

**PENERAPAN PEMAMPATAN CITRA DENGAN  
TRANSFORMASI FRAKTAL  
PADA SURATKABAR ELEKTRONIK (*ONLINE*)**

Oleh:

Ir. RINALDI MUNIR NIP: 132084796

Laboratorium Informatika Teori Jurusan Teknik Informatika ITB

**Makalah Penelitian Nomor: 19403598  
Dibiayai oleh DIK – ITB  
Tahun Anggaran 1998/1999**



**INSTITUT TEKNOLOGI BANDUNG  
1998**

## Intisari

Dokumen HTML (*Hypertext Markup Language*) seperti surat kabar elektronik (*online*) di internet terdiri atas unsur teks dan gambar atau citra (*image*). Gambar yang terdapat di dalam surat kabar elektronik pada umumnya berukuran sangat besar sehingga membutuhkan waktu pengiriman yang lama dari situs *web* (*server*) ke komputer *client* melalui saluran telekomunikasi.

Penelitian ini mencoba menerapkan metode pemampatan citra dengan transformasi fraktal pada surat kabar elektronik. Dalam hal ini, gambar disimpan di komputer *server* dalam format fraktal (FRA). Di komputer *client*, gambar direkonstruksi dan ditampilkan di halaman *web*.

Penelitian ini telah membuat dua buah program, yaitu program **PCFrak** (Pemampatan Citra dengan transformasi **Fraktal**) dan program *plug-in* **CitraFraktal**. **PCFrak** memampatkan berkas citra dalam format *bitmap* (BMP) menjadi berkas fraktal (FRA), sedangkan program *plug-in* **CitraFraktal** diintegrasikan dengan program *web browser* *Microsoft Internet Explorer*. Apabila program *web browser* menemukan perintah untuk menampilkan gambar dengan format FRA di dalam dokumen HTML, maka program *plug-in* **CitraFraktal** dipanggil untuk merekonstruksi citra di halaman *web*.

Kelebihan citra dengan format fraktal adalah ukurannya yang kecil serta membutuhkan waktu rekonstruksi citra yang cepat. Karena itu, metode pemampatan fraktal menawarkan format citra alternatif -selain format JPG dan GIF yang populer saat ini- yang dapat digunakan untuk dokumen HTML seperti surat kabar elektronik maupun aplikasi multimedia lainnya.

**Kata kunci:** pemampatan citra, rekonstruksi citra, fraktal, *web*, *plug-in*, HTML

## Abstract

*HTML (Hypertext Markup Language) documents, as electronic newspaper (online) on internet, consist of texts and images. Generally, images have large size (in bytes), and need long time to transfer them on telecommunication channel between web sites (server) and client computers.*

*This research of image compression using fractal transformation method will be applied on electronic newspaper, in which, images are saved at server computer in fractal format (FRA). At client computers, images will be reconstructed to be viewed on web pages.*

*This research has made two programs. They are **PCFrak** (**Pemampatan Citra dengan Transformasi Fraktal – Image Compression using Fractal Compression Method**) and **CitraFractal** plug-in program. **PCFrak** compresses image files of bitmap format (BMP) into image files of fractal format (FRA), and the second program will be integrated in web browser program as Microsoft Internet Explorer after installation. When browser finds instruction to view images of fractal format in a HTML document, **CitraFractal** will be called to reconstruct images on web pages.*

*The advantages of fractal files is small size and fast reconstruction time. The fractal image format can be used for HTML document as newspaper electronic or other multimedia applications. So, fractal image compression offers an alternative image format among other image formats, for examples JPG, GIF, etc.*

*Keywords: image compression, image reconstruction, fractalm web, plug-in, HTML*

## 1. Pendahuluan

Pada umumnya, representasi citra (*image*) digital membutuhkan kapasitas ruang (*space*) penyimpanan yang besar. Pemampatan (*compression*) citra adalah upaya untuk merepresentasikan citra dalam bentuk yang lebih kompak sehingga dapat diperoleh penghematan tempat penyimpanan dan penghematan biaya komunikasi data.

Salah satu metode pemampatan citra yang relatif baru dan memberi harapan yang besar untuk aplikasi yang membutuhkan nisbah pemampatan yang sangat tinggi dan waktu rekonstruksi yang cepat adalah metode pemampatan citra dengan transformasi fraktal (*fractal image compression*). Metode ini dikembangkan oleh Arnaud E. Jacquin, pada tahun 1992, dengan nama *Partitioned Iterated Function System* (PIFS). Dasar pemikirannya adalah bahwa sebuah citra pada umumnya memiliki bagian yang mirip dengan bagian lainnya. Kemiripan tersebut dapat dieksploitasi dengan sekumpulan matriks *transformasi affine*. Proses pemampatan dilakukan dengan membagi citra atas sejumlah blok (disebut jelajah atau *range*) berukuran kecil dan tidak saling beririsan, lalu untuk setiap blok jelajah carilah blok (disebut ranah atau *domain*) citra yang berukuran lebih besar (masih pada citra yang sama) dan paling cocok dengan blok jelajah tersebut. Kemudian, diturunkan transformasi *affine* yang memetakan blok ranah ke blok jelajah.

Keuntungan pemampatan dengan metode fraktal mampu menghasilkan berkas citra – dalam format fraktal- yang berukuran kecil dan waktu rekonstruksi citranya (*decompression*) berlangsung sangat cepat. Sifat yang terakhir ini dapat dimanfaatkan oleh aplikasi yang membutuhkan waktu penayangan citra dengan cepat, seperti aplikasi multimedia atau dokumen elektronik. Proses pemampatan citra cukup dilakukan satu kali saja, sedangkan proses rekonstruksi citra dapat dilakukan kapan saja diinginkan. Citra yang disimpan di dalam media penyimpanan tidak lagi berupa sekumpulan *pixel*, tetapi berupa sekumpulan matriks transformasi *affine*. Proses rekonstruksi akan menampilkan citra ke layar peraga (monitor).

Salah satu aplikasi multimedia yang disebutkan di atas adalah surat kabar elektronik (lebih dikenal dengan sebutan koran *online*). Halaman-halaman surat kabar elektronik disimpan di komputer *server* dalam bentuk dokumen HTML – *HyperText Markup Language*. Program *browser* di komputer client menerjemahkan dokumen HTML yang dikirim dari *server* menjadi halaman-halaman *web*.

Surat kabar elektronik umumnya terdiri atas teks dan gambar (citra). Kelemahan dokumen HTML yang mengandung gambar terletak pada proses pengiriman dokumen dari

*server* ke *client* berlangsung lambat. Hal ini disebabkan citra yang harus dikirim berukuran besar sehingga membutuhkan waktu pengiriman yang lama. Kelemahan ini diatasi dengan mengirimkan citra dalam bentuk mampatnya (yang berukuran lebih kecil dari citra semula). Program *browser* di tempat *client*-lah yang akan melakukan proses rekonstruksi citra, lalu menampilkan citranya di halaman surat kabar elektronik bersama-sama dengan teks.

Program *browser* yang ada saat ini mendukung citra dengan format JPG dan GIF (format BMP juga dimungkinkan, namun jarang digunakan). Format JPG dan GIF masing-masing memiliki kelebihan dan kekurangan. Berkas citra dalam format JPG ukurannya kecil namun waktu rekonstruksinya relatif lambat. Sebaliknya, citra dalam format GIF ukurannya lebih besar namun waktu rekonstruksinya cepat.

