

# Implementasi Sistem Temu Kembali Citra Berdasarkan Histogram Parameter Fraktal

Rully Soelaiman\*, Irfan Subakti\* dan Galih Satria\* \*

\* Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 60111, Indonesia  
Email: rully130270@gmail.com, yifana@gmail.com

## ABSTRAK

Sistem temu kembali citra berdasarkan parameter fraktal adalah sebuah sistem untuk mencari citra pada basis data yang mirip dengan citra query dengan menggunakan parameter fraktal sebagai ciri. Algoritme pencarian dengan menggunakan histogram grayscale tidak dapat membedakan citra yang secara visual memiliki tekstur yang berbeda. Oleh karena itu, pada penelitian berikut dikembangkan sistem temu kembali citra berdasarkan metode fraktal.

Citra dikodekan dengan menggunakan metode fraktal yang akan menghasilkan parameter fraktal. Dua parameter fraktal (yaitu skala kontras dan range block mean) dijadikan dasar untuk membuat empat histogram vektor ciri yang dimanfaatkan sebagai indeks. Keempat indeks tersebut adalah (1) range block mean terkuantisasi, (2) gabungan range block mean dan skala kontras terkuantisasi, (3) gabungan 2D antara range block mean dan skala kontras terkuantisasi, dan (4) gabungan perkalian tensor antara range block mean dan skala kontras terkuantisasi. Pengukuran kemiripan dihitung berdasarkan empat histogram tadi dengan menggunakan metrik  $L_p$ -norm orde pertama.

Uji coba dilakukan pada 368 citra yang berasal dari 23 citra brodatz yang dipotong dalam ukuran 128 x 128 piksel. Hasil uji coba ini menunjukkan bahwa nilai average retrieval rate terbaik sebesar 70,3% dengan mengambil empat citra yang paling mirip dengan citra query dengan menggunakan indeks-1 sebagai ciri citra.

**Kata Kunci:** Sistem temu kembali citra, pengkodean fraktal, indeks citra.

## 1. PENDAHULUAN

Sistem temu kembali citra pada basis data masih menggunakan query tekstual sehingga setiap citra pada basis data harus diberikan keterangan teks yang digunakan sebagai

indeks. Cara query secara tekstual tentu saja tidak efektif dan memiliki akurasi yang rendah.

Beberapa algoritme penemuan kembali citra dikembangkan sehingga memungkinkan pengguna melakukan query dengan menggunakan citra untuk mencari citra pada basis data yang memiliki kemiripan dengan citra yang di-query. Salah satu algoritme yang dikembangkan adalah menggunakan histogram nilai piksel grayscale sebagai vektor ciri. Namun demikian seperti ditunjukkan oleh gambar 1 dan gambar 2, gambar tersebut secara visual berbeda namun memiliki histogram nilai piksel grayscale yang sama.

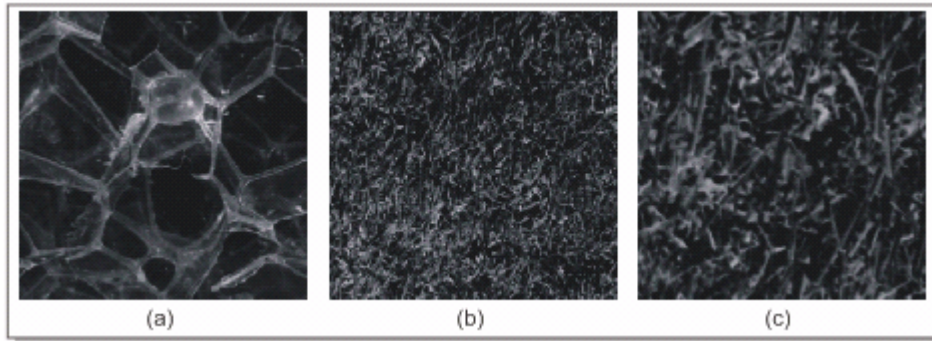
Parameter fraktal hasil pengkodean citra dengan metode fraktal memiliki independensi terhadap nilai piksel grayscale sehingga cukup kokoh terhadap masalah tersebut. Seperti yang ditunjukkan oleh gambar 3, histogram range block mean dapat membedakan citra brodatz [1] pada gambar 1. Berdasarkan sifat tersebut, parameter fraktal diajukan sebagai vektor ciri dalam sistem temu kembali citra yang menggunakan citra sebagai query [3].

## 2. PENGKODEAN CITRA DENGAN METODE FRAKTAL

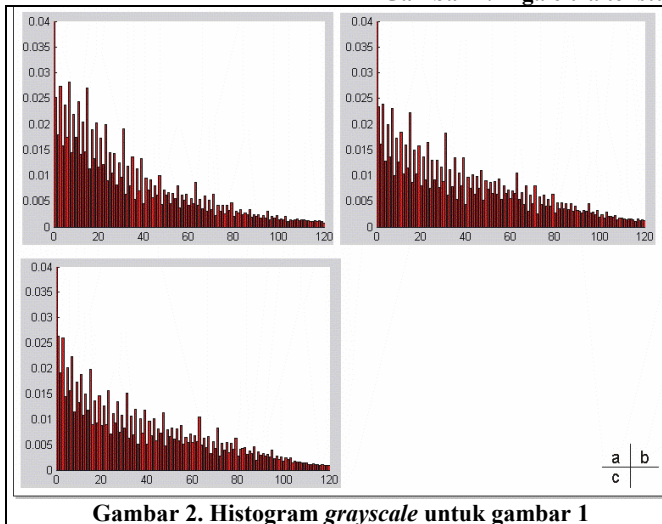
Pada pengkodean citra dengan metode fraktal, ada beberapa tahap yang dilakukan untuk mendapatkan parameter fraktal yang akan digunakan sebagai ciri, yaitu proses pembangunan blok domain, partisi quadtree, dan pencarian blok range yang kontraktif terhadap blok domain dengan menggunakan jarak Root Mean Square (RMS).

