

KOMPRESI CITRA DIGITAL DENGAN FRAKTAL SEBAGAI TEKNIK KOMPRESI ALTERNATIF

Aris Sugiharto¹ dan Agus Harjoko²
1. Jurusan Matematika FMIPA UNDIP
2. Jurusan Fisika FMIPA UGM

Abstrak: Kompresi citra digital merupakan sebuah metode yang bertujuan untuk meminimalisir adanya redundansi. Beberapa metode kompresi telah dikembangkan antara lain adalah kompresi JPEG, wavelet, serta neural network. Sedangkan pada penelitian ini dikembangkan metode kompresi dengan menggunakan fraktal. Pada metode ini citra digital dibagi menjadi dua blok yaitu blok range yang berukuran 4x4, 8x8 atau 16x16 dan blok domain yang berukuran dua kali dari blok range. Dasar dari kompresi ini adalah mencari kemiripan bagian citra yang ada diantara blok range dengan blok domain dalam bentuk transformasi. Koefisien-koefisien transformasi inilah yang selanjutnya disimpan sebagai ukuran file hasil kompresi. Ukuran file kompresi fraktal pada umumnya sangat kecil akan tetapi kualitas citra yang dihasilkan memiliki nilai PSNR yang rendah dan waktu untuk kompresi yang relatif lama.

Kata Kunci: kompresi, fraktal, blok range, blok domain, PSNR

PENDAHULUAN

Citra digital saat ini banyak digunakan dalam berbagai bidang. Mulai dari keperluan sehari-hari seperti cetak foto, pemetaan hutan, identifikasi forensik, rekam medis dengan menggunakan citra kedokteran (*medical images*) sampai pada citra satelit. Hampir semua citra digital memerlukan media penyimpanan (*storage*) yang cukup besar. Sehingga hal ini menimbulkan masalah jika citra disimpan dalam database yang memiliki keterbatasan media penyimpanan. Masalah lain adalah jika diinginkan untuk mengirimkan citra digital dengan menggunakan jalur komunikasi atau internet. Dengan ukuran yang besar maka citra digital juga memerlukan waktu pengiriman yang lama. Sehingga diupayakan suatu teknik yang dapat mereduksi besarnya ukuran file citra digital dengan kompresi. Hampir semua teknik kompresi memiliki keuntungan seperti di atas yaitu dapat mereduksi ukuran file citra digital, sehingga dapat meminimalisir semua kendala di atas. Akan tetapi dibalik keuntungan ini ada sisi lain yang merugikan yaitu turunnya kualitas citra. Banyak metode kompresi yang sudah dikembangkan antara lain adalah JPEG dan JPEG 2000. Pada penelitian ini akan dikembangkan metode kompresi dengan fraktal sebagai teknik kompresi alternatif.

TINJAUAN PUSTAKA

Kompresi

Saat ini kebanyakan aplikasi menginginkan representasi citra digital dengan menggunakan kebutuhan memori yang seminimal mungkin. Kompresi citra (*images compression*) mempunyai tujuan meminimalkan kebutuhan memori untuk merepresentasikan sebuah citra digital. Prinsip umum yang digunakan pada proses kompresi citra digital adalah mengurangi duplikasi data di dalam citra sehingga memori yang dibutuhkan untuk merepresentasikan citra menjadi lebih sedikit dari pada citra digital aslinya. Terdapat dua proses utama dalam permasalahan kompresi citra digital.

a. Kompresi citra (*images compression*)

Pada proses ini citra digital dalam representasinya yang asli (belum dikompres) dikodekan dengan representasi yang meminimumkan kebutuhan memori. Citra dengan format bitmap pada umumnya tidak dalam bentuk kompresan. Citra yang sudah dikompres disimpan ke dalam arsip dengan menggunakan format tertentu.

b. Dekompresi citra (*images decompression*)

Pada proses dekompresi, citra yang sudah dikompresi harus dapat dikembalikan lagi menjadi representasi citra seperti citra aslinya. Proses ini diperlukan jika citra ingin ditampilkan ke layar atau disimpan ke dalam arsip dengan format yang tidak terkompres.

