

# PENGELOMPOKAN GAMBAR BERDASARKAN FITUR WARNA DAN TEKSTUR DENGAN FGKA CLUSTERING (FAST GENETICS K-MEANS ALGORITHM) UNTUK PENCOCOKAN GAMBAR

Dewi Wulansari, S.ST<sup>1</sup>, Entin Martiana K, M.Kom<sup>2</sup>, Nana Ramadijanti, M.Kom<sup>2</sup>  
Mahasiswa Jurusan Teknik Informatika<sup>1</sup>, Dosen Jurusan Teknik Informatika<sup>2</sup>  
Politeknik Elektronika Negeri Surabaya (PENS), Institut Teknologi Sepuluh Nopember (ITS)  
{kweeek@yahoo.co.id, entin@eepis-its.edu, nana@eepis-its.edu}

## Abstrak

*Koleksi gambar-gambar digital jumlahnya semakin banyak. Cara yang biasa dipakai untuk mencari koleksi tersebut adalah menggunakan metadata (seperti caption atau keywords). Tentu saja cara ini dirasa kurang efektif dari sisi penggunaan, ukuran basis data yang besar maupun keakuratan hasil karena bersifat subjektif dalam mengartikan gambar.*

*Berangkat dari hal itulah, dewasa ini telah dikembangkan beragam cara untuk melakukan pencarian gambar yang menggunakan image content suatu gambar (berupa warna, bentuk dan tekstur) yang lebih dikenal dengan istilah CBIR (Content Based Image Retrieval).*

*Penggunaan centroid hasil pengelompokan dataset yang didapat dari hasil HSV histogram dan filter Gabor dari beberapa gambar menggunakan FGKA, bisa digunakan sebagai acuan untuk melakukan pencarian. FGKA merupakan gabungan antara Algoritma Genetika dan Algoritma K-Means.*

*FGKA selalu konvergen pada wilayah global. Pengelompokan dan pencarian gambar berdasarkan fitur warna-tekstur didapati tingkat akurasi yang lebih tinggi dibandingkan dengan berdasarkan fitur warna saja, tekstur saja ataupun tanpa klustering.*

**Kata Kunci :** Algoritma Genetika, K-Means Clustering, CBIR, HSV Histogram, Filter Gabor.

## 1. Pendahuluan

Sebelum tahun 1990-an pencocokan gambar dilakukan dengan menggunakan pendekatan indeks dan informasi citra berbasis text. Tanpa adanya kemampuan untuk memeriksa *content* sebuah gambar, yang biasa digunakan dalam proses pencarian adalah *metadata* suatu gambar (misalnya, *captions* atau *keywords*). Kata kunci yang dikodekan adalah terbatas pada beberapa istilah yang dihasilkan untuk masing-masing referensi gambar. Akibatnya, beberapa gambar yang dihasilkan akan tampak sangat berbeda dibandingkan dengan keinginan user.

Untuk menghindari teknik tersebut, maka dikembangkan pendekatan alternatif yaitu teknik mencari gambar hanya berdasarkan informasi yang ada pada gambar. Teknik *image retrieval* ini dapat mencapai rata-rata kemampuan pencocokan yang tinggi.

*Content Based Image Retrieval (CBIR)*, yang juga dikenal dengan istilah *Query By Image Content (QBIC)* dan *Content Based Visual Information Retrieval (CBVIR)* adalah suatu aplikasi *computer vision* yang digunakan untuk melakukan pencarian gambar-gambar digital pada suatu database.

Yang dimaksud dengan "*Content-based*" di sini adalah bahwa yang dianalisa dalam proses pencarian itu adalah *actual contents* (kandungan aktual) sebuah gambar. Istilah *content* pada konteks ini merujuk pada warna, bentuk, tekstur, atau informasi lain yang

diperoleh dari gambar itu sendiri.