Penelitian ini mencoba menawarkan format citra alternatif, yaitu format fraktal (FRA), untuk diterapkan pada surat kabar elektronik. Di tempat *server*, citra disimpan dalam format fraktal, sedangkan di tempat *client* citra direkonstruksi untuk ditampilkan di layar. Karena *browser* tidak mendukung citra dalam format FRA, maka program rekonstruksi harus di-*plug-in* ke dalam *browser*. Bila *browser* menemukan nama berkas yang ber-*extension* FRA di dalam HTML, maka program rekonstruksi dipanggil untuk menampilkan citra di halaman *web*. Dengan cara ini, ada beberapa keuntungan yang diperoleh. Pertama, citra dapat disimpan di *server* dalam ukuran yang lebih kecil. Kedua, sebagai akibat dari keuntungan pertama, waktu pengiriman citra dari *server* ke *client* dapat dipersingkat. Dan keuntungan ketiga, proses rekonstruksi citra di layar *client* berlangsung cepat.

## **2. Tujuan Penelitian**

Penelitian ini bertujuan menerapkan metode pemampatan citra dengan transformasi fraktal pada surat kabar elektronik. Inti dari penerapan metode adalah membuat perangkat lunak pemampatan citra dan program *plug-in* rekonstruksi citra yang diintegrasikan pada program *web browser*. Hasil terapan kemudian dianalisis dan disimpulkan.

## **3. Batasan Masalah**

Untuk penyederhanaan masalah, beberapa batasan perlu dikemukakan di dalam penelitian ini, yaitu:

1. Citra yang akan dimampatkan terbatas pada citra berskala-abu (*greyscale image*) saja. Program pemampatan belum dapat diterapkan untuk gambar berwarna.
2. Format citra yang dapat dimampatkan adalah citra dengan format *bitmap* (BMP). Program pemampatan belum dapat diterapkan untuk format yang lain.
3. Program *web browser* HTML yang digunakan adalah *Microsoft Internet Explorer* versi 4.0 ke atas.
4. Program *plug-in* rekonstruksi citra harus ditempatkan di komputer pengguna. Pengguna diandaikan sudah menginstalasi program *plug-in* tersebut di komputernya.

#### **4. Metodologi Penelitian**

Penelitian ini dibuat dengan mengikuti kaidah metodologi ilmiah. Metodologi penelitian yang dilakukan adalah sebagai berikut:

1. Kajian literatur.  
Mengkaji berbagai literatur tentang fraktal, pemampatan citra dengan transformasi fraktal, surat kabar elektronik, dan literatur tentang pembuatan program *plug-in*.
2. Analisis kebutuhan.  
Mengidentifikasi kebutuhan pembuatan program pemampatan citra dan program *plug-in* rekonstruksi citra
3. Pembangunan perangkat lunak pemampatan fraktal dan program *plug-in* rekonstruksi citra.  
Merancang dan mengimplementasikan perangkat lunak pemampatan fraktal beserta program *plug-in* rekonstruksi citra fraktal. Program *plug-in* diintegrasikan pada program *web browser*.
5. Membuat surat kabar elektronik yang menggunakan citra dengan format fraktal pada dokumen HTML-nya.
6. Melakukan analisis terhadap hasil yang diperoleh, dan menyarikannya menjadi kesimpulan dan saran pengembangan.

#### **5. Dasar Teori**

##### *5.1 Iterated Function System (IFS)*

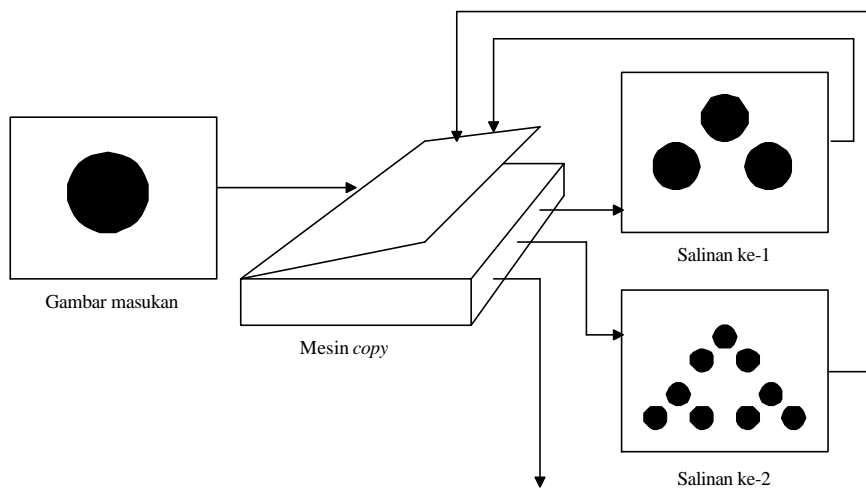
Sejarah fraktal dapat dirunut dari buku Benoit Mandelbrot yang berjudul *The Fractal Geometry of Nature* [MAN82]. Fraktal adalah obyek yang memiliki kemiripan dirinya-

sendiri (*self-similarity*) namun dalam skala yang berbeda [NEL96]. Ini artinya, bagian-bagian dari obyek akan tampak sama dengan obyek itu sendiri bila dilihat secara keseluruhan. Gambar 1 memperlihatkan tiga buah contoh fraktal, yaitu segitiga Sierpinski, daun pakis Barsnsley, dan pohon fraktal.



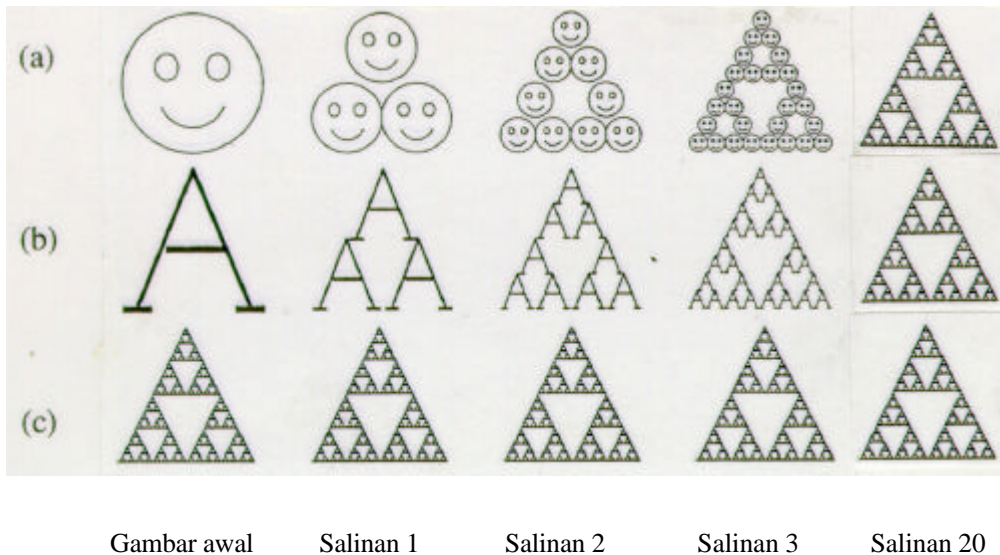
**Gambar 1** Segitiga Sierpinski, daun pakis Barsnsley, dan pohon fraktal

Adalah Michael Barnsley (1988) yang merepresentasikan fraktal ke dalam model matematika yang dinamakan IFS (*Iterated Function System*), melalui buku, *Fractals Everywhere* [BAR88]. IFS dimetaforakan sebagai sebuah mesin *foto copy* yang disebut *Multiple Reduction Copy Machine* (MRCM). MRCM memiliki banyak lensa dan setiap lensa melakukan pengecilan gambar dalam jumlah yang banyak. Gambar dihasilkan dari mesin *copy* dioperasikan kembali sebagai masukan untuk membuat salinan berikutnya (Gambar 2).