Beberapa Metode Kompresi

Kompresi JPEG

Teknik kompresi JPEG (Joint Photographic Expert Group) menggunakan transformasi DCT untuk merubah nilai – nilai pixel dalam domain spasial menjadi koefisien – koefisien dalam domain frekuensi. Dalam kompresi JPEG, mula – mula citra yang akan dikompresi dibagi menjadi blok-blok tertentu berukuran 8x8 yang selanjutnya ditransformasi dengan DCT untuk mendapatkan koefisien – koefisien dalam domain frekuensi [11]. Langkah selanjutnya adalah melakukan proses kuantisasi dan pengkodean dengan metode zig-zag menggunakan RLE, Huffman. Setelah itu dilakukan konversi vector kuantisasi menjadi bit-bit untuk mengkontruksi file JPEG.

Kompresi Wavelet

Selain JPEG teknik lain yang berkembang adalah kompresi wavelet dengan menggunakan transformasi wavelet DWT [1]. Transformasi DWT digunakan untuk mentransformasi citra menjadi beberapa koefisien wavelet. Selanjutnya koefisien wavelet ini ditanam kembali dengan menggunakan metode EZW [9].

Kompresi dengan Neural Network

Teknik kompresi lain yang sedang diteliti adalah teknik kompresi menggunakan neural network dimana pada teknik ini mula-mula citra ditransformasi dengan DWT atau WP (*Wavelet Packet*) sehingga menghasilkan koefisien-koefisien wavelet, selanjutnya koefisien ini direpresentasikan melalui neural network sehingga menghasilkan koefisien baru dengan space minimal melalui pelatihan – pelatihan neuron – neuron [5].

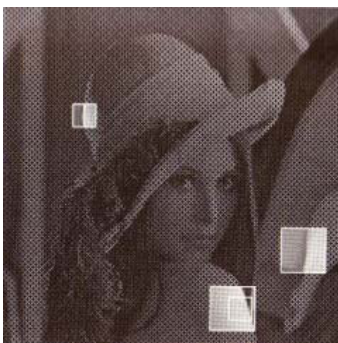
PEMBAHASAN

Menyimpan citra digital dengan menggunakan kumpulan pixel akan membutuhkan media penyimpanan (storage) yang sangat besar, namun apabila citra tersebut yang disimpan adalah transformasi affine-nya, maka kebutuhan tersebut akan dapat dikurangi secara signifikan. Dari latar belakang inilah muncul gagasan untuk mengkodekan citra dengan nisbah kompresi yang tinggi.

Kesulitan yang paling besar dalam upaya melakukan kompresi citra dengan menggunakan fraktal adalah menemukan bagian citra yang mirip dengan keseluruhan citra. Untuk itu diperlukan sebuah teknik khusus yang dapat digunakan untuk menemukan bagian-bagian yang mirip antara bagian citra dengan bagian yang lainnya.

PIFS (Partitioned Iterated Function System)

Citra secara alamiah umumnya hampir tidak pernah memiliki sifat self-similarity secara keseluruhan. Karena itu, citra alamiah secara umum juga tidak memiliki transformasi affine terhadap dirinya sendiri. Tetapi citra pada umumnya memiliki self-similarity lokal yaitu memiliki bagian – bagian citra yang mirip dengan bagian citra lainnya (Gambar 1).



