

# Klasifikasi Citra *Content-Based Image Retrieval* Dengan Metode *Shape Base Threshold*

Pandapotan Siagian<sup>1</sup> Erick Fernando<sup>2</sup>

STIKOM DB Jambi

Jl. Sudirman Thehock Jambi, 0741-35095

e-mail: siagian.p@gmail.com<sup>2</sup>, Erick.Fernando\_88@yahoo.com<sup>2</sup>

## Abstrak

*Content Based Image Retrieval (CBIR)* bekerja dengan cara mengukur kemiripan citra dengan semua citra yang ada dalam database sehingga ketepatan pencarian berbanding lurus dengan jumlah citra dalam database. Pencarian citra yang paling mirip mempunyai tingkat lamanya pencarian dengan melakukan klasifikasi citra yang bertujuan untuk mengurangi waktu pencarian lebih singkat dan akurat pada CBIR.

Implementasi *Shape Base Thresholding* untuk klasifikasi citra serta mengukur tingkat akurasi dan waktu klasifikasinya. Penelitian ini, dirancang aplikasi perangkat lunak dengan pemrograman java JDK, aplikasi klasifikasi citra CBIR yang akan mampu mengekstrak fitur warna dan tekstur dari sebuah citra dengan menggunakan *Shape Base Threshold Color Histogram* dan *Entropi Base Histogram*. Hasil dari proses ekstraksi fitur kemudian digunakan oleh perangkat lunak dalam proses learning dan klasifikasi dengan metode *Shape Base Threshold*. Perangkat lunak dibangun dengan metode analisis dan perancangan terstruktur kemudian diimplementasikan dengan Java JDK. Adapun Citra learning yang terdapat pada 8 kelas citra fitur yang di simpan query database yaitu 596 citra bmp dan jpg dengan ukuran 400x400, sebagai sample pengujian dan masing masing citra yang terdapat pada query data base yaitu *Color Histogram* dan *Shape Base Threshold Histogram* yang berbeda.

Aplikasi klasifikasi citra CBIR yang dihasilkan kemudian diuji dengan parameter tingkat akurasi dan waktu klasifikasi. Hasil pengujian menunjukkan bahwa kombinasi fitur warna dan tekstur memberikan tingkat akurasi yang lebih tinggi dibandingkan dengan klasifikasi berdasarkan fitur warna saja atau tekstur saja namun membutuhkan waktu klasifikasi yang lebih lama.

**Kata kunci** : Klasifikasi Citra CBIR, Ekstraksi fitur, *Shape Base Threshold*, *Entropy Base Threshold*

## 1. Pendahuluan

Penyebaran informasi yang dapat di klasifikasi dengan *Content Based Image Retrieval (CBIR)*. Penyebaran database citra yang besar untuk berbagai aplikasi kini telah menjadi realisasi. Database yang dikembangkan, untuk pengelompokan obyek citra satelit dan medis. Banyak pengguna di berbagai bidang profesional - misalnya, geografi, kedokteran, arsitektur, periklanan, desain, fashion, dan penerbitan. Efektif dan efisien mengakses citra yang diinginkan dari database citra yang besar dan beragam sekarang menjadi kebutuhan.

*Content Based Image Retrieval (CBIR)* adalah pengambilan citra berdasarkan fitur visual seperti warna, tekstur dan bentuk [13], [16]. Alasan perkembangannya adalah bahwa dalam banyak database citra besar, metode tradisional pengindeksan citra telah terbukti tidak cukup, melelahkan, dan sangat memakan waktu. Metode-metode lama pengindeksan citra, mulai dari menyimpan citra di database dan menghubungkannya dengan kata kunci atau nomor, untuk menghubungkannya dengan deskripsi dikategorikan, telah menjadi usang. Ini bukan CBIR. Dalam CBIR, setiap citra yang disimpan dalam database memiliki fitur-fiturnya diekstrak dan dibandingkan dengan fitur dari citra query. Hal ini melibatkan dua langkah yaitu Fitur Ekstraksi (*Citra Extraction*), proses ini mengekstraksi fitur citra untuk sebagian dibedakan dan Penyesuaian (*Matching*) proses tahapan ke dua yaitu pencocokan fitur untuk menghasilkan hasil yang secara visual yang sama. Future yang akan di peoses yaitu [1], [3] [16], [18] :

### a. Warna

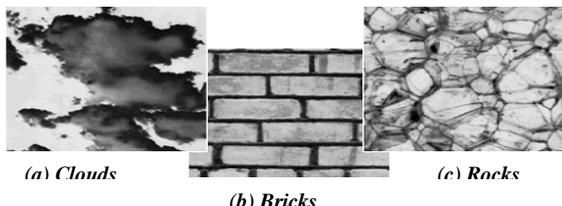
Fitur yang paling penting di ekstrak dari suatu citra yaitu warna. Warna adalah properti citra hasil tergantung pada refleksi cahaya ke mata dan pengolahan informasi di otak. Citra warna sehari-hari dapat dibedakan antara obyek, tempat dan waktu pengambilan citra. Biasanya warna dapat didefinisikan dalam tiga dimensi ruang warna yaitu RGB (*Red, Green, dan Blue*), atau HSV (*Hue, Saturation, dan Value*) atau HSB (*Hue, Saturation, Brightness*). Dua yang terakhir adalah tergantung pada persepsi manusia yaitu HSB (*Hue, Saturation, Brightness*).