**Gambar 2** *Multiple Reduction Copy Machine* (MRCM) [KOM95]

Hal yang menarik dari MRCM adalah, apapun gambar awal yang digunakan, MRCM selalu konvergen ke gambar akhir yang sama. Gambar 3 memperlihatkan hasil salinan setelah beberapa kali lelaran dari MRCM yang disusun oleh tiga buah lensa, setiap lensa memiliki faktor pengecilan setiap lensa adalah  $\frac{1}{2}$ . Gambar akhir yang dihasilkan selalu segitiga Sierpinski.



**Gambar 3** Apapun gambar awalnya, MRCM selalu menghasilkan segitiga Sierpienski [FIS94].

Secara matematis, 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 penggeseran terhadap salinan (*copy*) citra masukan.

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

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

Sembarang titik  $(x,y)$  pada gambar masukan 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*  $w_i$  menghasilkan salinan citra yang lebih kecil; yaitu, untuk sembarang citra awal  $A$  yang diberikan, dihasilkan salinan *affine*,  $w_1(A), w_2(A), \dots, w_n(A)$ . Gabungan dari seluruh salinan tersebut adalah  $W(A)$ , yang merupakan keluaran dari mesin,



$$W(A) = w_1(A) + w_2(A) + \dots + w_n(A) \quad (3)$$

$W$ , yang dinamakan operator Hutchinson, adalah gabungan (*collage*) dari sejumlah transformasi individual  $w_i$ , yaitu

$$W = w_1 \cup w_2 \cup \dots \cup w_n = \bigcup_{i=1}^n w_i \quad (4)$$

Setiap transformasi *affine*  $w_i$  bersifat kontraktif, yaitu  $w_i$  memetakan dua buah titik menjadi lebih dekat. Ini berlaku untuk semua titik di bidang citra. Akibatnya, MRCM menghasilkan salinan gambar yang ukurannya lebih kecil daripada ukuran gambar semula.

Sifat kontraktif saja tidak begitu penting. Sifat ini menjadi penting bila MRCM dijalankan dengan skema kalang umpan-balik terhadap gambar awal. Yaitu, diberikan citra awal  $A_0$ , diperoleh

$$\begin{aligned} A_1 &= W(A_0) = \bigcup_{i=1}^n w_i(A_0) \\ A_2 &= W(A_1) = W(W(A_0)) = W^2(A_0) \\ A_3 &= W(A_2) = W(W(A_1)) = W(W(W(A_0))) = W^3(A_0) \\ &\dots \\ A_n &= W^n(A_0) \end{aligned}$$

Jika  $W$  seluruhnya kontraktif, maka untuk  $n \rightarrow \infty$  lelarannya konvergen ke sebuah citra yang unik,  $A_\infty$ . Citra  $A_\infty$  disebut titik-tetap (*fixed-point*) atau *invariant* dari proses lelaran, dan *attractor* dari  $W$ . Titik-tetap adalah citra  $A_\infty$  sedemikian sehingga

$$A_\infty = W(A_\infty) \quad (5)$$

Jadi, jika  $A_\infty$  dipilih sebagai citra awal, maka tidak ada perubahan pada hasil transformasinya. Sedangkan *attractor* dari  $W$  adalah citra  $A_\infty$  sedemikian sehingga

$$A_\infty = \lim_{n \rightarrow \infty} W^n(A_0) \text{ untuk sembarang citra awal } A_0. \quad (6)$$

Persamaan yang terakhir ini menyatakan bahwa tidak peduli apa pun citra awal yang digunakan, limit lelarannya selalu menghasilkan citra akhir yang sama. Dengan kata lain, citra *attractor* adalah unik.

Transformasi *affine* yang menghasilkan citra titik-tetap segitiga Sierpinski adalah

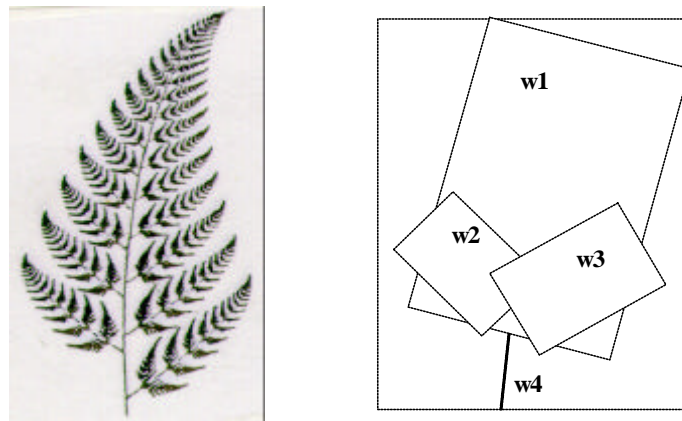
$$w_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \quad w_2 = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix} \quad w_3 = \begin{bmatrix} 0.5 & 0 & 0.25 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

Jika jumlah transformasi meningkat menjadi empat dengan setiap  $w_i$  adalah sebagai berikut

$$w_1 = \begin{bmatrix} 0.85 & 0.04 & 0.00 \\ -0.04 & 0.85 & 1.6 \end{bmatrix} \quad w_2 = \begin{bmatrix} 0.20 & -0.26 & 0.00 \\ 0.23 & 0.22 & 1.6 \end{bmatrix}$$

$$w_3 = \begin{bmatrix} -0.15 & 0.28 & 0.00 \\ 0.26 & 0.52 & 0.44 \end{bmatrix} \quad w_4 = \begin{bmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.16 & 0.00 \end{bmatrix}$$

maka MRCM konvergen ke citra fraktal yang terkenal, yang dinamakan tanaman pakis Barnsley (*Barnsley's fern*) – Gambar 4. Di sini,  $w_1$  mengendalikan keseluruhan bentuk,  $w_2$  membangkitkan daun kiri,  $w_3$  membangkitkan daun kanan, dan  $w_4$  menghasilkan batang.



**Gambar 4** Pakis Barnsley dan empat buah transformasi *affine*-nya

## 5.2 Pengkodean Citra dengan IFS

Menyimpan citra sebagai kumpulan *pixel* membutuhkan memori yang besar, namun bila yang disimpan adalah transformasi *affine*-nya, maka memori yang dibutuhkan jauh lebih sedikit. Cara ini melahirkan gagasan pengkodean citra dengan nisbah pemampatan yang tinggi. Pakis Barnsley misalnya, dibangkitkan dengan empat buah transformasi *affine*, masing-masingnya terdiri atas enam buah bilangan riil (4 *byte*), sehingga dibutuhkan  $4 \times 6 \times 4 \text{ byte} = 96 \text{ byte}$  untuk menyimpan keempat transformasi itu. Bandingkan bila citra pakis Barnsley disimpan dengan representasi *pixel* hitam putih (1 *pixel* = 1 *byte*) berukuran  $550 \times 480$  membutuhkan memori sebesar 264.000 *byte*. Maka, nisbah pemampatan citra pakis adalah  $264.000 : 96 = 2750 : 1$ , suatu nisbah yang sangat tinggi.