Gambar 1. Self-similarity lokal
(Sumber : [3])

Secara umum kompresi menggunakan fraktal dilakukan dengan membagi citra asli menjadi beberapa blok yang tidak saling beririsan (*non overlapping*) yang dinamakan dengan blok range yang berukuran 4x4 atau 8x8 dan biasanya untuk mempermudah diambil blok yang berbentuk persegi. Selanjutnya dibuat juga beberapa blok domain yang ukurannya diambil 2 kali blok range [7]. Untuk blok domain yang diambil dapat beririsan ataupun tidak beririsan. Keuntungannya adalah jika digunakan yang tidak beririsan maka jumlah blok domainnya menjadi lebih sedikit dan waktu pencocokannya menjadi lebih cepat. Namun tentu saja hasilnya tidak

sebaik jika digunakan blok domain yang beririsan. Karena dengan menggunakan blok domain yang beririsan jumlah blok domainnya semakin banyak sehingga kemungkinan self-similarity lokalnya juga tinggi namun waktu pencocokannya menjadi lebih lama. Selanjutnya untuk setiap blok range dicari bagian citra yang memiliki kemiripan (self-similarity) tinggi dengan blok domain. Tingkat kemiripan ini diukur dengan menggunakan RMS [4].

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

Untuk blok range dengan ukuran 8 x 8, maka akan diambil blok domain dengan ukuran 16 x 16 pixel. Sehingga jika terdapat sebuah citra dengan ukuran 256 x 256 pixel, maka dapat dibagi menjadi $(256/8)^2 = 32^2 = 1024$ buah blok yang tidak saling beririsan. Sedangkan untuk blok domain yang dapat dibentuk adalah $(256 - 16 + 1)^2 = 58.081$ blok yang saling beririsan.

Sebelum setiap blok yang ada dalam blok range dicocokkan, maka setiap blok dalam blok domain harus terlebih dahulu diskalakan menjadi 1/2 bagian. Penskalaan ini dimaksudkan untuk menjaga agar jarak antara blok domain dan blok range menjadi lebih mudah dihitung. Penskalaan dapat dilakukan dengan cara menjadikan 2x2 buah pixel menjadi satu buah pixel. Nilai satu buah pixel tersebut adalah rata-rata dari nilai keempat pixel. Jika ditemukan blok range dan blok domain yang memiliki kemiripan tinggi maka dilakukan transformasi affine w_i untuk memetakan blok domain ke blok range. Transformasi affine yang digunakan adalah [4] :

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

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

$$z' = s_i z + o_i$$

Parameter s_i menyatakan faktor kontras pixel, jika s_i bernilai 0 maka pixel menjadi gelap dan jika s_i sama dengan 1 maka kontrasnya tidak mengalami perubahan, sedangkan jika bernilai antara 0 sampai 1 maka kontrasnya menjadi berkurang dan jika lebih besar dari 1 maka kontrasnya akan bertambah. Parameter o_i menyatakan offset kecerahan (*brightness*) pixel. Nilai o_i positif akan mencerahkan gambar dan jika nilai o_i negatif maka akan menjadi gelap. Kedua parameter s_i dan o_i dapat memetakan secara akurat blok domain yang berskala abu-abu ke blok range yang berskala abu-abu juga.

Untuk menjamin efek kontraktif dalam arah spasial, maka blok domain harus berukuran lebih besar dari pada blok range. Untuk alasan sederhana maka biasanya blok domain diambil 2 x blok range. Jadi jika terdapat sebuah citra dengan blok range yang berukuran $n \times n$ maka dapat diambil ukuran blok domainnya $2n \times 2n$ pixel. Perbandingan ini membuat persamaan transformasi affine menjadi lebih sederhana yaitu :

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

Parameter e_i dan f_i menyatakan pergeseran sudut kiri blok domain ke sudut kiri blok range yang bersesuaian. Sedangkan s_i dan o_i dihitung dengan menggunakan rumus regresi berikut :

$$E = \sum_{i=1}^n (d'_i - r_i)^2 = \sum_{i=1}^n (s_i d_i + o - r_i)^2$$

$$DRMS = \frac{\sqrt{E}}{n}$$

Selanjutnya transformasi affine w_i diuji terhadap blok domain D_i untuk menghasilkan blok uji $T_i = w_i(D_i)$. Jarak antara T dan R_i dihitung dengan DRMS. Transformasi affine yang terbaik adalah transformasi w yang meminimumkan jarak antara R_i dan T .

