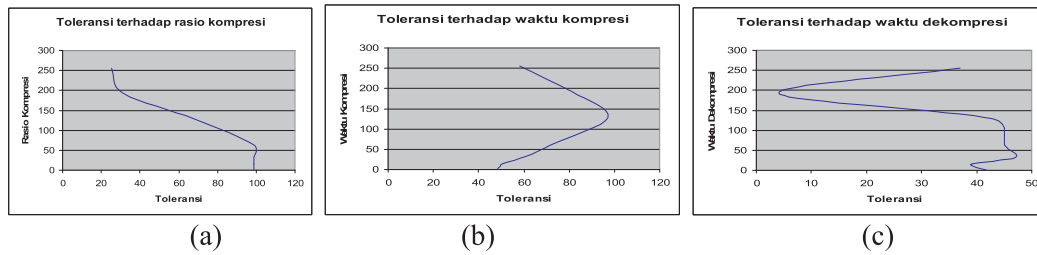


Gambar 5. Hasil Kompresi dan Data kompresi

### 4. Evaluasi Sistem

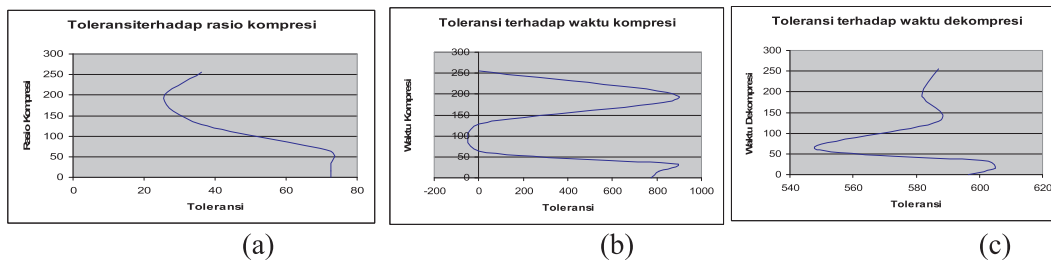
Terhadap sistem yang telah dikembangkan, dilakukan beberapa percobaan untuk melihat toleransi rasio kompresi/dekompresi dan waktu kompresi/dekompresi. Dari masing-masing percobaan, disiapkan empat macam berkas citra digital dengan ukuran resolusi pixel yang berbeda, yaitu 32x32, 128x128, 256x256, 512x512, dan 1024x1024.

Di bawah ini adalah grafik toleransi terhadap citra berukuran 32x32 pixel:



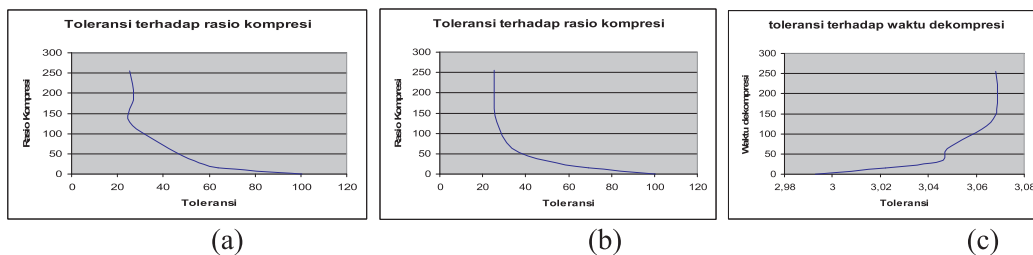
Gambar 6. (a) Toleransi terhadap Rasio Kompresi citra 32x32 pixel, (b) Toleransi terhadap waktu kompresi citra 32x32 pixel, (c) Toleransi terhadap waktu dekompresi citra 32x32 pixel.

Di bawah ini adalah grafik toleransi terhadap citra berukuran 128 x 128 pixel:



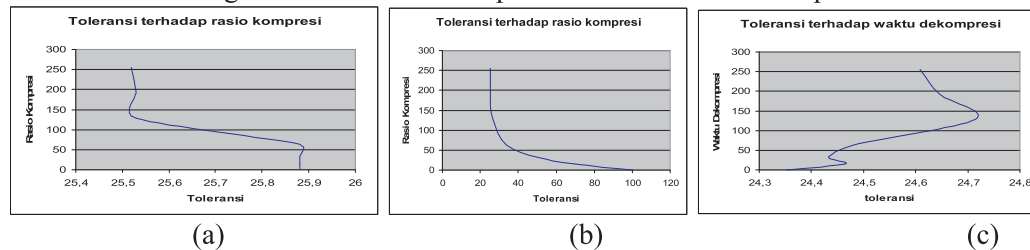
Gambar 7. (a) Toleransi terhadap Rasio Kompresi citra 128x128 pixel, (b) Toleransi terhadap waktu kompresi citra 128x128 pixel, (c) Toleransi terhadap waktu dekompresi citra 128x128 pixel

Di bawah ini adalah grafik toleransi terhadap citra berukuran 256x256 pixel:



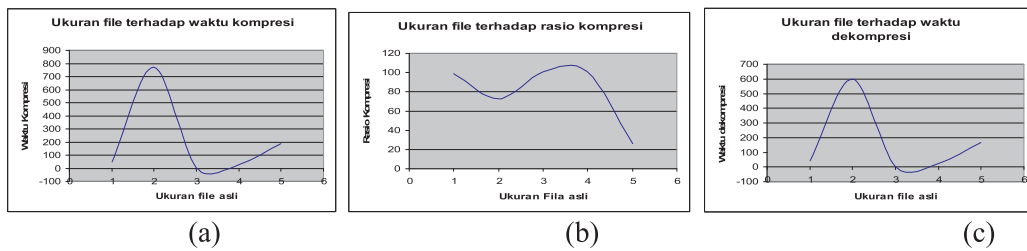
Gambar 8. (a) Toleransi terhadap Rasio Kompresi citra 256x256 pixel, (b) Toleransi terhadap waktu kompresi citra 256x256 pixel, (c) Toleransi terhadap waktu dekompresi citra 256x256 pixel

Di bawah ini adalah grafik toleransi terhadap citra berukuran 512x512 pixel:



Gambar 8. (a) Toleransi terhadap Rasio Kompresi citra 512x512 pixel, (b) Toleransi terhadap waktu kompresi citra 512x512 pixel, (c) Toleransi terhadap waktu dekompresi citra 512x512 pixel

Di bawah ini adalah grafik toleransi terhadap citra berukuran 1024x1024 pixel:



Gambar 9. (a) Toleransi terhadap Rasio Kompresi citra 1024x1024 pixel, (b) Toleransi terhadap waktu kompresi citra 1024x1024 pixel, (c) Toleransi terhadap waktu dekompresi citra 1024x1024 pixel

Dari hasil perbandingan Tabel 3. maka dibuat tiga buah grafik sesuai dengan toleransinya, yang mana menyatakan: ukuran file terhadap rasio kompresi, ukuran file terhadap waktu kompresi dan dekompresi.

Tabel 3.  
Hasil perbandingan kompresi dan dekompresi

No	Ukuran (pixel)	U.File bmp (byte)	Jenis (bit)	Toleransi	Rasio Kompresi (%)	Waktu Kompresi (milidetik)
1	32 x 32	3.126	24	0	99.01	48.1
				16	99.01	51.1
				32	99.01	61
				64	99.01	73
				128	68.30	97
				192	31.45	81.2
				255	25.30	58
2	128 x	49.206	24	0	72.72	773.2
				16	72.72	809.2
				32	72.72	889.5
				64	72.72	1.1047
				128	36.04	1.1059
				192	25.51	900.1
				255	36.04	1.1059
3	256 x	196.662	24	0	100.50	3.5641
				16	69.26	4.3182
				32	49.73	4.4736
				64	33.82	4.4442
				128	26.74	4.3302
				192	25.52	4.236
				255	25.52	4.25
4	512 x	786.486	24	0	100.51	26.7414
				16	63.87	30.3617



Tabel. 3. ( Lanjutan )  
 Hasil perbandingan kompresi dan dekompresi

				32	53.47	30.3556
				64	42.34	31.1216
				128	25.53	30.80925
				192	26.963	29.0776
				255	25.52	27.2572
5	1024 x	3.145.782	24	0	25.88	189.503
				16	25.88	196.0316
				32	25.88	176.06025
				64	25.88	153.8095
				128	25.53	190.4743
				192	25.53	208.4963
				255	25.52	132.2536

## 5. Kesimpulan

Dari hasil pembahasan pada beberapa percobaan diatas dapat ditarik kesimpulan sebagai berikut:

1. Metode pencarian lokal, dapat dipergunakan untuk kompresi citra melalui pemampatan fraktal.
2. Berdasarkan pada hasil perbandingan citra input dan output kompresi, maka diketahui bahwa: toleransi sangat berpengaruh terhadap rasio kompresi, semakin tinggi toleransi kompresi semakin baik (rasio semakin kecil), tetapi semakin tinggi toleransi hasil dekompresi semakin buruk (data hilang semakin tinggi)
3. Berdasarkan grafik hasil penyajian, semakin tinggi ukuran file, maka semakin baik kompresi (rasio semakin kecil)
4. Ukuran file tidak berpengaruh pada waktu kompresi dan dekompresi

## Daftar Pustaka

- Aniati, Murni, A., &Setiawan, S.. (1992).*Pengantar Pengolahan Citra*. Jakarta: PT. Alex Media Komputindo.
- Barnsley, M. F. (1988).*Fractal Everywhere*;New York: Academic Press.
- Fisher, Y. (1992).Fractal Image Compression.*ACM SIGGRAPH Course Note*, vol 12 pp 7.1-7.19.
- Achmad, B. & Firdausy, K. (2005).*Teknik Pengolahan Citra Digital menggunakan Delphi*. Yogyakarta: Andi Publishing.
- Ahmad, U. (2005).*Pengolahan Citra Digital dan Tenik Pemrogramannya*. Yogyakarta: CV. Graha Ilmu.