### a) Pembangunan Blok Domain

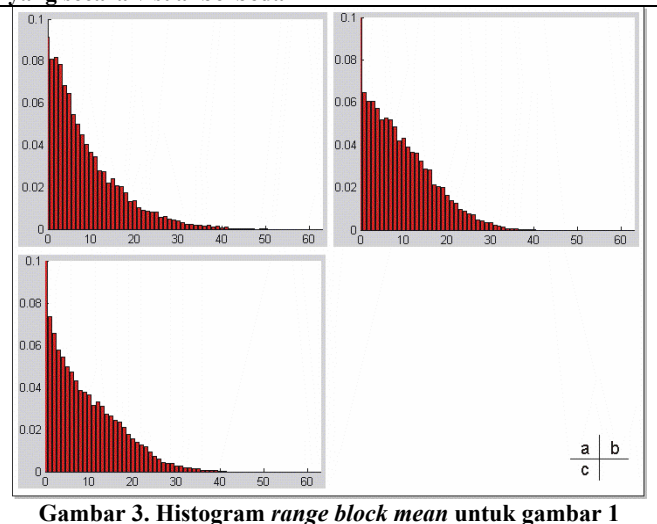
Blok domain dibangun dengan cara citra dipartisi menjadi subcitra dalam berbagai ukuran mulai dari 16x16 piksel, 8x8 piksel, 4x4 piksel, hingga 2x2 piksel yang dinamakan dengan blok domain. Partisi dilakukan mulai dari pojok kiri atas citra bergeser hingga ke pojok kanan bawah dan memungkinkan terjadinya overlapping.



Gambar 1. Tiga citra tekstur yang secara visual berbeda



Gambar 2. Histogram grayscale untuk gambar 1



Gambar 3. Histogram range block mean untuk gambar 1

Blok domain merupakan bentuk dasar dari citra yang dikodekan. Dari bentuk-bentuk dasar inilah dibangun citra dengan melalui transformasi (rotating dan flipping).

Blok domain dibangun pada awal proses pengkodean fraktal untuk mencegah penghitungan properti-properti blok domain (misalnya jumlah piksel pada blok, jumlah kuadrat, varians, dan lain-lain) secara berulang pada saat proses quadtree. Untuk mencegah duplikasi perhitungan, citra didesimasi dengan menggunakan rumus berikut ini:

$$a + b + \frac{1}{4} \times (c + d) \quad (1)$$

dimana  $a$  adalah kuadran piksel bagian kiri atas,  $b$  adalah kuadran piksel bagian kanan atas,  $c$  adalah kuadran piksel bagian kiri bawah, dan  $d$  adalah kuadran piksel bagian kanan bawah. Untuk mengetahui transformasi yang akan digunakan blok domain pada saat *decoding*, blok domain diklasifikasikan berdasarkan nilai piksel dan tingkat kecerahan pada blok tersebut. Terdapat tiga kelas dengan 24 subkelas untuk masing-masing kelas.

Tiap-tiap kelas memiliki cara transformasi dalam urutan yang berbeda, namun jenis transformasi yang dilakukan

hanya ada dua, yaitu rotasi dan pelipatan horizontal (*horizontal flipping*) [2].

#### b) Partisi Quadtree

Untuk mendapatkan bagian-bagian citra yang kontraktif terhadap blok domain, maka citra dipotong menjadi subcitra yang dinamakan blok *range* dengan menggunakan partisi *quadtree*. Partisi ini membagi citra menjadi empat bagian yang sama besar dan secara rekursif setiap bagian tersebut dibagi lagi menjadi empat bagian dan seterusnya hingga mencapai ukuran tertentu.

Ketika subcitra sudah mencapai ukuran tertentu, misalnya  $B \times B$  piksel, subcitra dibandingkan dengan blok domain yang memiliki ukuran  $2B \times 2B$  piksel yang memiliki kelas dan subkelas yang sama. Antara blok *range* dan blok domain diukur jarak RMS-nya.

Pengukuran jarak RMS pada dasarnya melakukan minimalisasi persamaan 2 berikut ini:

$$E(R, D) = \sum (s * d_i + g - r_i)^2 \quad (2)$$

dimana  $R$  adalah blok *range* yang beranggotakan  $r_i$ ,  $D$  adalah blok domain yang beranggotakan  $d_i$ ,  $s$  adalah skala kontras dan  $g$  adalah tingkat kecerahan. Minimalisasi persamaan 2 akan terjadi jika skala kontras ( $s$ ) dan tingkat kecerahan ( $g$ ) memenuhi persamaan 3 dan 4.