Format citra yang banyak di gunakan yaitu JPEG, BMP, JPG, GIF, dengan menggunakan ruang warna RGB untuk menyimpan informasi. Ruang warna RGB didefinisikan dengan bentuk kubus dengan merah, hijau, biru dan *background*. Dengan demikian, vektor dan tiga koordinat ruang. Bila ketiga

koordinat diatur pada sumbu koordinat xyz nilainya 0, warna yang dihasilkan adalah hitam. Bila ketiga sumbu koordinat di set dengan nilai 1, maka warna yang dihasilkan putih.

**b. Tekstur**

Tekstur adalah properti bawaan dari permukaan yang menggambarkan pola visual dan masing-masing properti memiliki homogenitas. Ini berisi informasi penting tentang pengaturan struktural dari permukaan, seperti; awan, daun, batu bata, kain, dll. Fitur yang menggambarkan komposisi fisik yang khas dari permukaan. Sifat tekstur meliputi : Kekasaran (*Coarseness*), Kontras (*Contrast*), Directionality (*Directionality*), Line-rupa (*Line-likeness*), Keteraturan (*Regularity*) [2], [15], [17], Kekasaran (*Roughness*) dan tekstur terdapat pada Gambar 1.



Gambar 1. Tekstur

Tekstur paling penting dari suatu citra dan tekstur ditandai dengan distribusi spasial tingkat keabuan (gray level) di sekitar citra. Untuk membedakan hasil citra tergantung dari nilai spasial gray-level, yang berkontribusi terhadap persepsi tekstur, tekstur dua dimensi tergantung dengan analisis tekstur matriks.

**2. Co-occurrence Matrix**

*Co-occurrence* matriks merupakan representasi fitur tekstur yang dieksplorasi dari tingkat keabuan (*gray level*) spasial. Matriks *Co-occurrence* dapat dirumuskan dalam matematika yaitu [7] , [16] , [12] :

- a. Operator posisi P (i, j), Misalkan citra A adalah sebuah matriks n x n, maka elemen citra A [i] [j]. Dimana A [i] [j] adalah Titik piksel dengan gray level (intensitas) g [i] terjadi, dalam posisi yang ditentukan oleh P, relatif terhadap poin dengan tingkat keabuan (gray level) g [j].
- b. Misalkan C adalah matriks nxn yang dihasilkan dengan membagi A dengan jumlah pasangan titik yang memenuhi P. C [i] [j] adalah ukuran probabilitas pasangan titik piksel P akan memiliki nilai g [i], g [j].
- c. C disebut matriks *Co-occurrence* didefinisikan oleh titik piksel P. Contoh untuk titik piksel P yaitu : "i di atas j", atau " posisi ke kanan dan dua titik piksel ke bawah j".

*Co-occurrence matrix*  $C_t$  daerah region didefinisikan untuk setiap tingkat keabuan (gray level). Tingkat keabuan (a, b) dapat ditulis dalam persamaan :

$$C_t(a,b) = \text{card}\{(s, s+t) \in R^2 | A[s] = a, A[s+t] = b\} \dots\dots\dots(1)$$

Dimana,  $C_t(a, b)$  adalah jumlah pasangan titik piksel, dan disimbolkan dengan (s, s + t) dan dipisahkan oleh vektor t, maka gray-level dari s, dan b menjadi keabuan tingkat s + t.

**3. Metode Shape-Based Thresholding**

**3.1 Klasifikasi**

Klasifikasi adalah proses mencari sekumpulan model atau fungsi yang mendeskripsikan dan membedakan kelas-kelas data untuk tujuan agar fungsi tersebut dapat digunakan untuk memprediksi label objek yang label kelasnya tidak diketahui. Model yang dihasilkan adalah berdasarkan hasil analisis terhadap sekumpulan data *learning* [4] , [12], [16].

**3.1.1 Ekstraksi Fitur**

Fitur dari sebuah gambar adalah karakteristik atau atribut dari sebuah gambar yang dapat membedakannya dari gambar yang lain [4], [13].

Pada referensi [5] , [13] disebutkan bahwa jika diinginkan sebuah sistem yang dapat membedakan objek-objek dengan tipe yang berbeda, maka pertama-tama harus ditetapkan karakteristik objek yang dapat diukur sebagai parameter yang mendeskripsikan objek. Karakteristik inilah yang disebut dengan fitur. Dari beberapa definisi diatas maka ekstraksi fitur dapat didefinisikan sebagai proses mengekstrak karakteristik deskriptif dari sebuah gambar dengan menggunakan metode tertentu. Fitur yang dipilih harus mampu mendeskripsikan objek yang diwakilinya dengan baik. Adapun kriteria fitur yang baik menurut [5] , [15] adalah :

**a. Discrimination**

Fitur diharapkan mampu memberikan perbedaan nilai yang signifikan untuk objek-objek yang tidak berada dalam satu kelas.