Runtutan pencarian dilanjutkan untuk blok-blok range berikutnya sampai semua blok range sudah dipasangkan dengan blok domain dengan transformasi affine-nya. Hasil dari proses kompresi adalah sejumlah IFS lokal yang dinamakan dengan PIFS. Seluruh parameter PIFS dipak dan disimpan ke dalam berkas eksternal. Parameter PIFS yang perlu disimpan hanyalah parameter e_i , f_i , s_i , o_i dan jenis operasi simetri terhadap setiap blok range. Dalam kenyataannya, parameter e_i dan f_i diganti dengan posisi blok domain yang dipetakan ke blok range. Sedangkan parameter a_i , b_i , c_i dan d_i tidak perlu disimpan karena nilainya sudah tetap yaitu 1/2 untuk a_i dan d_i serta 0 untuk b_i dan c_i .

Untuk merekonstruksi citra setelah dikompres, maka dilakukan proses iterasi PIFS dari citra awal sebarang. Karena IFS adalah lokal kontraktif baik kontraktif dalam domain intensitas maupun domain spasial maka iterasinya akan konvergen ke citra tetap PIFS. Kontraktif intensitas penting untuk menjamin konvergensi

ke citra semula (original). Sedangkan kontraktif spasial penting dalam menjamin rincian pada citra untuk setiap skala. Jika PIFS yang ditemukan selama proses kompresi baik, yaitu gabungan dari transformasi seluruh blok domain yang dekat dengan citra semula, maka titik tetap PIFS juga dekat dengan citra semula.

Selama proses pemulihan setiap IFS lokal mentransformasikan sekumpulan blok domain menjadi sekumpulan blok range. Karena blok range tidak saling beririsan dan mencakup seluruh pixel dalam citra, maka gabungan seluruh blok range akan menghasilkan citra titik tetap yang menyerupai citra semula.

Untuk mengetahui adanya perubahan kualitas citra sebagai akibat proses kompresi fraktal, digunakan ukuran PSNR (Peak Signal to Noise Ratio) [2] :

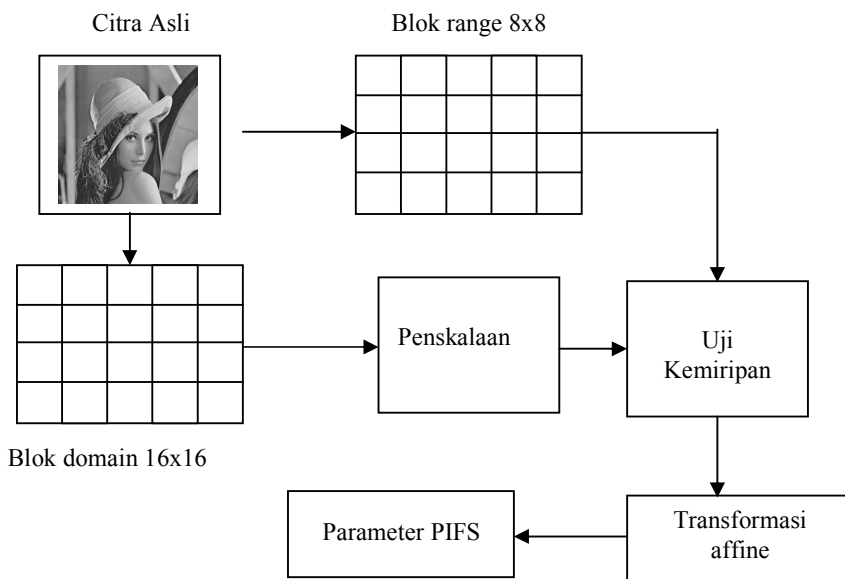
$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x'(i, j) - x(i, j)]^2$$