Pasangan blok *range* dan blok domain yang memiliki jarak RMS terkecil disimpan pada media penyimpanan sebagai parameter fraktal. Parameter fraktal yang didapatkan yaitu (1) skala kontras, (2) tingkat kecerahan, (3) nilai rata-rata piksel blok *range* (*range block mean*), (4) posisi koordinat  $x$  pada blok domain, dan (5) posisi koordinat  $y$  pada blok domain [2].

$$s = \frac{n^2 \left( \sum_{i=1}^n d_i r_i \right) - \left( \sum_{i=1}^n d_i \right) \left( \sum_{i=1}^n r_i \right)}{n^2 \sum_{i=1}^n d_i^2 - \left( \sum_{i=1}^n d_i \right)^2} \quad (3)$$

$$g = \frac{\sum_{i=1}^n r_i - s \sum_{i=1}^n d_i}{n^2} \quad (4)$$

Notasi pada persamaan 3 dan 4 dijelaskan sebagai berikut:  $r_i$  adalah elemen-elemen blok *range* jika blok *range* adalah  $R = \{ r_1, r_2, \dots, r_n \}$ ,  $d_i$  adalah elemen-elemen blok domain jika blok domain adalah  $D = \{ d_1, d_2, \dots, d_n \}$ , dan  $n$  adalah jumlah intensitas piksel yang ada pada blok yang sedang diperiksa.

Jika kondisi minimal di atas terpenuhi, maka jarak RMS dapat dihitung melalui persamaan 5 berikut ini:

$$RMS = \frac{\sum_{i=1}^n r_i^2 + s \left( s \sum_{i=1}^n d_i^2 - 2 \left( \sum_{i=1}^n d_i r_i \right) + 2g \sum_{i=1}^n d_i \right) + g \left( gn^2 - 2 \sum_{i=1}^n r_i \right)}{n^2} \quad (5)$$

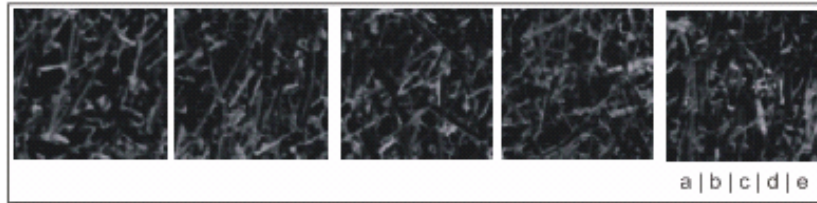
### 3. PEMBANGUNAN KEMBALI CITRA DARI KODE FRAKTAL

Proses pembangunan kembali citra bertujuan untuk memastikan bahwa kode fraktal yang disimpan telah benar sehingga layak untuk dijadikan ciri citra. Proses ini merupakan kebalikan dari proses pengkodean. Pada dasarnya, pasangan blok domain dan blok *range* yang telah disimpan dibaca kembali dan berdasarkan kelas dan subkelasnya, blok domain ditransformasikan agar mendapatkan posisi semula. Penghitungan ini diiterasikan dalam beberapa perulangan. Hasil penghitungan pada iterasi ke- $i$  akan dijadikan dasar untuk membangun citra pada iterasi ke- $(i+1)$  sehingga semakin banyak iterasi yang dilakukan, citra yang dihasilkan akan semakin halus.

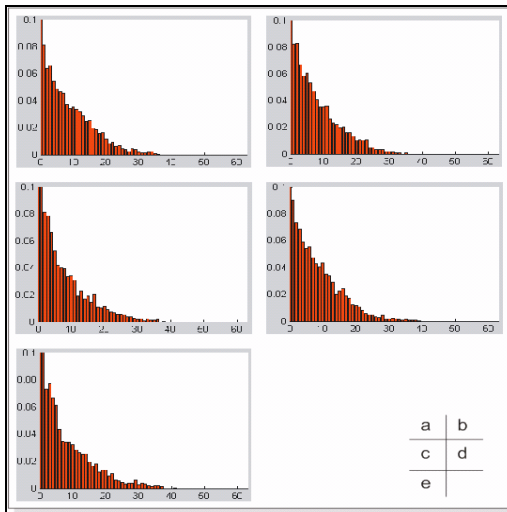
Jika nilai rata-rata piksel citra hasil iterasi sebelumnya dilambangkan dengan  $rp$ , skala kekontrasan  $s$  dan tingkat kecerahan  $g$ , maka nilai piksel untuk citra yang baru hasil iterasi saat ini dihitung dengan rumus yang dijelaskan pada persamaan 6 berikut ini:

$$image[x][y] = bound \left( \frac{1}{2} + (s \times rp) + g \right) \quad (6)$$

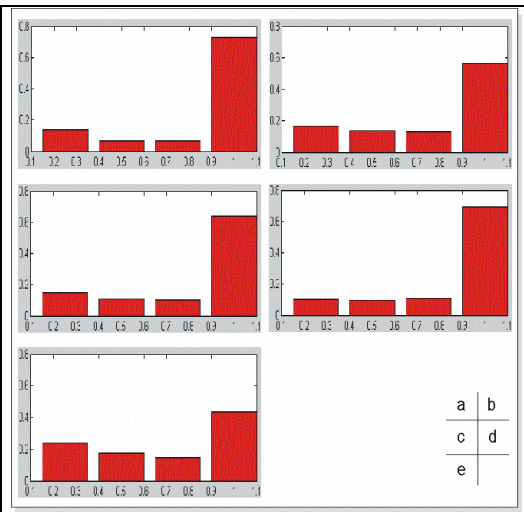
dimana  $image[x][y]$  adalah nilai piksel yang akan ditempatkan pada citra koordinat  $x, y$ . Bound adalah fungsi pembatas hasil perhitungan sehingga nilai hasil di antara 0 – 255.