Kesulitan utama pemampatan dengan IFS adalah menemukan bagian citra yang mirip dengan keseluruhan citra. Intervensi manusia diperlukan untuk memandu menemukan bagian citra yang mirip dengan citra secara keseluruhan. Selain itu, pemampatan citra dengan IFS yang telah dikemukakan di atas hanya dapat diterapkan untuk citra yang memiliki *self-similarity* saja. Citra alami, disamping mempunyai warna, hampir tidak pernah *self-similar* secara keseluruhan. Karena itu, pemampatan sembarang citra (baik citra berwarna maupun citra *greylevel*) dengan IFS tidak dapat dilakukan.

Terobosan penting dibuat oleh Arnaud D. Jacquin, mahasiswa Barnsley. Melalui tulisan monumentalnya pada tahun 1992, Jacquin meyajikan skema otomatis pengkodean citra yang dikenal dengan nama *Partitioned Iterated Function System* (PIFS) [JAC92]. PIFS dapat digunakan untuk memampatkan sembarang citra, baik citra skala-abu maupun citra berwarna, dan tidak terbatas untuk citra fraktal saja.

#### 5.4 *Partitioned Iterated Function System (PIFS)*

Citra alami (*natural image*) umumnya hampir tidak pernah *self-similar* secara keseluruhan. Karena itu, citra alami pada umumnya tidak mempunyai transformasi *affine* terhadap dirinya sendiri. Tetapi, untunghlah citra alami seringkali memiliki *self-similarity* lokal, yaitu memiliki bagian citra yang mirip dengan bagian lainnya, misalnya citra berskala-abu (*greyscale*) Lena pada Gambar 5.



**Gambar 5** Kemiripan lokal pada citra Lena

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. Setiap transformasi itu disebut IFS lokal. Gabungan dari seluruh hasil transformasi itu adalah citra fraktal yang menyerupai (atau menghampiri) citra semula.

Langkah pertama yang dilakukan dalam proses pemampatan adalah membagi citra atas sejumlah blok yang berukuran sama dan tidak saling beririsan, yang disebut blok jelajah (*range*). Untuk menyederhanakan masalah, blok jelajah diambil berbentuk bujursangkar. Untuk setiap blok jelajah, dicari bagian citra yang berukuran lebih besar dari blok jelajah –yang disebut blok ranah (*domain*)– yang mirip (cocok) dengan blok jelajah tersebut, kemudian turunkan transformasi *affine* (IFS lokal)  $w_i$  yang memetakan blok ranah ke blok jelajah. Hasil dari semua pemasangan ini adalah *Partitioned Iterated Function System* (PIFS).

Kemiripan antara dua buah (blok) citra diukur dengan metrik jarak. Metrik jarak yang banyak digunakan dalam raktek adalah jarak RMS (*Root Mean Square*) atau nama lainnya jarak Euclidean, yaitu

$$L_2 = \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \quad (10)$$

dengan  $z$  dan  $z'$  adalah nilai intensitas *pixel* dua buah citra. jarak

Transformasi *affine*  $w$  untuk citra berskala-abu disusun oleh bagian spasial yang memetakan posisi *pixel* di blok ranah ke posisi *pixel* di blok jelajah, dan bagian intensitas yang mengubah nilai intensitas *pixel*. Titik  $(x,y)$  dengan intensitas  $z$  yang termasuk di dalam blok ranah  $D_i$  dipetakan oleh  $w$  menjadi

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (11)$$

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

$$z' = s_i z + o_i \quad (12)$$

Parameter  $s_i$  menyatakan faktor kontras *pixel* (seperti tombol kontras di TV). Bila  $s_i$  bernilai 0, maka *pixel* menjadi gelap, bila  $s_i$  sama dengan 1 kontrasnya tidak berubah; antara 0 dan 1 *pixel* berkurang kontrasnya, di atas 1 kontrasnya bertambah. Parameter  $o_i$  menyatakan ofset kecerahan (*brightness*) *pixel* (seperti tombol kecerahan di TV. Nilai  $o_i$  positif mencerahkan gambar dan nilai  $o_i$  negatif menjadikannya gelap. Kedua parameter

tersebut dapat memetakan secara akurat blok jelajah berskala abu ke blok jelajah berskala abu.

Untuk menjamin efek kontraktif dalam arah spasial, maka blok ranah harus berukuran lebih besar daripada blok jelajah. Untuk alasan praktis, ukuran blok ranah diambil dua kali ukuran blok jelajah (perbandingannya 2:1). Jadi, jika ukuran blok jelajah adalah  $B \times B \text{ pixel}$ , maka ukuran blok ranah adalah  $2B \times 2B \text{ pixel}$ . Perbandingan ini membuat transformasi *affine* menjadi lebih sederhana, yaitu

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (13)$$

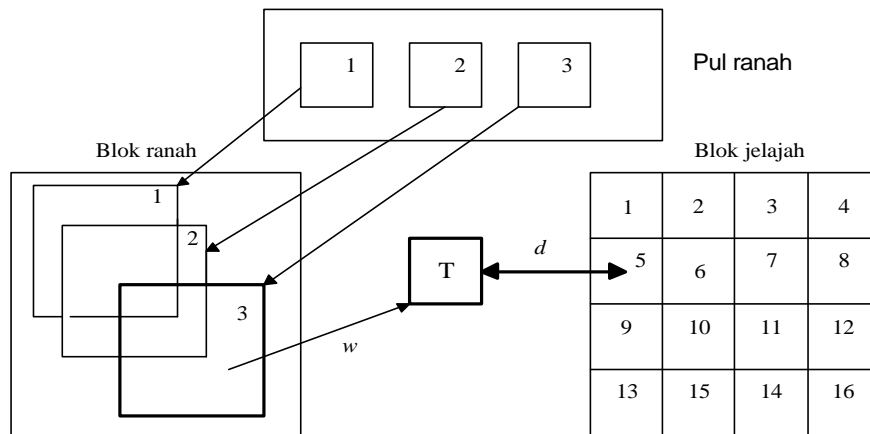
### 5.5 Algoritma Pemampatan Citra dengan PIFS

Jantung dari proses pemampatan dengan PIFS adalah menemukan kecocokan blok jelajah dengan blok ranah. Citra dibagi atas sejumlah blok ranah yang mungkin saling beririsan. Himpunan blok ranah yang digunakan dalam proses pencarian kecocokan disebut *pul ranah (domain pool)*. Pul ranah yang besar menghasilkan kualitas pemampatan yang lebih baik, tetapi membutuhkan waktu pencocokan yang lebih lama.

Sebelum proses pencocokan dimulai, setiap blok ranah di dalam kantong ranah diskalakan sehingga ukurannya sama dengan ukuran blok jelajah. Penskalaan ini dimaksudkan agar jarak antara blok jelajah dan blok ranah mudah dihitung dengan persamaan (10). Penskalaan dilakukan dengan menjadikan  $2 \times 2$  buah *pixel* menjadi satu buah *pixel* [KOM95]. Nilai satu buah *pixel* tersebut adalah rata-rata nilai keempat *pixel*.