**b. Reliability**

Fitur diharapkan mampu memberikan perbedaan nilai yang minimal untuk objek-objek yang berada dalam satu kelas yang sama.

**c. Independence**

Jika digunakan beberapa fitur, sebaiknya fitur-fitur tersebut tidak saling berkorelasi satu sama lain, artinya nilai fitur yang satu tidak mempengaruhi nilai fitur yang lain.

**d. Small Number**

Fitur diharapkan memiliki dimensionalitas yang kecil sehingga jika direpresentasikan sebagai vektor, maka vektor yang dihasilkan akan memiliki jumlah elemen yang sesedikit mungkin.

**3.1.2 Color Histogram**

Warna merupakan fitur yang paling ekspresif dibandingkan dengan fitur visual yang lain [4] [15], [19]. Warna juga merupakan fitur yang paling banyak digunakan dalam *image retrieval*[18]. Salah satu metode yang dapat digunakan untuk mengekstrak fitur warna dari sebuah citra adalah dengan menggunakan metode *Color Histogram*.

Kategori metode *Shape-Based Thresholding* dapat digunakan untuk menghasilkan histogram *Color Histogram*. Ada dua proses yang mendasar dari *Shape-Based Thresholding histogram* yaitu peak dan intervening valley[16]. Metode *convex hull, Hull(g)*, dengan pmf dan analisis concavities dari  $h(g)$  atau sama dengan  $|Hull(g) - p(g)|$ . Convex hull dengan pmf akan menjumlahkan nilai pixel yang sama. Nilai Threshold dapat dihitung dengan persamaan (2) yaitu [15], [19]:

$$T = \arg \max \{ \max [ p(g) - Hull(g) ] \text{ and object smoothness} \} \dots \dots \dots (2)$$

**3.1.3 Spectral Analisis**

Total penjumlahan spektrum daya sebuah citra akan diperoleh dari sinyal eksponensial multi-complex Fungsi eksponensial tersebut akan memperoleh histogram citra yang lebih smooth [16], [19]. Citra smooth suatu hasil iterpretasi dari pmf  $p(g)$  dan refleksi lingkungan sekitar  $g = 0, p(-g)$ , hal ini sering disebut sebagai densitas noise spektral daya, (noisy power spectral density). Untuk memperkecil koefisien  $k = 0 \dots G$ , dengan IDFT (*Inverse Discrete Fourier Transform*), histogram citra asli dapat diinterpretasi  $r(k) = IDFT[\tilde{p}(g)]$  dimana  $\tilde{p}(g) = p(g)$  for  $g \geq 0$  dan  $p(-g)$  for  $g \leq 0$ . Histogram warna akan mendapatkan variansi matrik R yang nilainya sama dengan citra asli.dengan mengubah nilai  $\{r(k)\}$ . Kestabilan nilai smooth threshold diperoleh dengan melewati nilai minimum, sesuai dengan persamaan (3).

$$T_{opt} = \min_g \frac{1}{\left| 1 - \sum_{i=1}^{N_p} a_i e^{-j2\pi g / 256} \right|^2} \text{ dimana } a = R^{-1}r \quad r = \begin{bmatrix} r(1) \\ \dots \\ r(N_p) \end{bmatrix} \text{ dan } R = \begin{bmatrix} r(0) & \dots & r(N_p - 1) \\ r(1) & & \dots \\ \dots & & \dots \\ r(N_p - 1) & \dots & r(0) \end{bmatrix} \dots \dots (3)$$

**3.1.4 Entropy-Based Thresholding**

*Entropy-Based Thresholding (EBT)* menggambarkan distribusi spasial entropy. Distribusi *entropy* merupakan karakteristik tekstur yang berguna dalam proses *image matching* bahkan dengan kondisi dimana citra tidak memiliki tekstur yang homogen [15], [17]. Distribusi gray level maksimum entropy di ubah nilainya keluaran menjadi citra binary.Gray level histogram diperoleh rasio kurva hasil equalisasi entropy  $H'(T) = -P(T)\log(P(T)) - (1 - P(T))\log(1 - P(T))$  terhadap fungsi T, Threshold T dari sumber citra

entropy  $H(T) = -\sum_{g=0}^T p(g)\log(p(g)) - \sum_{g=T+1}^G p(g)\log(p(g))$  dan menghasilkan bulatan-bulatan titik pixel

citra yang lebih halus atau smooth, jika

$$\frac{H'(T)}{H} \geq \left[ \frac{\alpha \log P(T)}{\log(\max(p(1), \dots, p(T)))} + (1 - \alpha) \frac{\log(1 - P(T))}{\log(\max(p(T+1), \dots, p(G)))} \right]$$