Proses secara umum dari CBIR adalah gambar yang menjadi query dilakukan proses ekstraksi feature (*image contents*), begitu halnya dengan gambar yang ada pada sekumpulan gambar juga dilakukan proses seperti pada gambar query.

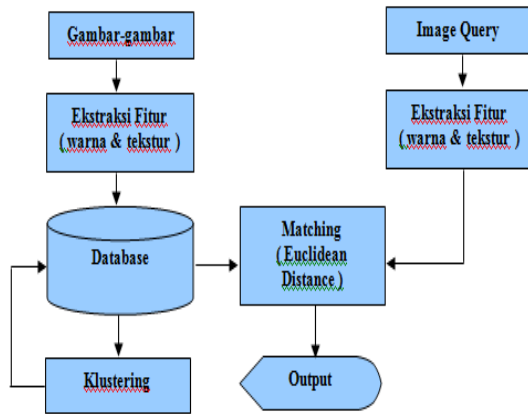
Ada dua teknik dalam sistem CBIR ini, yaitu dengan klustering dan tanpa klustering. Metode klustering akan mengelompokkan gambar-gambar yang ada dalam database ke dalam kluster-kluster terlebih dahulu, kemudian baru dihitung kedekatan centroid klusteranya dengan gambar query. Sedangkan jika tanpa klustering, akan menghitung kedekatan jarak gambar query dengan masing-masing gambar dalam database. Hal ini tentu membutuhkan waktu komputasi yang lebih lama, dan tingkat akurasi yang lebih rendah. Oleh karena itu, metode klustering dipilih dalam sistem matching ini dengan tujuan untuk mempercepat proses komputasi serta meningkatkan keakuratan hasil matching.

Beragam cara telah diajukan pada sistem CBIR ini, seperti pada penelitian sebelumnya yang mengelompokkan gambar berdasarkan fitur warna saja, menggunakan FGKA (*Fast Genetic KMEANS Algorithm*) yang memiliki hasil cukup akurat.

Mengacu pada penelitian tersebut, tentu hal ini bisa dikembangkan melalui proses penambahan metode pencarian gambar yang menggunakan penanda hasil segmentasi sejumlah data yang didalamnya sudah tersimpan fitur warna (*HSV color histogram*) dan tekstur (*Gabor filter*). Sedangkan teknik segmentasi yang

dipakai adalah *Fast Genetic K-Means Algorithm (FGKA)*.

Sehingga, diharapkan dari penelitian ini akan lebih meningkatkan nilai keakuratan apabila menggunakan beberapa ekstraksi fitur gambar lainnya.



Gambar 1: Blok Diagram Sistem

## 2. Metode Pencarian Gambar

Ada tiga tahapan utama dalam pencarian gambar ini, yaitu ekstraksi fitur, klustering dan *matching* (pencocokan).

Ekstraksi fitur adalah proses pengambilan histogram dan hasil gabor, baik dari gambar database maupun gambar *query*. Klustering adalah proses untuk mengelompokkan data-data yang mempunyai kemiripan. Sedangkan *matching* (pencocokan), adalah proses perbandingan antara gambar *query* dengan gambar dalam database.

### 2.1 Ekstraksi Fitur

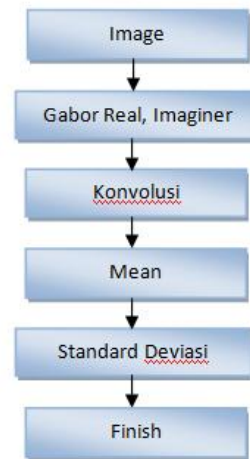
Seperti disebutkan diatas bahwa ekstraksi fitur adalah proses pengambilan histogram (dari warna) serta nilai mean dan standard deviasi (untuk fitur tekstur), baik dari gambar database maupun gambar query.