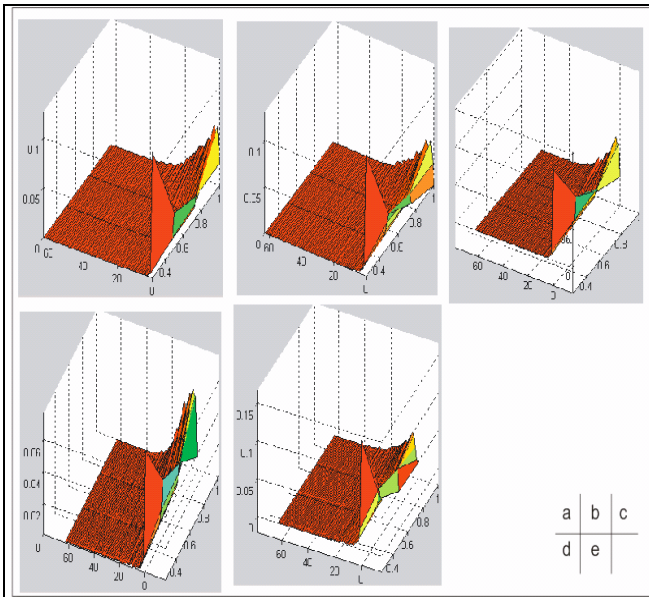
Gambar 4. Tekstur brodatz yang memiliki kemiripan dekat



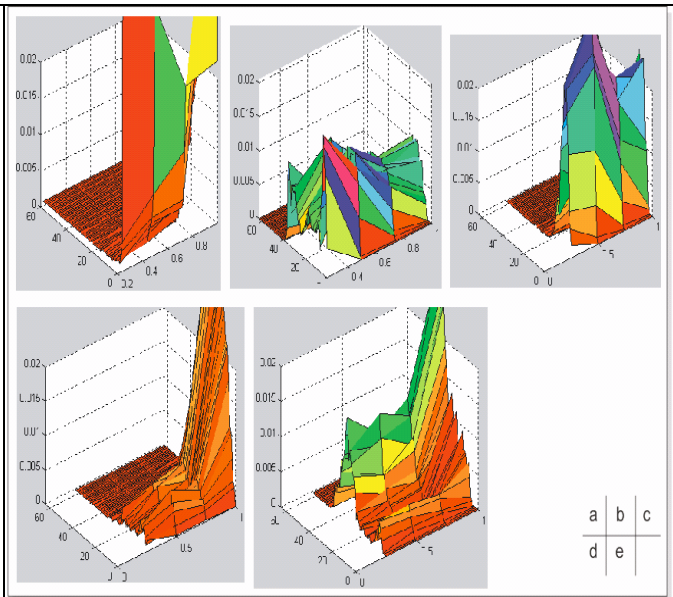
Gambar 5. Histogram range block mean



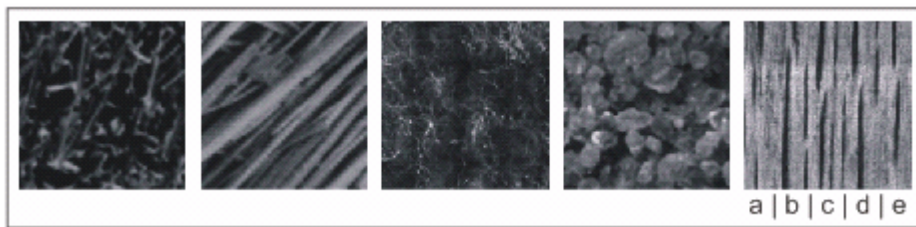
Gambar 6. Histogram skala kontras



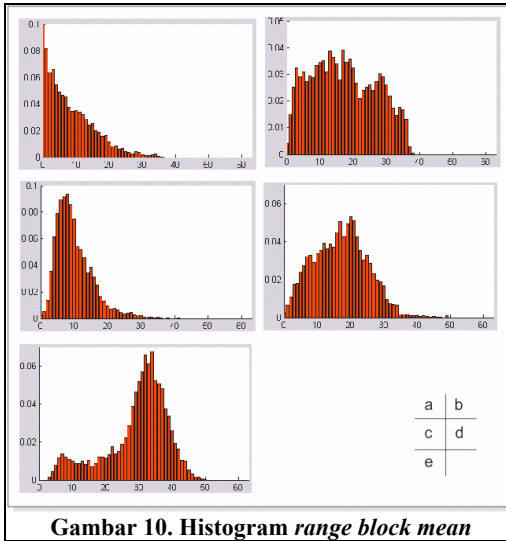
Gambar 7. Histogram indeks-3 untuk gambar 4



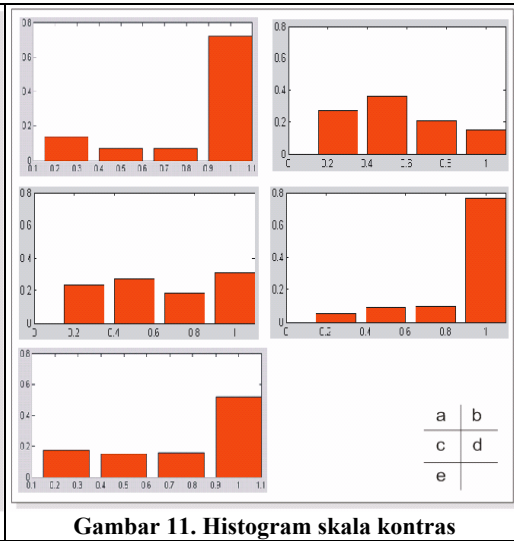
Gambar 8. Histogram indeks-3 untuk gambar 9