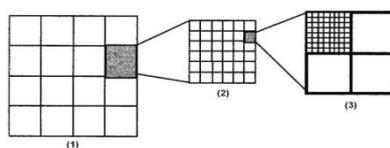
Untuk mendapatkan threshold yang optimal, maka dapat digunakan persamaan (4).

$$T_{opt} = \arg \max_T \{ equal(H_f(T) = \alpha H) \} \dots \dots \dots (4)$$

Dimana parameter  $\alpha$  adalah satu bulatan titik pixel dari spektrum rendah dan  $H_f(T)$  adalah nilai pixel entropy dari citra[18]. Histogram yang optimal dapat juga diperoleh dengan persamaan (5).

$$T_{opt} = \arg \max_T \left\{ \sum_0^T p(g) = (0.5 + |0.5 - \alpha|) \right\} \dots \dots \dots (5)$$

Tahapan ekstrak fitur tekstur dengan menggunakan metode EBT sesuai dengan Gambar 2 adalah sebagai berikut [2], [13] :



Gambar 2. Ilustrasi ekstraksi fitur tekstur dengan menggunakan EBT

- Bagi gambar menjadi 4x4 region yang sama besar
- Bagi tiap region menjadi *sub-block* dengan ukuran yang sama
- Tiap *sub-block* dibagi menjadi 2x2 partisi
- Untuk setiap *sub-block*, nilai dalam tiap partisi dirata-ratakan sehingga tiap *sub-block* dapat diperlakukan sebagai gambar 2x2 *pixel*
- Terapkan *entropy optimum* pada tiap *sub-block*. Sebuah *sub-block* dinyatakan sebagai *entropy block* jika hasil operasi *sub-block* dengan *entropy smooth detector* melebihi nilai *entropy threshold* yang telah ditetapkan sebelumnya.

### 3.2 Prinsip Kerja Shape Base Thresholding

*Shape Base Thresholding* adalah *supervised learning algorithm* dimana sebuah objek diklasifikasikan berdasarkan kelas mayoritas dari  $k$  buah tetangga terdekatnya. Klasifikasi memanfaatkan mekanisme *voting* dari  $k$  buah objek terdekat dan bila hasil *voting* seri, maka label untuk objek akan dipilih secara acak [9] [14], [19].

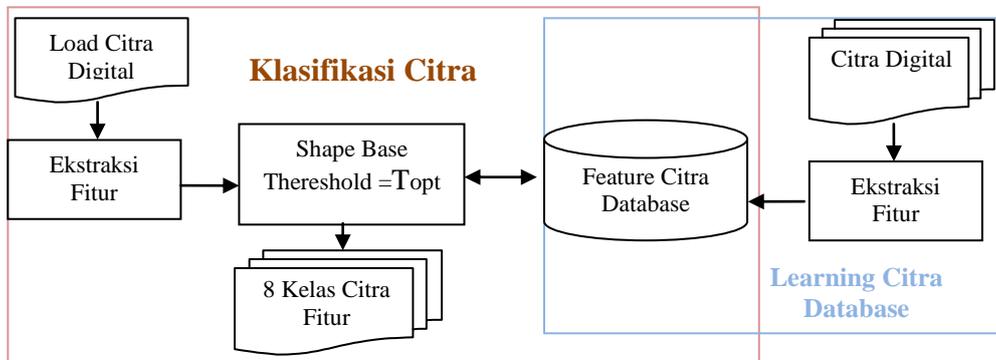
*Shape Base Thresholding* berdasarkan konsep '*learning by analogy*'. Data *learning* dideskripsikan dengan atribut numerik  $n$ -dimensi. Tiap data *learning* merepresentasikan sebuah titik dalam ruang  $n$ -dimensi [4]. Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *Shape Base Thresholding* akan mencari  $k$  buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang  $n$ -dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Quadratic Distance Metric*. Persamaan kita digunakan untuk menurunkan jarak antara dua histogram warna adalah jarak kuadrat metric :

$$d^2(Q, I) = (H_Q - H_I)' A (H_Q - H_I) \dots\dots\dots(6)$$

Persamaan ini terdiri bagian, penurunan masing-masing bagian yang akan dijelaskan dalam bagian berikut [15], [16] :

Bagian pertama terdiri dari perbedaan antara dua histogram warna, atau lebih tepatnya perbedaan jumlah pixel dalam setiap sisi. Bagian ini jelas vektor karena terdiri dari satu baris. Jumlah kolom dalam vektor ini adalah jumlah sisi di histogram. Bagian ketiga adalah transpos vektor itu. Bagian tengah adalah matriks kesamaan. Hasil akhir merupakan jarak warna antara dua gambar. Semakin dekat jaraknya ke nol semakin dekat gambar dalam kesamaan warna. Jarak yang lebih jauh dari nol kurang mirip gambar dalam kesamaan warna.