Gambar 2: Diagram Blok Ekstraksi Warna

Tahap ini terdiri dari beberapa sub tahapan, yaitu:

- Pengambilan nilai RGB tiap pixel yang kemudian langsung dikonversi ke HSV.
- Kuantisasi warna dari yang semula berjumlah (360 x 255 x 255) atau 23409000 kemungkinan warna, diubah menjadi (4 x 4 x 4) atau 64 kemungkinan warna. Dengan cara ini, nilai H berkisar antara 0 sampai dengan 3, S berkisar antara 0 sampai dengan 3, dan V berkisar antara 0 sampai dengan 3.
- Normalisasi.
- Pembuatan HSV Histogram. Pada langkah ini juga dilakukan pembuatan Thumbnails yang berguna untuk menampilkan hasil pencarian dalam bentuk icon.



Gambar 3: Diagram Blok Ekstraksi Tekstur

Tahap ini terdiri dari beberapa sub tahapan, yaitu:

- Inisialisasi variabel sebelum digunakan pada fungsi GaborMask. Langkah berikut ini untuk inisialisasi variabel sebelum digunakan pada fungsi GaborMask:
  - Lakukan inisialisasi nilai rata-rata jarak antar ridge.
  - Lakukan inisialisasi sudut orientasi.
  - Lakukan inisialisasi lebar kernel.
  - Lakukan inisialisasi frekuensi.
  - Lakukan inisialisasi luas spasial dan bandwidth dari filter.
  - Panggil fungsi masking dengan filter gabor.
- Konvolusi. Hasil pembangkitan berbagai fungsi filter Gabor kemudian akan kita lakukan konvolusi terhadap gambar. Dengan langkah berikut ini dapat dilakukan konvolusi pada gambar dengan kernel yang sebelumnya dibangkitkan :
  - Kirimkan kernel yang telah dibangkitkan ke fungsi convolution.

2. Pada fungsi convolution deklarasikan variabel op yang merupakan operator BufferedImageOp.
3. Inisialisasikan op dengan operator ConvolveOp dengan parameter kernel yang diterima.
4. Lakukan konvolusi pada image dengan menggunakan operator op.filter menggunakan parameter gambar sumber image dan parameter lain null.
5. Kembalikan image hasil konvolusi pada fungsi yang memanggilnya.

c. Mean

Langkah – langkah dalam mendapatkan dari gambar query dan gambar training adalah sebagai berikut :

1. Tentukan nilai Mean  $mean[jmlSudut]$ .
2. Tentukan  $x$ =lebar gambar dan  $y$ =tinggi gambar.
3. Untuk  $k=0$  sampai  $k=jmlSudut$ , kerjakan langkah nomor 4.
4. Tentukan nilai mean ke- $k = 0$ .
5. Untuk  $i=0$  sampai  $i=x$ , kerjakan langkah nomor 6.
6. Untuk  $j=0$  sampai  $j=y$ , kerjakan langkah nomor 7.
7. Jumlahkan nilai mean ke- $k$  dengan nilai pixel gambar hasil ekstraksi pada pixel ke- $[i][j]$ .
8. Setelah keluar dari iterasi  $i$  dan  $j$ , tentukan nilai mean ke- $k$  sebagai hasil bagi nilai mean ke- $k$  dengan jumlah pixel observasi.

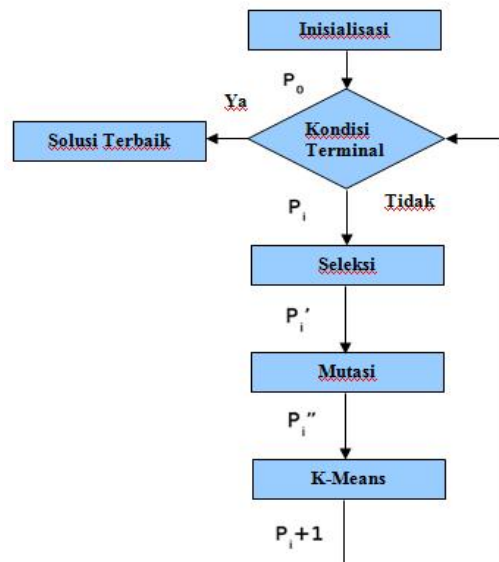
d. Standard Deviasi

Merupakan pengukuran penyebaran dari kumpulan data. Dalam sistem ini, Standard Deviasi dihitung dengan langkah-langkah sebagai berikut:

1. Tentukan nilai Standard Deviasi  $stdDev[jmlSudut]$ .
2. Tentukan  $x$ =lebar gambar dan  $y$ =tinggi gambar.
3. Tentukan  $xy[jmlSudut]$  sebagai variabel penampung nilai pixel sementara.
4. Untuk  $k=0$  sampai  $k=jmlSudut$ , kerjakan langkah nomor 5.
5. tentukan nilai  $xy$  ke- $k = 0$ .
6. Untuk  $i=0$  sampai  $i=x$ , kerjakan langkah nomor 7.
7. Untuk  $j=0$  sampai  $j=y$ , kerjakan langkah nomor 8.
8. Jumlahkan nilai  $xy$  ke- $k$  dengan nilai pixel gambar hasil ekstraksi pada pixel ke- $[i][j]$  dikurangi nilai mean ke- $[k]$  kemudian dikuadratkan.
9. Setelah keluar dari iterasi  $i$  dan  $j$ , tentukan nilai  $stdDev$  ke- $k$  sebagai akar kuadrat  $xy$  ke- $[k]$  kemudian dibagi jumlah pixel observasi.

## 2.2 Klastering

Tahap ini merupakan implementasi dari algoritma FGKA untuk melakukan klasterisasi terhadap sejumlah HSV histogram yang di *combine* dengan nilai hasil filter gabor, sesuai dengan kedekatan jarak (kemiripan) antara gambar-gambar.



Gambar 4: Diagram Blok FGKA

Tahap klastering diawali dengan inisialisasi dataset, dan probabilitas mutasi, besarnya  $K$ , besar populasi, dan jumlah generasi pada tiap populasi. Dataset masukan berasal dari obyek yang menyimpan Array histogram tiap gambar.

Contoh inisialisasi dataset :

DataSet = { 19, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 5, 19, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 9, 17, 3, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 1, 50, 44, 38, 37, 37, 38, 39, 46, 0, 0, 0, 0, 0, 0, 0, 0 }

Panjang array menjadi 80 didapatkan dari 64 dimensi dari histogram warna dan 16 dimensi dari hasil gabor dengan 8 sudut orientasi (mean dan standard deviasi).

### 2.2.1 Operator Seleksi

Operator seleksi yang digunakan dalam algoritma ini adalah seleksi proporsional. Hasil seleksi didapatkan dari populasi saat itu ( $S_1, S_2, \dots, S_Z$ ) yang mempunyai probabilitas ( $p_1, p_2, \dots, p_Z$ ) dengan definisi sebagai berikut:

$$p_z = \frac{F(S_z)}{\sum_{z=1}^Z F(S_z)} \quad (z = 1 \dots Z)$$

Dari probabilitas ini, kemudian dilakukan penyeleksian menggunakan *Roulette Wheel*, yang dengan cara itu, kromosom dengan probabilitas yang tinggi akan bertahan untuk ikut diproses dalam operator selanjutnya