$$PSNR = 10^{10} \log \left[\frac{255}{\sqrt{MSE}} \right]$$

Metode Penelitian

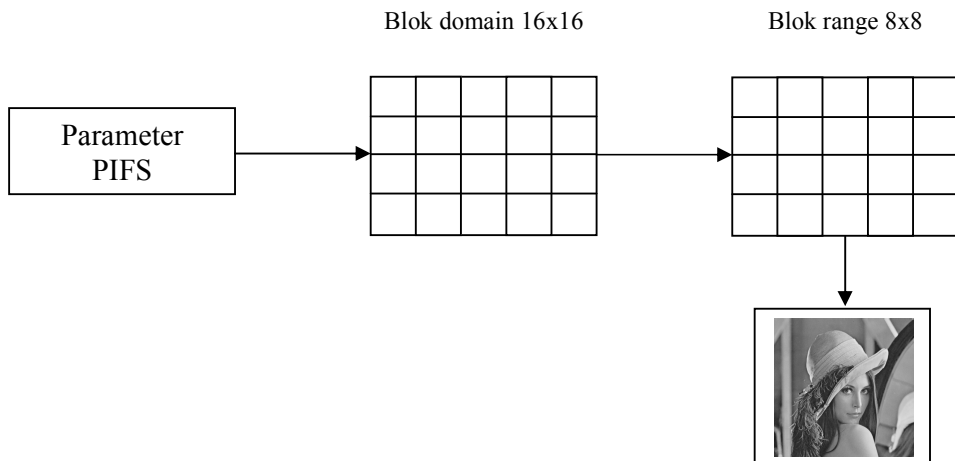
Metode yang digunakan dalam kompresi fraktal adalah sebagai berikut :

a. Tahapan proses kompresi



Gambar 3. Tahapan Proses Kompresi

b. Tahapan proses De-kompresi



Gambar 4. Tahapan Proses De-Kompresi

Adapun algoritma untuk kompresi citra digital dengan fraktal adalah :

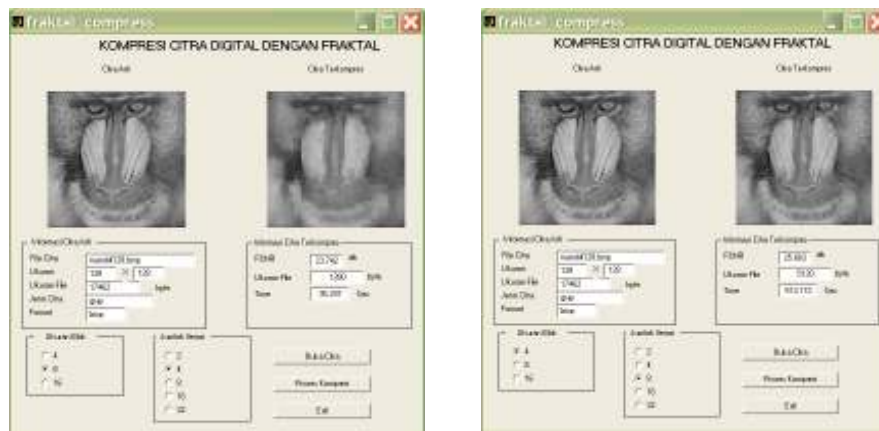
- a. Baca citra asli
- b. Menentukan ukuran matriks citra asli
- c. Menentukan ukuran blok range
- d. Menentukan ukuran blok domain
- e. Blok domain diskalakan ukurannya menjadi $\frac{1}{2}$ kali ukuran semula.
- f. Untuk setiap blok range
 - f.1 Dicari kemiripan antara blok range ke i dengan semua blok domain dengan menggunakan RMS.
 - f.2 Hitung transformasi affine untuk antara blok range ke i dengan blok domain yang terpilih.
 - f.3 Simpan koefisien transformasi affine ke i.
- g. Simpan semua parameter dalam PIFS

Sedangkan untuk algoritma de kompresi adalah :

- a. Baca citra asli (pembanding)
- b. Menentukan jumlah iterasi
- c. Mengambil semua parameter PIFS
- d. Untuk setiap blok range ke i
 - Transformasi affine dengan menggunakan parameter dari blok domain ke blok range ke i.
- e. Blok range hasil transformasi adalah citra hasil kompresi.