### 4. Framework Sistem Klasifikasi Citra CBIR



Gambar 3. Framework Sistem Klasifikasi Citra CBIR

Perangkat lunak akan terdiri dari 2 buah proses utama yang saling berkaitan, yaitu proses *learning* dan proses klasifikasi. Input untuk proses *learning* adalah kumpulan *learning image* yang sudah diketahui label kelasnya. Adapun output yang dihasilkan adalah fitur citra yang disimpan dalam sebuah *citra database*. Input untuk proses klasifikasi adalah citra yang akan diklasifikasikan dan fitur *learning image* yang tersimpan dalam *citra database*. Adapun outputnya adalah label kelas citra yang diinputkan. Dengan demikian jelaslah bahwa proses *learning* harus dilakukan sebelum proses klasifikasi sebab output dari proses *learning* menjadi salah satu input yang dibutuhkan dalam proses klasifikasi.

### 3.3 Tahap Learning

Pada proses *learning*, input perangkat lunak adalah kumpulan *learning image* yang telah diketahui label kelasnya. Semua *learning image* akan diekstrak fiturnya dengan *Shape Base Threshold Color Histogram* dan *Entropy Base Histogram*. Hasil dari proses ekstraksi fitur akan disimpan dalam sebuah *citra database*. Berikut ini adalah tahapan yang dilakukan dalam proses *learning* :

- Baca data bitmap *learning image*
- Proses ekstraksi fitur warna dengan *Shape Base Threshold Color Histogram*
- Proses ekstraksi fitur tekstur dengan *Entropy Base Histogram*.

### 3.4 Tahap Classification

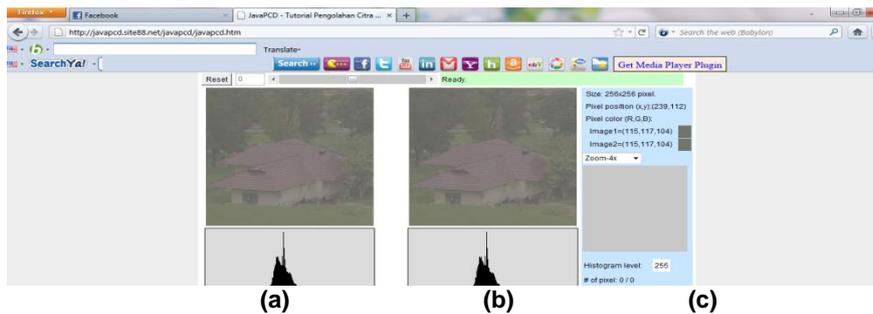
Proses klasifikasi, input perangkat lunak adalah *testing image* yang belum diketahui label kelasnya. *Testing image* akan mengalami proses ekstraksi fitur. Hasil dari proses ekstraksi fitur akan dibandingkan dengan fitur yang tersimpan dalam *citra database* untuk menentukan label kelas *testing image*. Berikut ini adalah tahapan yang dilakukan dalam proses klasifikasi yaitu :



- Baca data bitmap *testing image*
- Proses ekstraksi fitur warna dengan *Shape Base Thershold Color Histogram*
- Proses ekstraksi fitur tekstur dengan *Entropy Base Histogram*.
- Memnyamakan antara hasil ekstraksi fitur *testing image* dengan fitur learning image yang tersimpan dalam *citra database*
- Melakukan voting untuk menentukan label kelas image

### 3.5 Implementasi Sistem

Implementasi merupakan hasil dari tampilan rancangan program yang telah didesain sebaik mungkin, berdasarkan desain yang dirancang oleh penulis dengan menggunakan Web java JDK. Untuk dapat mengakses aplikasi ini anda harus menggunakan browser yang mendukung Java applet. Aplikasi ini telah diujicoba dan dapat dibuka di Internet Explorer, Firefox dan Opera. Tampilan Menu Utama, merupakan tampilan dari menu utama atau halaman depan pada saat program dijalankan oleh pengunjung website. Halaman menu utama pada Gambar 3 adalah implementasi dari rancangan system.



Gambar 3 Halaman Utama dan Proses Klasifikasi CBIR  
 (a) Citra Asli (b) Citra Hasil Praproses (c) Tampilan Informasi Content

## 4 Pengujian

### 4.1 Pra Proses

Data yang digunakan untuk pengujian terdiri dari 8 kelas citra warna bmp, jpg colour 256 bit dengan ukuran 400x400, yang terbagi dalam 8 kelas citra yaitu River.bmp, jpg, Lansat.bmp, jpg, Water.bmp, jpg, Rock.bmp, jpg, Bus.bmp, jpg, Flower.bmp, jpg, Horse.bmp, jpg dan Home.bmp, jpg.