Pada proses pencocokan, setiap blok jelajah  $R_i$  diperiksa dengan sejumlah blok ranah di dalam pul ranah. Untuk setiap blok ranah  $D_i$ , harus ditentukan transformasi *affine*  $w_i$  yang memetakan  $D_i$  ke  $R_i$ . Untuk setiap pencocokan, parameter  $e_i$ ,  $f_i$ ,  $s_i$ , dan  $o_i$  dari persamaan (13) dihitung. Parameter  $e_i$  dan  $f_i$  mudah dihitung karena keduanya menyatakan pergeseran sudut kiri blok ranah ke sudut kiri blok jelajah yang bersesuaian. Sedangkan  $s_i$  dan  $o_i$  dihitung dengan rumus regresi [FIS95].

Selanjutnya, transformasi *affine*  $w_i$  diuji terhadap blok ranah  $D_i$  untuk menghasilkan blok uji  $T_i = w_i(R_i)$  (Gambar 6). Jarak antara  $T$  dan  $R_i$  dihitung dengan persamaan (17). Transformasi *affine* yang terbaik ialah transformasi  $w$  yang meminimumkan jarak antara  $R_i$  dan  $T$ .



**Gambar 6** Blok jelajah 5 dibandingkan dengan blok ranah 3 di dalam pul ranah. Transformasi  $w$  ditentukan, lalu blok ranah 3 ditransformasikan dengan  $w$  menghasilkan  $T$ . Jarak antara  $T$  dengan blok jelajah 5 diukur.

Runtunan pencarian dilanjutkan untuk blok jelajah berikutnya sampai seluruh blok jelajah sudah dipasangkan dengan blok ranah. Hasil dari proses pemampatan adalah sejumlah IFS lokal yang disebut PIFS. Seluruh parameter PIFS di-pak dan disimpan di dalam berkas eksternal.

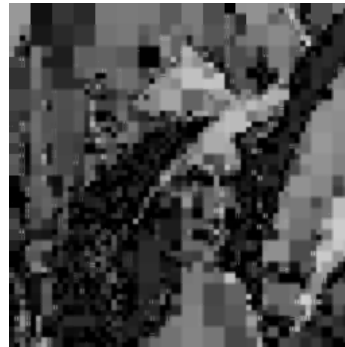
### 5.6 Rekonstruksi Citra

Rekonstruksi (*decoding*) citra dilakukan dengan melelarkan PIFS dari citra awal sembarang. Karena setiap IFS lokal kontraktif, baik kontraktif dalam matra intensitas maupun kontraktif dalam matra spasial maka berdasarkan Teorema Pemetaan Kontraktif lelarannya akan konvergen ke citra titik-tetap PIFS. Kontraktif intensitas penting untuk menjamin konvergensi ke citra semula, sedangkan kontraktif spasial berguna untuk membuat rincian pada citra untuk setiap skala. Jika PIFS yang ditemukan selama proses pemampatan bagus, yaitu gabungan dari transformasi seluruh blok ranah dekat dengan citra semula (diukur dengan metrik jarak yang dipakai), maka titik-tetap PIFS juga dekat dengan citra semula tersebut.

Citra Awal



Lelaran 1



Lelaran 3



Lelaran 10



**Gambar 7** Rekonstruksi citra Lena dari citra awal yang homogen hitam.

Selama proses pemulihan, setiap IFS lokal mentransformasikan sekumpulan blok ranah menjadi sekumpulan blok jelajah. Karena blok jelajah tidak saling beririsan dan mencakup keseluruhan *pixel* citra, maka gabungan seluruh blok jelajah menghasilkan citra titik-tetap yang menyerupai citra semula. Gambar 7 memperlihatkan hasil rekonstruksi citra Lena setelah 1, 3, dan 10 lelaran. Citra awal yang digunakan adalah citra hitam.

Konvergensi ke citra titik-tetap berlangsung cepat. Konvergensi umumnya dapat diperoleh dalam 8 sampai 10 kali lelaran [NEL96]. Karena transformasi *affine* kontraktif dalam arah spasial, maka semakin banyak rincian citra yang dibuat pada setiap lelaran.

### 5.7 Surat kabar Elektronik

Surat kabar elektronik adalah dokumen yang dibuat dengan bahasa HTML (*Hypertext Markup Language*). Dokumen HTML ditampilkan di layar oleh program *browser World Wide Web* (WWW) dalam bentuk yang dikehendaki oleh pembuatnya. Dokumen HTML

sebenarnya adalah berkas teks yang berisi data teks dan gambar atau citra (*image*) yang akan ditampilkan dan memuat perintah yang memberi pengetahuan bagi *browser* bagaimana data itu harus ditampilkan [KUR97].

Kelemahan terbesar pada dokumen elektronik yang mengandung gambar adalah persoalan pengiriman berkas gambar melalui saluran telekomunikasi. Karena berkas citra seringkali berukuran sampai *megabyte*, maka pengiriman berkas gambar melalui saluran telekomunikasi membutuhkan waktu yang lebih lama dibandingkan dengan

Format citra yang terkenal dan paling banyak digunakan oleh program *browser* adalah GIF dan JPG. Gambar berformat GIF dimampatkan dengan metode LZW, sedangkan JPG dimampatkan dengan metode JPEG [KUR98]. Baik format GIF maupun format JPG keduanya memiliki kelebihan dan kekurangan. Berkas GIF mempunyai ukuran yang lebih besar daripada JPG, namun waktu untuk merekonstruksi citranya lebih cepat. Sebaliknya, berkas JPG memiliki ukuran yang kecil (sehingga mempercepat waktu pengiriman), namun membutuhkan waktu yang lebih lama untuk merekonstruksi citranya.

Penelitian ini mencoba menawarkan format citra alternatif yang dapat digunakan untuk dokumen *web* seperti surat kabar elektronik. Alasan yang mendasari adalah ukuran format fraktal dapat dibuat lebih kecil daripada berkas GIF atau JPG (dengan kualitas gambar yang sama) sehingga mempercepat waktu pengiriman dari *server* ke *client*. Selain itu, citra dalam format fraktal dapat direkonstruksi di halaman *browser* dalam waktu yang singkat.

## **6. Pembangunan Perangkat Lunak PCFrak**

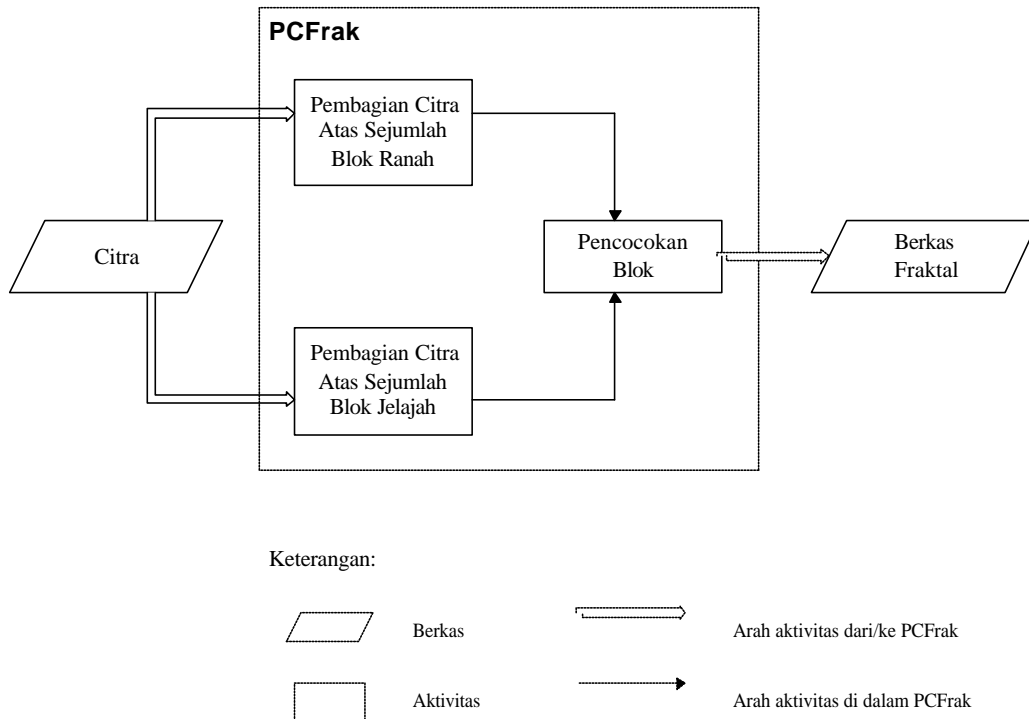
PCFrak adalah perangkat lunak yang ditujukan untuk memampatkan berkas citra alami menjadi berkas fraktal dengan nama format FRA (dari kata FRAktal). Berkas fraktal berukuran jauh lebih kecil daripada berkas citra semula, sehingga dapat menghemat kebutuhan memori tempat penyimpanannya.