Gambar 9. Tekstur brodatz yang memiliki kemiripan jauh



Gambar 10. Histogram range block mean



Gambar 11. Histogram skala kontras

#### 4. PEMANFAATAN PARAMETER FRAKTAL SEBAGAI INDEKS

Dari lima parameter fraktal yang tersimpan pada media penyimpanan, skala kontras dan *range block mean* independen [3]. Dua parameter fraktal inilah yang dijadikan dasar untuk membangun histogram vektor ciri. Ada empat histogram yang diajukan yaitu (1) *range block mean* terkuantisasi, (2) gabungan *range block mean* dan skala kontras terkuantisasi, (3) gabungan 2D antara *range block mean* dan skala kontras terkuantisasi, dan (4) gabungan perkalian tensor antara *range block mean* dan skala kontras terkuantisasi.

##### a) Indeks-1: Histogram Range Block Mean

Indeks pertama didapatkan dari nilai rata-rata piksel pada blok *range* yang telah mengalami proses kuantisasi. Kuantisasi adalah proses normalisasi suatu rentang nilai ke dalam rentang nilai baru. Nilai piksel pada blok *range* berkisar pada rentang nilai 0 – 255. Kuantisasi yang dilakukan adalah mengubah rentang nilai ini menjadi 6 bit atau 7 bit (rentang 0 – 63 atau 0 – 127). Misalnya untuk kuantisasi 6 bit, untuk nilai piksel antara 0 – 3 akan masuk kelas kuantisasi ke-0, nilai piksel antara 4 – 7 akan masuk kelas kuantisasi ke-1 dan seterusnya.

Nilai histogram yang diukur (sebagai sumbu-y) adalah nilai probabilitas dari setiap kelas kuantisasi. Seberapa banyak kemungkinan kejadian suatu kelas kuantisasi pada citra tertentu. Sehingga jumlah kemunculan pada setiap kelas kuantisasi dibagi dengan jumlah pasangan blok domain dan blok *range* yang tersimpan.

Untuk citra yang memiliki kemiripan dekat pada gambar 3, histogram indeks-1 memiliki tampilan seperti yang ditunjukkan oleh gambar 5.

##### b) Indeks-2: Histogram Gabungan Range Block Mean dan Skala Kontras Terkuantisasi

Indeks-2 merupakan gabungan antara indeks-1 dengan skala kontras yang terkuantisasi menjadi 2 bit (4 kelas) atau 3 bit (8 kelas). Sama halnya dengan kuantisasi indeks-1, nilai histogram yang diukur adalah besar probabilitas dari setiap kelas kuantisasi. Untuk gambar 4, histogram skala kontrasnya ditunjukkan oleh gambar 6.

Besar pengaruh indeks-1 dan skala kontras dalam indeks-2 ditentukan oleh bobot  $w$  yang ditunjukkan oleh persamaan 7 berikut ini:

$$w \{p(\bar{r})\}_{i=1}^I + (1 - w) \{v_{1,j}\}_{j=1}^J \quad (7)$$

dimana  $p(\bar{r})_{i=1}^I$  merupakan probabilitas *range block mean* terkuantisasi dan  $\{v_{1,j}\}_{j=1}^J$  adalah probabilitas skala kontras yang terkuantisasi. Nilai konstanta  $w$  berkisar antara 0 dan 1. Untuk citra berukuran 128x128 piksel, nilai  $w$  yang terbaik adalah 0.9 [3].

##### c) Indeks-3: Histogram Gabungan 2D Antara Range Block Mean dan Skala Kontras Terkuantisasi

Indeks-3 adalah kombinasi dari *range block mean* terkuantisasi dan skala kontras yang terkuantisasi yang penggabungannya berdasarkan notasi berikut ini:

$$\{q(\bar{r}_i, s_j)\} \quad (8)$$

dimana ( $i = 1, 2, \dots, I; j = 1, 2, 3, \dots, J$ ). Setiap kuantisasi piksel blok *range* akan dipasangkan dengan setiap kuantisasi skala kontras. Jika suatu *node* pasangan terbaik blok domain dan blok *range* yang memiliki skala kontras dan nilai piksel rata-rata yang sesuai, maka kelas histogram ditambahkan probabilitasnya. Probabilitas dihitung dengan membagi jumlah nilai histogram pada setiap kelas histogram dengan jumlah data. Untuk kuantisasi skala kontras dan *range block mean* 2-6 bit, jumlah kelas histogram adalah 256 kelas.

Karena histogram indeks-3 adalah gabungan 2D dari dua parameter fraktal, maka histogram yang terbentuk adalah dalam bentuk tiga dimensi seperti pada gambar 7.