Tabel 5.1. Data untuk pelatihan dan pengujian

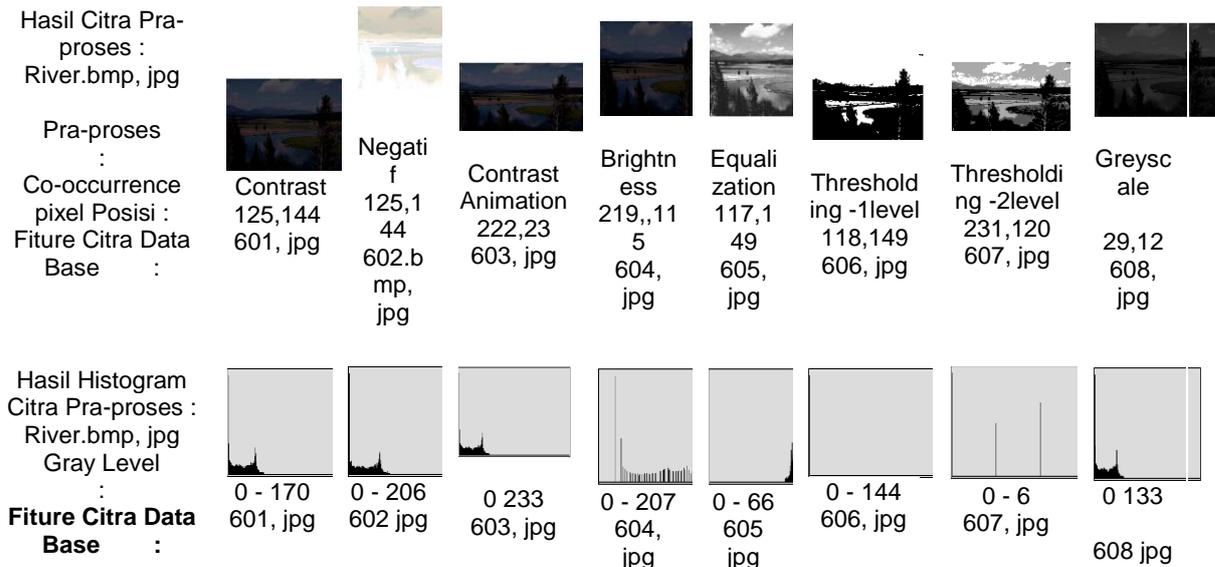
Kelas	Jumlah data learning	Jumlah data testing
River.bmp, jpg	8 Citra	8 Citra
Lansat.bmp, jpg	8 Citra	8 Citra
Water.bmp, jpg	8 Citra	8 Citra
Rock.bmp, jpg	8 Citra	8 Citra
Bus.bmp, jpg	8 Citra	8 Citra
Flower.bmp, jpg	8 Citra	8 Citra
Horse.bmp, jpg	8 Citra	8 Citra
Home.bmp, jpg	8 Citra	8 Citra
<b>Total</b>	64 Citra	64 Citra



Gambar 4. Pra-proses 8 kelas citra digital

Data yang akan di praproses terdiri dari 8 citra warna bmp, jpg colour 256 bit dengan ukuran 400x400, yang terbagi dalam 8 kelas citra yaitu River, Lansat, Water, Rock, Bus, Flower, Horse dan Home. Untuk memperoleh 8 citra untuk masing masing kelas feature citra database, Pra-proses Citra River.bmp, jpg terdapat pada Tabel 5.2.

Tabel 5.2 Pra-proses Citra Digital River.bmp, jpg Untuk Citra Database Kelas A



Tabel 5.3 Histogram Pra-proses Citra Digital River.bmp, jpg Untuk Citra Database Kelas A

Untuk data kelas B, C, D, E, F, G,H citra warna bmp, jpg colour 256 bit dengan ukuran 400x400, Lansat, Water, Rock, Bus, Flower, Horse dan Home, dapat dilakukan Pra-proses sesuai dengan tahapan sesuai dengan tahapan Pra-proses Citra Digital River.bmp, jpg Untuk Citra Database Kelas A yang terdapat pada Gambar 4. Setelah melakukan proses penyimpanan data citra untuk masing masing kelas, maka dapat dilakukan implementasi dan pengujian, adapun analisis hasil yang dicapai oleh system.

#### 4.2 Implementasi CBIR

Pengujian tingkat akurasi dan waktu klasifikasi yang akan dilakukan yaitu memproses dan menampilkan citra hasil yang paling sama ada tiga citra untuk masing-masing pengujian. Maka tingkat akurasi dihitung dengan menghitung persentase citra yang dapat diklasifikasikan dengan benar dalam satuan persen sedangkan waktu klasifikasi dihitung mulai proses pembacaan data bitmap sampai menghasilkan label kelas dan diukur dalam satuan detik per citra, terdapat pada Tabel 5.4.