PCFrak menerima masukan berkas citra dan parameter yang berhubungan dengan kualitas hasil pemampatan dan nisbah pemampatan. Kualitas dan nisbah pemampatan memiliki adalah dua faktor yang berlawanan. Pengguna PCFrak diberi keleluasaan untuk menentukan faktor mana yang diutamakan, atau mengimbangkan kedua faktor tersebut dengan memilih parameter yang sesuai.

Arsitektur perangkat lunak PCFrak digambarkan secara global pada Gambar 8. PCFrak menerima citra yang akan dimampatkan, dan melakukan aktivitas membagi citra



menjadi sejumlah blok ranah, membagi citra yang sama menjadi sejumlah blok jelajah. Selanjutnya, setiap blok jelajah dicocokkan dengan blok ranah untuk mendapatkan transformasi *affine* yang memetakan blok ranah ke blok jelajah. Kumpulan transformasi *affine* disimpan di dalam berkas fraktal.

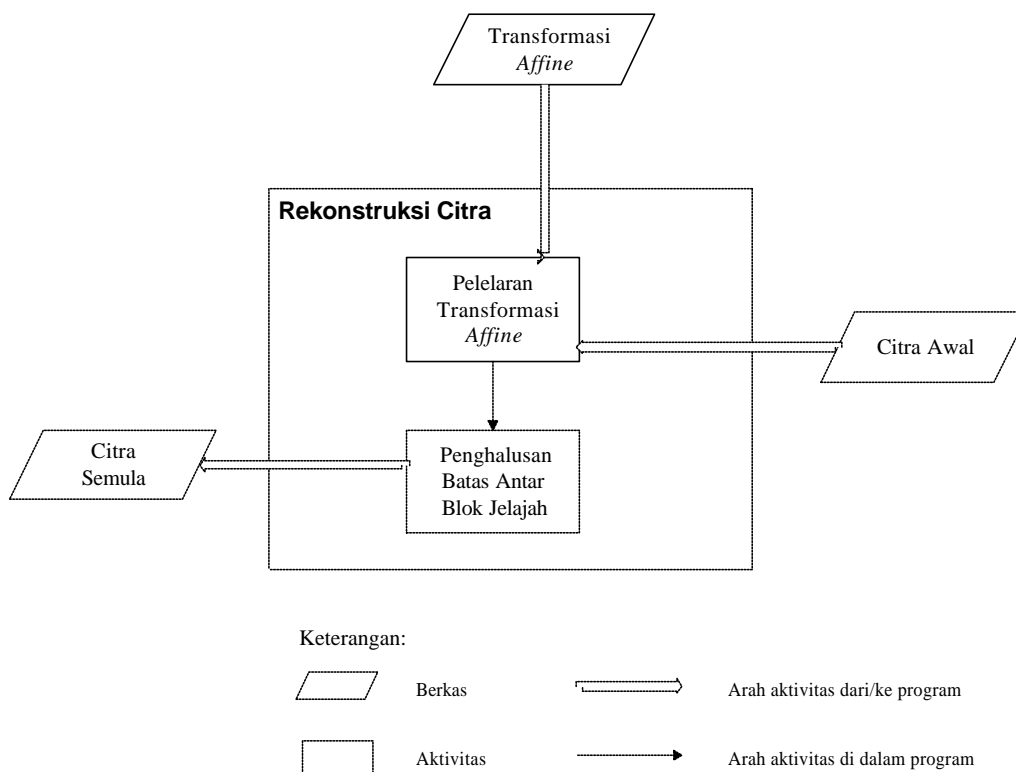


**Gambar 8** Arsitektur PCFrak

## 7. Pembangunan Program *Plug-in* CitraFraktal

Program *plug-in* yang dibuat, yang diberi nama *CitraFraktal Control*, sebenarnya adalah kendali *ActiveX* (*ActiveX control*) yang digunakan untuk menampilkan citra dengan format fraktal pada halaman *web*. Program *plug-in* ini berisi modul rekonstruksi citra yang akan dipanggil bila di dalam dokumen HTML terdapat perintah untuk menampilkan citra dengan format fraktal (FRA).

Modul program rekonstruksi citra menerima masukan berupa berkas fraktal yang berisi sekumpulan transformasi *affine*. Selanjutnya, setiap transformasi *affine* dilelarkan terhadap citra awal sembarang untuk memperoleh kembali gabungan blok jelajah yang membentuk citra semula. Pemrosesan akhir dilakukan untuk memperhalus batas antar blok jelajah. Gambar 9 memperlihatkan arsitektur program rekonstruksi citra.



**Gambar 9** Arsitektur modul program rekonstruksi citra.

Modul rekonstruksi citra tersebut disisipkan ke dalam program *plug-in* CitraFraktal sehingga dapat dipanggil oleh *browser* HTML. Bila *browser* HTML menemukan perintah untuk menampilkan gambar yang berformat fraktal (\*.FRA), maka modul rekonstruksi dipanggil untuk menampilkan gambar bersama-sama dengan teks di halaman *web*.

Kendali CitraFraktal dirancang untuk bekerja pada sistem operasi *Microsoft Windows 95* dan *98*. Kendali CitraFraktal digunakan dengan cara mengintegrasikannya pada *Microsoft Internet Explorer* versi 4 atau lebih. Untuk pemakai *Netscape Navigator*, kendali CitraFraktal dapat digunakan jika pada *Netscape Navigator*-nya telah memiliki *Active plug-in*. Salah satu *Active plug-in* untuk *Navigator* dibuat oleh NCompassLabs yang dapat di-download dari <http://www/ncompasslabs.com>

Untuk menginstalasi kendali CitraFraktal pada komputer pengguna (*client*), download-lah berkas *citrafraktal.exe* dari:

[http://www/if.itb.ac.id/deptinfo/research/fractal/citrafractal.exe](http://www.if.itb.ac.id/deptinfo/research/fractal/citrafractal.exe)

kemudian jalankan program tersebut dan kendali CitraFraktal sudah siap digunakan.