**d) Indeks-4: Histogram Gabungan Perkalian Tensor Antara Range Block Mean dan Skala Kontras Terkuantisasi**

Indeks-4 hampir sama dengan indeks-3 dimana parameter skala kontras dan nilai rata-rata piksel digabungkan. Setiap elemen kelas kuantisasi pada *range block mean* jumlah probabilitasnya dikalikan dengan jumlah probabilitas pada setiap elemen pada kelas kuantisasi skala kontras. Notasi untuk indeks-4 ditunjukkan pada notasi berikut ini:

$$\{p(\bar{r}_i)\}_{i=1}^I \otimes \{v_{1j}\}_{j=1}^J = \{p(\bar{r}_i) * v_{1j}\} \quad (9)$$

dimana ( $i = 1, 2, \dots, I; j = 1, 2, 3, \dots, J$ ) dan  $p(\bar{r}_i)$  merupakan probabilitas *range block mean* terkuantisasi serta  $\{v_{1j}\}_{j=1}^J$  adalah probabilitas skala kontras yang terkuantisasi.

Gambar 9 adalah citra brodatz [1] yang memiliki jarak kemiripan jauh. Histogram indeks-1, skala kontras, dan indeks-3 untuk gambar ini berturut-turut ditunjukkan oleh gambar 10, gambar 11, dan gambar 8.

**5. PENGUKURAN JARAK KEMIRIPAN CITRA**

Untuk mengetahui jarak kemiripan citra *query* dengan citra pada basis data, digunakan pengukuran histogram indeks citra dengan metrik  $L_p$ -norm orde pertama yang dinotasikan pada persamaan berikut ini:

$$d_{Lp}(Q, C) = \sqrt[p]{\frac{1}{V} \|f_Q(\cdot) - f_C(\cdot)\|^p} = \frac{1}{V} \sum_{n=1}^V |\lambda_n - \gamma_n| \quad (10)$$

dimana  $d_{Lp}$  adalah jarak  $L_p$ -norm,  $p$  adalah orde jarak  $L_p$ -norm,  $Q$  adalah citra *query*,  $C$  adalah citra pada basis data,  $V$  adalah panjang vektor ciri (jumlah elemen dalam histogram),  $\lambda_n$  adalah elemen-elemen vektor ciri citra *query*, dan  $\gamma_n$  adalah elemen-elemen vektor ciri citra pada basis data yang sedang dibandingkan.

Jarak untuk citra brodatz [1] pada gambar 4 dan gambar 9 ditunjukkan masing-masing oleh tabel 1 dan tabel 2. Pada tabel tersebut terlihat bahwa jarak untuk citra yang memiliki kemiripan dekat (gambar 4) memiliki jarak yang kecil sedangkan citra yang memiliki kemiripan jauh (gambar 9) memiliki jarak yang besar.

**Tabel 1. Jarak histogram parameter fraktal indeks-1 dan indeks-3 untuk citra tekstur pada gambar 4**

Indeks	$d_{L_1}(a, b)$	$d_{L_1}(a, c)$	$d_{L_1}(a, d)$	$d_{L_1}(a, e)$
Ind-1	0.0019	0.0024	0.0024	0.0026
Ind-3	0.0014	0.00091	0.00093	0.0025

**Tabel 2. Jarak histogram parameter fraktal indeks-1 dan indeks-3 untuk citra tekstur pada gambar 9**

Indeks	$d_{L_1}(a, b)$	$d_{L_1}(a, c)$	$d_{L_1}(a, d)$	$d_{L_1}(a, e)$
Ind-1	0.0134	0.0118	0.0151	0.0242
Ind-3	0.0061	0.0049	0.0041	0.0063

Ada dua strategi pengambilan citra berdasarkan jarak-jarak dari pengukuran histogram vektor ciri yaitu dengan menggunakan nilai *threshold* dan langsung melakukan pengurutan jarak.

Strategi pertama dilakukan dengan menentukan suatu nilai *threshold* yang akan diterapkan pada indeks-1 atau indeks-2. Citra yang memiliki jarak lebih besar daripada nilai *threshold* langsung dibuang. Citra yang tersisa diurutkan berdasarkan indeks-3 atau indeks-4 mulai dari citra yang memiliki jarak paling dekat hingga jarak terjauh. Kemudian empat citra dengan jarak terdekat diambil dan ditampilkan sebagai citra yang dianggap mirip.

Strategi kedua dilakukan dengan langsung mengurutkan semua citra pada basis data mulai jarak terdekat hingga jarak terjauh berdasarkan salah satu dari keempat indeks citra.

**6. UJI KEBENARAN KODE FRAKTAL**

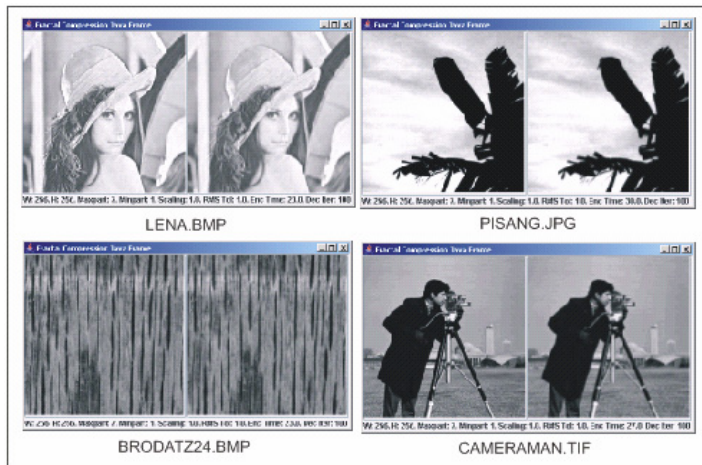
Kebenaran kode fraktal untuk menunjukkan bahwa parameter fraktal yang diambil layak untuk dijadikan ciri citra diuji dengan cara membuat modul *decoder* yang mengembalikan kode fraktal pada media penyimpanan menjadi citra kembali. Kode fraktal dikatakan benar jika nilai Peak Signal to Noise Ratio (PSNR) telah memenuhi syarat. PSNR sendiri dinotasikan dalam persamaan 11 dan persamaan 12 berikut ini:

$$MSE = \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \quad (11)$$

$$PSNR = 20 \log_{10} \left( \frac{255}{MSE} \right) \quad (12)$$

dimana  $f(i,j)$  adalah citra asli dan  $F(i,j)$  adalah citra hasil dengan ukuran  $N \times N$  piksel.  $MSE$  adalah *mean squared error* atau tingkat kesalahan yang terjadi pada citra hasil. Citra yang dikatakan baik adalah citra yang memiliki PSNR antara 20 dB hingga 40 dB.

Uji coba dilakukan dengan mengkodekan citra yang berukuran 256x256 piksel dan mengembalikannya kembali dengan mengamati nilai PSNR-nya. Iterasi yang dilakukan sebanyak 100 kali. Gambar 12 berikut adalah hasil-hasil pembangunan citra dari kode fraktal yang telah dikodekan dari citra asli menurut algoritme yang telah dijelaskan pada bagian 2. Sedangkan nilai PSNR-nya ditunjukkan oleh tabel 3. Sisi citra pada setiap *window* pada sebelah kiri adalah citra asli sedangkan sisi sebelah kanan adalah citra yang telah melalui proses *decoding*.



Gambar 12. Empat citra uji berukuran 256x256 piksel

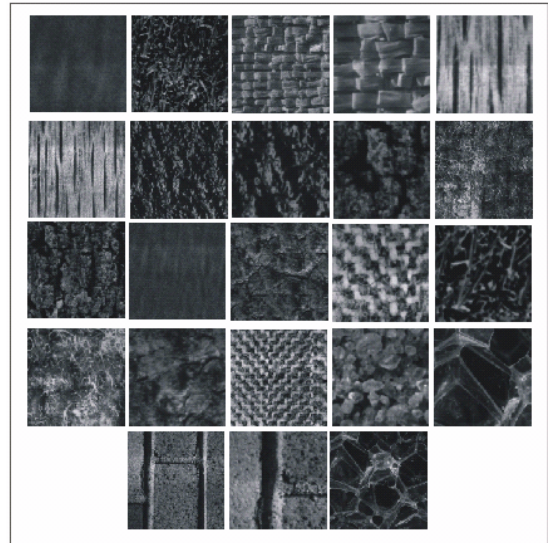
Tabel 3. PSNR citra hasil pengkodean yang ditunjukkan oleh gambar 12

Nama Citra	PSNR
LENA.BMP	30.107 dB
PISANG.JPG	33.229 dB
BRODATZ24.BMP	28.077 dB
CAMERAMAN.TIF	26.949 dB

## 7. UJI COBA DAN EVALUASI SISTEM TEMU KEMBALI CITRA

Uji coba dilakukan dengan menggunakan 23 tekstur brodatz [1] yang dipotong tanpa mengalami *overlapping* dalam ukuran 128x128 piksel. Dengan demikian total jumlah citra yang diobservasi sebanyak 368 buah.

Citra brodatz dipilih sebagai media uji coba karena citra brodatz memiliki detail tekstur yang rumit dan bervariasi. Karena parameter fraktal sebagai ciri mengenali citra dari bentuk teksturnya, citra brodatz sangat tepat dijadikan media evaluasi. Jika sistem telah lolos uji coba dengan citra brodatz, maka dapat dipastikan sistem telah kokoh dan tahan (*robust*) untuk pengenalan citra biasa, misalnya gambar orang, pemandangan, dan lain-lain. Bentuk tekstur citra brodatz tersebut ditunjukkan oleh gambar berikut:



Gambar 13. Bentuk tekstur Brodatz

Pengambilan citra dianggap benar jika hasil yang dikembalikan oleh sistem berasal dari sumber citra brodatz yang sama dengan sumber citra *query*. Misalnya citra *query* adalah potongan citra dari file brodatz **image01.bmp**, maka hasil pengembalian yang benar juga berasal dari file brodatz **image01.bmp**.

Untuk mendapatkan angka rata-rata penemuan citra yang benar, digunakan rumus berikut ini:

$$Avg = \frac{\sum_{z=1}^Z m_z}{F \times Z} \quad (13)$$

dimana  $Avg$  adalah *average retrieval rate* – ukuran rata-rata pengambilan citra yang benar,  $m_z$  adalah jumlah citra yang

ditemukan dengan benar pada setiap query ke- $z$ ,  $Z$  adalah jumlah citra keseluruhan, dan  $F$  adalah jumlah citra pada basis data yang diambil program untuk ditampilkan dimana citra tersebut dianggap program sebagai citra yang benar.

Ada enam skenario yang digunakan dalam uji coba ini yang bertujuan untuk mengamati faktor-faktor berikut: (1) kualitas akurasi pengambilan citra antara pasangan indeks-1/indeks-3 dengan pasangan indeks-2/indeks-4, (2) pengaruh perubahan kuantisasi parameter fraktal, (3) pencarian konfigurasi parameter terbaik untuk indeks-1, dan (4) pengaruh kuantisasi pada skala kontras jika indeks yang digunakan adalah indeks-1. Ringkasan pengaturan konfigurasi dan hasil uji coba ditunjukkan oleh tabel 4.