Tabel 5.4 Implementasi CBIR Citra Digital

Klasifikasi Citra			Jumlah data testing
CBIR	Metode Ekstraksi Fitur	Nilai Topt	
Warna	Shape Base Color Hist 256		10
	Entropy Base Color Hist 256		10
Tekstur	Entropy Base Color Hist 256	Contrast	10
Warna & Tekstur	Shape Base Color Hist & Entropy Base Color Hist	Negatif	10
		Contrast Animation	10
		Brightness	10
		Equalization	10
		Thresholding -1-level	10

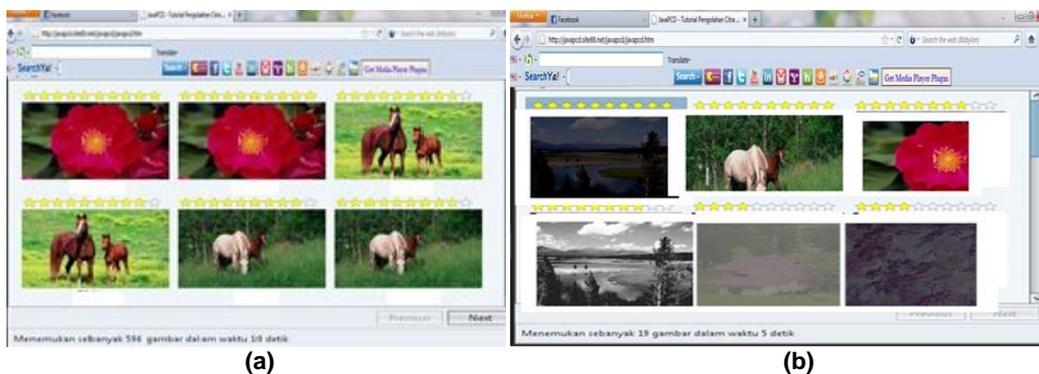
	Thresholding -2-level	10
	Thereshold	10

Total jumlah percobaan yang dilakukan adalah 100 pengujian data testing yang terdiri dari :

- 10 pengujian klasifikasi berdasarkan fitur warna yang diekstrak dengan metode Shape Base *Color Histogram* (SBCH) 256 untuk nilai *Topt*.
- 10 pengujian klasifikasi berdasarkan fitur tekstur yang diekstrak dengan metode *Entropy Base Color Histogram* (EBCH) untuk nilai *Topt*.
- 80 pengujian klasifikasi berdasarkan fitur warna dan tekstur yang diekstrak dengan 8 kombinasi metode ekstraksi fitur warna dan tekstur untuk nilai *Topt*.

### 5.1. Pencarian Berdasarkan Warna untuk SBCH dan EBCH

Pengujian untuk pencarian 596 citra yang terdapat pada fitur citra data base dengan melakukan 10 pengujian klasifikasi berdasarkan fitur warna yang diekstrak dengan metode Shape Base *Color Histogram* (SBCH) untuk nilai Tereshold optimal (*Topt*) membutuhkan waktu ekstrak 10 detik untuk 596 citra data base dan menampilkan 6 citra yang nilai tereshold yang paling optimal, Adapun hasilnya terdapat pada Gambar 5 (a).

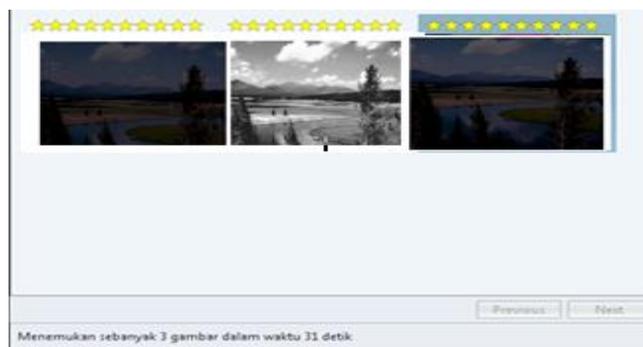


Gambar 5. Pencarian Berdasarkan Warna dengan SBCH dan EBCH (a) Shape Base *Color Histogram* (SBCH) 256 untuk nilai *Topt*. (b) *Entropy Base Color Histogram* (EBCH) untuk nilai *Topt*.

Pengujian untuk pencarian citra yang terdapat pada fitur citra data base dengan melakukan 10 pengujian klasifikasi berdasarkan fitur warna yang diekstrak dengan metode *Entropy Base Color Histogram* (EBCH) untuk nilai Tereshold optimal (*Topt*) membutuhkan waktu ekstrak 5 detik untuk pencarian 596 citra berdasarkan etropy thereshold hanya dapat mencari 19 citra dari 596 citra data base dan menampilkan 6 citra yang nilai tereshold yang paling optimal, Adapun hasilnya terdapat pada Gambar 5 (b).

### 5.2. Pencarian Berdasarkan Fitur dengan EBCH

Pengujian untuk pencarian citra yang terdapat pada fitur citra data base dengan melakukan 10 pengujian klasifikasi berdasarkan fitur fitur yang diekstrak dengan metode *Entropy Base Color Histogram* (EBCH) untuk nilai Tereshold optimal (*Topt*) membutuhkan waktu ekstrak 31 detik untuk pencarian 596 citra berdasarkan etropy thereshold hanya dapat mencari 3 citra dari 596 citra data base dan menampilkan 3 citra yang nilai tereshold yang paling optimal, Adapun hasilnya terdapat pada Gambar 6.