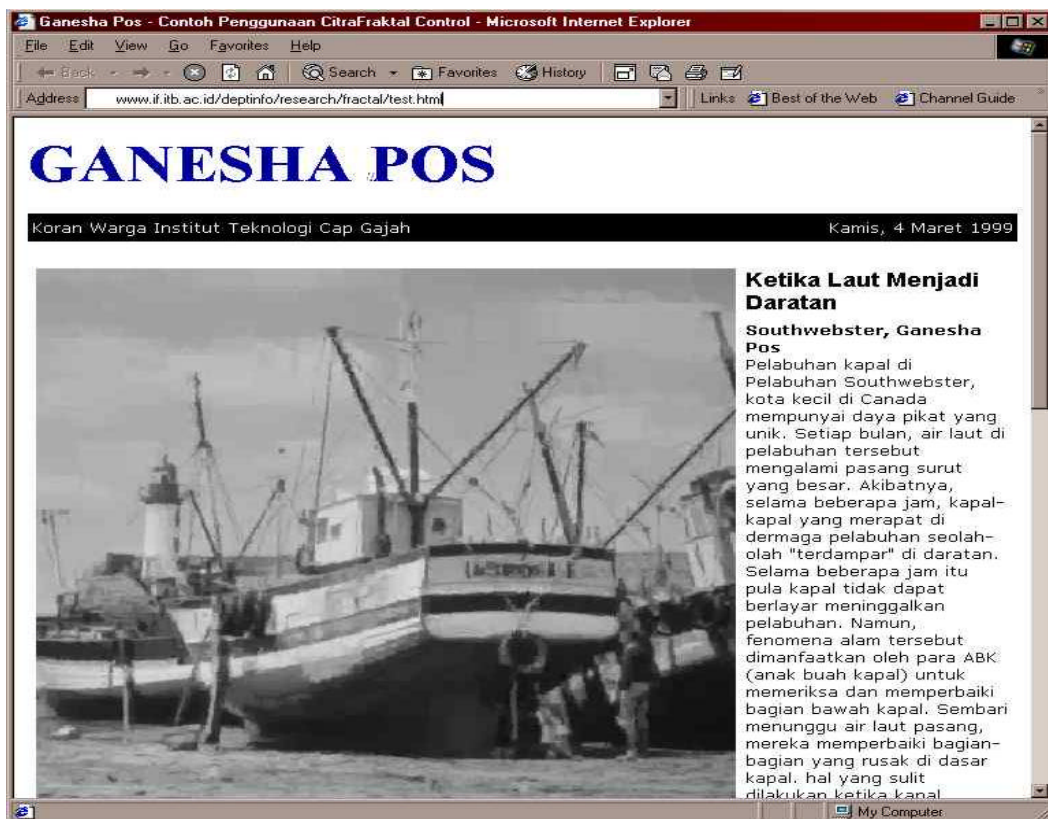
Kendali CitraFraktal memiliki ClassId: 161A5BC5-BCBE-11D2-8EFB-A9332D3C5859, sehingga untuk menggunakannya pada dokumen HTML, tag seperti contoh berikut harus disisipkan:

```
<OBJECT ID = "CitraFraktal1" WIDTH = 300 HEIGHT = 300
  CLASSID = "CLSID: 161A5BC5-BCBE-11D2-8EFB-A9332D3C5859">
<param name = "stretch" value = "true">
<param name = "urlink" value = "www.if.itb.ac.id">
<param name = "source" value = "colli256.fra">
<param name = "alt" value = "colli256 - citra"
</OBJECT
```

## 8. Pembuatan Surat kabar Elektronik

Surat kabar elektronik yang dibuat diberi nama GANESHA POS (Gambar 10). Untuk ujicoba penelitian ini, dokumen HTML GANESHA POS diberi nama *test.html*. Bukalah halaman surat kabar GANESHA POS dengan *browser Microsoft Internet Explorer* versi 4. Alamat situs GANESHA POS adalah:

<http://www.if.itb.ac.id/deptinfo/research/fractal/test.html>



**Gambar 10** Tampilan halaman surat kabar elektronik GANESHA POS

## 9. Analisis Hasil

### 9.1 Analisis Terhadap Ukuran Berkas Citra

Secara umum, berkas citra hasil pemampatan mempunyai ukuran yang lebih kecil daripada ukuran citra semula. Nisbah (*ratio*) pemampatan dihitung dengan rumus:

$$\frac{\text{ukuran citra semula} - \text{ukuran citra mampat}}{\text{ukuran citra semula}} \times 100\%$$

Tabel 1 di bawah ini berisi perbandingan ukuran empat buah citra sebelum dan sesudah dimampatkan.

**Tabel 1** Perbandingan ukuran citra sebelum dan sesudah dimampatkan

No.	Citra BMP ( <i>byte</i> )	Ukuran ( <i>byte</i> )	Citra FRA ( <i>byte</i> )	Ukuran	Nisbah (%)
1	KAPAL512.BMP	263.222	KAPAL512.FRA	8.956	96,6
2	LENA256.BMP	66.614	LENA256.FRA	8.137	87,6
3	COLLI256.BMP	66.614	COLLI256.FRA	9.150	86,3
4	POTRET.BMP	128.782	POTRET.FRA	17.437	86,5

**Tabel 2** Perbandingan ukuran citra berformat BMP, JPG, GIF, dan BMP

Nama Citra	Format BMP ( <i>byte</i> )	Format JPG ( <i>byte</i> )	Format GIF ( <i>byte</i> )	Format FRA ( <i>byte</i> )
KAPAL512	263.222	24.367	242.452	8.956
LENA256	66.614	7.126	70.292	8.137
COLLI256	66.614	7.021	69.965	9.150
POTRET	128.782	16.377	136.377	17.437

Dari Tabel 1 di atas terlihat bahwa metode pemampatan fraktal dapat menghasilkan ukuran berkas fraktal dengan nisbah pemampatan yang tinggi. Ukuran berkas fraktal yang kecil ini sangat menguntungkan untuk aplikasi yang membutuhkan pengiriman berkas gambar melalui saluran telekomunikasi, misalnya aplikasi surat kabar elektronik, sebab waktu pengiriman berkas gambar dari *server* ke komputer *client* dapat dipersingkat.

Tabel 2 di bawah ini memperlihatkan perbandingan antara berkas citra dari berbagai format yang didukung oleh program *web browser*, yaitu format BMP, format mampat JPG, format GIF, serta format mampat FRA. Citra dengan gormat JPG dan format GIF

diperoleh dengan menggunakan *Microsoft Photo Editor*. Dari tabel 2 tersebut terlihat bahwa ukuran citra dengan format fraktal tidak berbeda jauh dengan format JPG. Untuk contoh citra yang diukur di sini, format GIF memberikan hasil pemampatan yang buruk.

### 9.2 Analisis Terhadap Waktu Rekonstruksi Citra

Salah satu kelebihan citra dengan format fraktal adalah waktu untuk merekonstruksi citra dan menampilkannya ke halaman *web* berlangsung cepat. Untuk berkas citra yang digunakan di dalam dokumen HTML surat kabar GANESHA POS di atas, yaitu antara 2 sampai 4 detik untuk setiap citra (citra KAPAL512.FRA membutuhkan waktu kira-kira 4 detik untuk ditampilkan ke halaman *web*, sedangkan citra COLLI256 dan LENA256.FRA membutuhkan waktu kira-kira 2 detik). Komputer yang digunakan pada pengukuran ini memiliki *processor* Pentium dan memori 32 MB. Sistem operasi yang digunakan adalah *Microsoft Windows 98*, dan *web browser* yang digunakan adalah *Microsoft Internet Explorer 4.0*. Untuk pengukuran waktu ini, dokumen HTML surat kabar GANESHA.POS (*test.html*) serta berkas citra fraktalnya tidak dibuka dari *server* yang terhubung di internet, tetapi dari *harddisk* komputer lokal. Hal ini dilakukan agar pengukuran waktu tidak bergantung pada lama pengiriman dokumen HTML dan berkas fraktal dari *server* ke komputer *client*.