**a) Skenario-1 dan Skenario-2**

Skenario-1 dan skenario-2 bertujuan untuk mendapatkan pasangan indeks terbaik antara pasangan indeks-1/indeks-3 dengan pasangan indeks-2/indeks-4. Skenario-1 dan skenario-2 menggunakan nilai *threshold* 0.0087.

Dari hasil uji coba skenario-1 dan skenario-2, terlihat bahwa pasangan indeks-1/indeks-3 memiliki hasil yang lebih baik daripada pasangan indeks-2/indeks-4 dengan hasil 60.39%.

**b) Skenario-3 dan Skenario-4**

Dua skenario ini bertujuan untuk melihat pengaruh perubahan kuantisasi parameter skala kontras terhadap hasil

pengambilan citra. Skenario-3 dan skenario-4 menggunakan kelas kuantisasi skala kontras sebanyak 8 kelas sedangkan kuantisasi untuk *range block mean* adalah 64 kelas.

Dari hasil uji coba, terlihat bahwa hasil skenario-3 dan skenario-4 yang menggunakan kuantisasi 3-6 bit memiliki tingkat kebenaran yang lebih rendah daripada skenario-1 dan skenario-2 yang menggunakan kuantisasi 2-6 bit. Dengan demikian dapat disimpulkan bahwa kuantisasi 2-6 bit lebih baik daripada kuantisasi 3-6 bit.

**c) Skenario-5 dan Skenario-6**

Skenario-5 dan skenario-6 menggunakan strategi pencarian citra tanpa menggunakan nilai *threshold*. Indeks citra yang diamati adalah indeks-1. Skenario-5 menggunakan kelas kuantisasi untuk skala kontras sepanjang 4 kelas, sedangkan skenario-6 menggunakan kelas kuantisasi untuk skala kontras sepanjang 8 kelas. Keduanya menggunakan kelas kuantisasi 64 kelas untuk *range block mean*.

Dari hasil uji coba tersebut, terlihat bahwa perubahan kuantisasi parameter skala kontras sama sekali tidak berpengaruh pada hasil pengembalian citra jika indeks yang digunakan adalah indeks-1. Hal ini dikarenakan indeks-1 sama sekali tidak melibatkan parameter skala kontras, berbeda dengan indeks-2, indeks-3, dan indeks-4 yang merupakan gabungan antara *range block mean* dan skala kontras.

**Tabel 4. Ringkasan Parameter dan Hasil Uji Coba untuk Masing-Masing Skenario**

	Skenario-1	Skenario-2	Skenario-3	Skenario-4	Skenario-5	Skenario-6
Minpart	1	1	1	1	1	1
Maxpart	6	6	6	6	6	6
Skala kontras maksimum	1.0	1.0	1.0	1.0	1.0	1.0
Toleransi RMS terbesar	1.0	1.0	1.0	1.0	1.0	1.0
Threshold	0.0087	0.0087	0.0087	0.0087	-	-
Kuantisasi	2-6	2-6	3-6	3-6	2-6	3-6
Indeks threshold	Indeks-1	Indeks-2	Indeks-1	Indeks-2	-	-
Indeks pengurut	Indeks-3	Indeks-4	Indeks-3	Indeks-4	Indeks-1	Indeks-1
Hasil-Hasil						
PSNR	25.8 dB	25.8 dB	25.8 dB	25.8 dB	25.8 dB	25.8 dB
Citra Hasil yang benar	889	851	880	821	1035	1035
Average retrieval rate	60.39%	57.81%	59.78%	55.77%	70.3%	70.3%



## 8. KESIMPULAN

Dari sistem yang telah dibuat dan uji coba yang telah dilakukan terhadapnya, dapat diambil beberapa kesimpulan sebagai berikut:

- Hasil uji coba visual telah menunjukkan bahwa kode fraktal yang dibuat telah benar, dibuktikan dengan hasil PSNR berkisar antara 25 dB hingga 35 dB.
- Indeks-1 adalah indeks terbaik yang digunakan sebagai ciri citra, yaitu dengan *average retrieval rate* sebesar 70.3%.
- Jika menggunakan strategi pencarian dengan nilai *threshold*, hasil terbaik yang didapatkan adalah sebesar 60.39%, yaitu dengan konfigurasi indeks menggunakan pasangan indeks-1/indeks-3 dan dengan kelas kuantisasi 2-6 bit.

- Jika hanya menggunakan indeks-1, perubahan kuantisasi pada parameter skala kontras sama sekali tidak berpengaruh pada hasil pengambilan citra.
- Kuantisasi terbaik adalah 2-6 bit atau 4 kelas kuantisasi skala kontras dan 64 kelas kuantisasi *range block mean*.

## DAFTAR PUSTAKA

- [1] Citra tekstur brodatz [Online]. URL: <http://www.cipr.rpi.edu/re-source/stills/brodatz.html>
- [2] Y. Fisher, "Fractal image compression," *SIGGRAPH '92 Course Notes*, 1992.
- [3] M. Pi, M. K. Mandal and A. Basu, "Image retrieval based on histogram of fractal parameters," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 597-605, Agustus 2005.