Gambar 6. Pencarian Berdasarkan Fitur dengan EBCH

### 5.3. Pencarian Berdasarkan Warna dan Fitur

Pengujian untuk pencarian citra yang terdapat 80 pengujian klasifikasi berdasarkan fitur warna dan tekstur yang diekstrak dengan 8 kombinasi metode ekstraksi fitur warna dan tekstur untuk mencari nilai *Topt*. Pencarian nilai *Topt* untuk masing-masing Contrast, Negatif, Contrast Animation, Brightness, Equalisation, Threshold 1 level, Threshold 1 level, Gray scale membutuhkan waktu ekstrak 31 detik untuk pencarian 596 citra dan hanya dapat mencari 3 citra dari 596 citra data base dan menampilkan 3 citra yang nilai tereshold yang paling optimal, Adapun hasilnya terdapat pada Gambar 6.

## 6. Kesimpulan

Implementasi sistem akan melakukan pencarian threshold yang paling optimal (*Topt*) dengan metoda EBCH dan SBCH dan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa fitur warna melakukan *voting* paling tepat untuk klasifikasi citra yaitu dengan metode *Shape Base Citra Thresholding*.

Fitur Citra yang memberikan *voting* pencarian paling cepat adalah EBCH yaitu sebesar 5 detik dan Pencarian akurat adalah fitur warna melakukan *voting* dengan waktu 10 detik

Klasifikasi berdasarkan fitur warna dan tekstur dengan *pencarian nilai Topt untuk masing-masing* Contrast, Negatif, Contrast Animation, Brightness, Equalisation, Threshold 1 level, Threshold 1 level, Gray scale membutuhkan waktu yang paling lama mengekstrak yaitu 31 detik untuk pencarian 596 citra dan hanya dapat mencari 3 citra dari 596 citra data base dan menampilkan 3 citra yang nilai tereshold yang paling optimal,

### Daftar Pustaka :

- [1] B. S. Manjunath et al, "Color and Texture Descriptors", IEEE Transcations on Circuits and Sistem for Video Technology, 2011.
- [2] CBIR:Citras, <http://www.ee.columbia.edu/~xlx/courses/vis-hw3/page2.html>, didownload pada tanggal 11 April 2010
- [3] CBIR: Texture Citras, [www.cs.auckland.ac.nz/compsci708s1c/](http://www.cs.auckland.ac.nz/compsci708s1c/), didownload pada tanggal 11 April 2010
- [4] C.K. Leung, F.K. Lam, Performance analysis of a class of iterative image thresholding algorithms, Pattern Recognition, 29(9) (1996) 1523-1530.
- [5] J. Cai, Z.Q. Liu, A New Thresholding Algorithm Based on All-Pole Model, ICPR'98, Int. Conf. on Pattern Recognition, Australia, 1998, pp:34-36.
- [6] Jiawei Han, Micheline Kamber, "Data Mining Concept and Techniques", Academic Press, 2002
- [7] Kenneth R. Castleman, "Digital Image Processing", Prentice Hall, 1996.
- [8] Maher A. Sid Ahmed, "Image Processing: Theory, Algorithm and Architecture", McGrawHill, 1995
- [9] N. Ramesh, J.H. Yoo, I.K. Sethi, Thresholding Based on Histogram Approximation, IEE Proc. Vis. Image, Signal Proc., 142(5) (1995) 271-279.
- [10] Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Pentice Hall, 2002.
- [11] Shalahuddin, M dan Rosa, Belajar Pemrograman dengan Bahasa C++ dan Java. Bandung: Informatika Bandung, 2009.
- [12] Suharto, Herry, dkk, "Pemrograman GUI Swing Java dengan Netbeans 5", Penerbit Andi : Yogyakarta, 2006.
- [13] Sundaram RMD, "Image Mining, Intricacies and Innovations", <http://www.amrita.edu/cde/>, didownload pada tanggal 11 April 2010
- [14] T.W. Ridler, S. Calvard, Picture thresholding using an iterative selection method, IEEE Trans. System, Man and Cybernetics, SMC-8 (1978) 630-632.
- [15] Usman Ahmad, *Pengolaha Citra Digita & Teknik Pemrogramannya*. Yogyakarta : Graha Ilmu, 2005.
- [16] Uniform Quantization, <http://www.cs.wpi.edu/~matt/courses/cs563/>, 2007, didownload pada tanggal 11 April 2010
- [17] William K. Pratt, 1991, "Digital Image Processing", Wiley-Interscience Publication
- [18] Wicaksono, Ady, Dasar-Dasar Pemrograman Java 2, Jakarta: PT.Elex Media Komputindo, Gramedia, 2002.
- [19] Zijun Yang, Jay Kuo, "Survey on Image Content Analysis, Indexing, and Retrieval Techniques and Status Report of MPEG-7", Tamkang Journal of Science and Engineering, 1999.