Simulasi dan Hasil

Simulasi kompresi citra digital dengan menggunakan fraktal dilakukan dengan menggunakan perangkat lunak Matlab 6.5. Adapun citra yang digunakan adalah mandril128.bmp yang berukuran 128 x 128.



(a)

(b)

Gambar 5. Simulasi kompresi dengan menggunakan citra Mandril (a). blok range 8x8 dan jumlah iterasi 4, (b) blok range 4x4 dan jumlah iterasi 8.

Setelah dilakukan beberapa simulasi diperoleh hasil sebagai berikut :

Tabel 1. Kompresi Fraktal Citra Mandril 128x128 dengan blok range 4x4

NO	Iterasi	Time (sec)	PSNR(dB)	Size (byte)
1.	2	515.922	25.072	5120
2.	4	522.594	25.759	5120
3.	8	513.172	25.803	5120
4.	16	509.688	25.800	5120
5.	32	524.281	25.789	5120

Tabel 2. Kompresi Fraktal Citra Mandril 128x128 dengan blok range 8x8

NO	Iterasi	Time (sec)	PSNR(dB)	Size (byte)
1.	2	37.813	22.962	1280
2.	4	35.297	23.742	1280
3.	8	36.625	23.701	1280
4.	16	37.468	23.701	1280
5.	32	38.047	23.701	1280

Tabel 3. Kompresi Fraktal Citra Mandril 128x128 dengan blok range 16x16

NO	Iterasi	Time (sec)	PSNR(dB)	Size (byte)
1.	2	2.672	20.912	320
2.	4	2.843	21.115	320
3.	8	2.859	21.11	320
4.	16	3.047	21.110	320
5.	32	3.407	21.110	320

KESIMPULAN

Dari uraian sebelumnya dapat disimpulkan bahwa teknik kompresi dengan fraktal menghasilkan ukuran file yang relatif kecil akan tetapi terjadi penurunan kualitas citra yang signifikan. Hal ini dapat dilihat dari nilai PSNR yang berkisar 20.912 hingga 25.803. Selain itu dengan kompresi fraktal waktu yang dibutuhkan untuk proses kompresi relatif lama berkisar 2.672 sampai 524.281 detik.

UCAPAN TERIMA KASIH

Terima kasih kepada Program Hibah Kompetisi A2 jurusan matematika FMIPA UNDIP tahun 2006 yang telah mendanai program magang penelitian ini.

DAFTAR PUSTAKA

- [1] Antonini, M., et. al., Images Coding Using Wavelet Transform, *IEEE Trans. On Image Processing*, Vol. 1, pp. 205 – 220, 1992.
- [2] Chen C., *On The Selection of Image Compression Algorithms*, NSC Grant Departemen of Computer Science National Tsing Hua University, Taiwan, 1998.
- [3] Fisher, Y., *Fractal Image Compression : Theory and Application* , Spinger – Verlag, 1994.
- [4] Galabov, M., Fractal Image Compression, *International Confrence On Computer Science System and Technologies – CompSysTech*, University Of Veliko Turnovo, 2003.
- [5] Gillespie, W., *Still Image Compression Using Neural Networks*, Utah State University, Logan Utah, 2003.
- [6] Gonzalez , R.C. and Woods, R.E., *Digital Image Processing*, Addison Wesley Publishing, 1993.
- [7] Khalid Kamali, *Fractal Video Compression*, Bachelor of Engineering (Computer Science) Faculty of Engineering and Surveying University of Southern Queensland, 2005.
- [8] Munir R, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Penerbit Informatika, Bandung, 2004.
- [9] Shapiro, J.M., Embedded Image Coding Using Zerotree of Wavelet Coefficient, *IEEE Trans. On Image Processing*, Vol. 41, pp. 3445 – 3462, 1993.
- [10] Sugiharto Aris, *Pemrograman GUI dengan MATLAB*, Penerbit Andi Yogyakarta, 2006.
- [11] Wallace C.J., The JPEG Still Picture Compression Standart, *Communication ACM*, Vol. 34, pp. 31 – 44, 2005.