Waktu rekonstruksi citra yang singkat ini sangat menguntungkan untuk aplikasi yang membutuhkan penayangan gambar ke layar dengan cepat, seperti surat kabar elektronik ini. Karena itu, format citra fraktal dapat menjadi alternatif penyimpanan gambar di internet.

### 9.3 Analisis Terhadap Penggunaan Format Fraktal pada Surat kabar Elektronik

Karena *web browser* seperti *Internet Explorer* dan *Netscape Navigator* tidak memfasilitasi citra dengan format fraktal, maka di komputer *client* haruslah diinstalasi program *plug-in* CitraFraktal terlebih dahulu. Pengguna harus menjalankan program program *plug-in* agar tersebut agar gambar berformat fraktal yang ada di dokumen HTML dapat ditampilkan di halaman *web* oleh program *browser*.

Masalah timbul karena tidak semua orang sempat atau mau bersusah payah untuk menginstalasi dan menjalankan program *plug-in* CitraFraktal di komputernya. Masalah ini tidak muncul pada citra dengan format JPG atau GIF, sebab program *browser* umumnya sudah mengandung modul rekonstruksi citra untuk kedua format tersebut. Kelemahan ini

sebenarnya dapat diatasi dengan cara mengotomasikan proses instalasi dan eksekusi program *plug-in* CitraFraktal. Caranya, bila *browser* tidak menemukan program *plug-in* CitraFraktal di komputer *client*, maka program *plug-in* tersebut di-*download* secara otomatis dari *server* ke komputer *client*, lalu mengaktifkannya. Namun cara tersebut tidak dapat dilakukan sekarang karena pemilik *server* harus membayar royalti ke *World Wide Consortium* di Amerika Serikat, dan untuk saat ini hanya orang Amerika atau perusahaan Amerika Serikat saja yang diperkenankan membeli royalti tersebut.

## 10. Kesimpulan dan Saran

### 10.1 Kesimpulan

Citra dengan format fraktal dapat dijadikan alternatif penyimpanan gambar di lingkungan internet, selain format JPG dan GIF yang sudah umum digunakan saat ini. Citra dengan format fraktal dapat disimpan dengan persentase pemampatan yang tinggi, sedangkan waktu untuk merekonstruksinya berlangsung cepat. Pemampatan citra cukup dilakukan satu kali saja dengan program PCFrak, sedangkan rekonstruksi citranya dapat dilakukan kapan saja.

Suratkabar elektronik yang dibuat di dalam penelitian ini menggunakan citra dengan format fraktal di dalam dokumen HTML-nya. Karena program *web browser* tidak mendukung format ini, maka di komputer *client* harus diinstalasi dan dijalankan program *plug-in* rekonstruksi citra yang diberi nama kendali CitraFraktal. Untuk saat ini, program *web browser* yang dapat digunakan adalah *Microsoft Internet Explorer* versi 4.0 ke atas.

Hasil penelitian memperlihatkan bahwa, untuk citra yang sama, citra dengan format fraktal mempunyai ukuran yang tidak begitu jauh berbeda dengan citra dalam format JPG. Penggunaan citra dengan format fraktal pada dokumen HTML menunjukkan bahwa waktu untuk merekonstruksi citranya di halaman web hanya membutuhkan waktu kurang dari 4 detik.

### 10.2 Saran

PCFrak dan *plug-in* CitraFraktal yang dibuat di dalam penelitian ini masih memiliki berapa kekurangan dan kelemahan, yaitu:

1. PCFrak baru dapat memampatkan citra bitmap berskala-abu (*greyscale*) dengan derajat keabuan dari 0 sampai 255. Pengembangan lebih lanjut seharusnya dilakukan sehingga

PCFrak dapat memampatkan citra berwarna. Untuk citra dengan 256 warna, pengembangan ini mudah dilakukan dengan cara menyimpan palet warnanya ke dalam berkas fraktal. Palet warna digunakan pada saat menampilkan citra ke layar.

2. Perlu diupayakan agar program *plug-in* rekonstruksi CitraFraktal dapat di-*download* secara otomatis dari *server* apabila program *web browser* tidak menemukan *plug-in* tersebut di komputer *client*.
3. Dengan teknologi *ActiveX* pada pengembangan kendali CitraFraktal, diharapkan penggunaan citra berformat fraktal dapat lebih meluas lagi, tidak hanya pada dokumen HTML saja, melainkan juga pada aplikasi lainnya. Untuk itu, perlu dilakukan pengembangan pada kendali CitraFraktal yaitu pada halaman propertinya (*property page*), sehingga kendali CitraFraktal dapat diakses dengan mengisi nilai propertinya secara langsung melalui *edit box*, tidak lagi dengan menggunakan *script* seperti yang telah digunakan pada dokumen HTML.

#### Daftar Pustaka

- [ANS93] Anson, Louisa F., *Fractal Image Compression*, Byte October 1993
- [BAR88] Barnsley, Micahel, *Fractals Everywhere*, Academic Press, 1988
- [BYT93] -----, *Fractal Compression Goes On-Line*, Byte September 1993
- [CLU98\_1] Cluts, Nancy Winnick, *Creating ActiveX Components in C++*, Microsoft Corporation, November 1998
- [CLU98\_2] Cluts, Nancy Winnick, *ActiveX Controls Overview*, Microsoft Corporation, October 1998
- [EDW98] Edwards, Michael, *Downloading Code on The Web*, Microsoft Corporation, April 1998
- [FIS94] Fisher, Yuval, *Fractal Image Compression: Theory and Application*, Springer-Verlag, 1994
- [HAK89] Hakim, Irfan, *Pemampatan Citra dengan Teknik Block Truncation Coding*, Tugas Akhir Teknik Informatika, 1989
- [JAC92] Jacquin, Arnaud E., *Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation*, IEEE Transaction on Image Processing, 1992
- [JOH96] Johns, Paul, *Signing and Marking ActiveX Controls*, Microsoft Corporation, October 1996

- [KAU96] Kauffman, Sanders & Jeff Perkins, *Teach Yourself ActiveX Programming in 21 Days*, SAMS.NET, 1996
- [KOM93] Kominek, John, *Understanding Fractal Image Compression*, University of Waterloo, 1993
- [KOM97] Kominek, John, *Advances in Fractal Compression for Multimedia Application*, Multimedia Systems Springer Verlag, 1997
- [KUR97] Kurniawan, Dios, *HTML 3 Untuk Publikasi di Internet*, BPFE Yogyakarta, 1982
- [MAN82] Mandelbrot, Benoit B., *The Fractal Geometry of Nature*, W.H. Freeman Company, 1982
- [NEL96] Nelson, Mark & Jean-Loup Gailly, *The Data Compression Book, Second Edition*, M & T Books
- [OLI92] Oliver, Dick, *Memandang Realita dengan Fractal Vision*, Penerbit ANDI Yogyakarta, 1992
- [PEI92] Peitgen, Jurgens Saupe, *Fractals for The Classroom*, Springer-Verlag, 1992
- [TAL96] Tall, Eric & Mark Ginsburg, *Late Night ActiveX*, Macmillan Computer Publishing USA, 1996