### 2.2.2 Operator Mutasi

Pada operator ini, tiap kromosom dikodekan dengan  $a_1 a_2 \dots a_N$  dan operator mutasi melakukan mutasi pada suatu gen  $a_n$  ( $n = 1 \dots N$ ) dengan nilai baru  $a_n'$  dengan sejumlah  $0 < MP < 1$  sebagai parameter yang dimasukkan oleh pengguna. Nilai tersebut dinamakan probabilitas mutasi. Mutasi dilakukan dengan  $a_n'$  yang dipilih secara random dari  $\{1, 2, \dots, K\}$  dengan distribusi  $(p_1, p_2, \dots, p_z)$  yang didefinisikan dengan rumus:

$$p_k = \frac{1.5 * d_{max}(\vec{X}_n) - d(\vec{X}_n, \vec{c}_k) + 0.5}{\sum_{k=1}^K (1.5 * d_{max}(\vec{X}_n) - d(\vec{X}_n, \vec{c}_k) + 0.5)}$$

dimana  $d(\vec{X}_n, \vec{c}_k)$  adalah jarak Euclidean antara data  $\vec{X}_n$  dan titik pusat  $\vec{c}_k$  dari kluster ke-k. [2,3]

### 2.2.3 Operator K-Means

Operator K-Means ini digunakan untuk mempercepat konvergensi. Solusi yang ada dikodekan dengan  $a_1 a_2 \dots a_N$ . Operator ini akan mengganti isi dari gen  $a_n$  ( $n = 1 \dots N$ ) dengan nilai baru  $a_n'$ , dimana nilai yang baru merupakan kluster dengan jarak terpendek dari data  $a_n$  yang dihitung menggunakan rumus Euclidean.

### 2.3 Matching

Setelah proses klastering selesai dilakukan, maka tiap kluster tersebut dihitung nilai gabungan antara histogram dengan nilai gabor rata-ratanya (untuk dijadikan centroid). Nilai centroid-centroid ini kemudian dibandingkan dengan *HSV histogram* dan nilai gabor gambar query. Centroid yang memiliki jarak paling dekat merupakan solusinya.

Cara yang dipakai untuk mengukur jarak antar dua histogram adalah menggunakan Euclidean distance. Rumusnya:

$$d(A, B) = \sqrt{\sum_{j=1}^n (H_j^A - H_j^B)^2}$$

Setelah centroid yang memiliki jarak paling dekat tadi ditemukan, seluruh *HSV histogram* anggota centroid tersebut kemudian diukur jaraknya dengan *HSV histogram* gambar query menggunakan rumus euclidean distance. Hasilnya kemudian diurutkan. Hanya 10 gambar dengan selisih paling kecil saja yang ditempatkan pada posisi teratas.

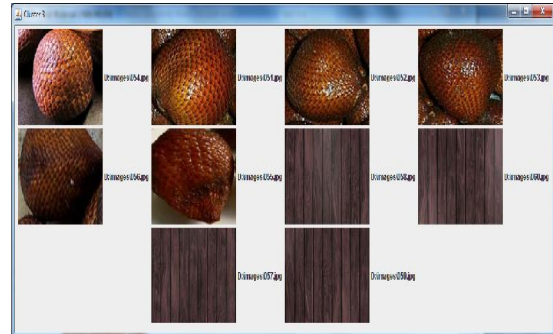
## 3. Hasil Percobaan

Hasil percobaan pada makalah ini terdiri dari dua bagian. Bagian adalah pertama hasil klastering, sedangkan bagian kedua adalah hasil matching.

## 3.1 Hasil Klastering

Berikut ini merupakan contoh hasil klastering dengan probabilitas mutasi (MP) = 0,1.

### A. Untuk JE&JP=3



Gambar 5 : Hasil Klastering JE&JP=3; MP=0,1

Pada percobaan ini, hasil kluster didominasi dengan gambar salak, dengan tingkat kemiripan 0,6. Sedangkan hasil kluster yang lain dapat dilihat pada Tabel 1.

Kluster	Tingkat Kemiripan
0	1
1	0,56
2	0,83
3	0,6
4	0,54
5	0,8
6	0,67
7	1
8	0,75
9	0,53
<b>Rata-rata</b>	0,728
<b>TWCV</b>	605165,2256410257
<b>Waktu Komputasi</b>	1232

### B. Untuk JE&JP=5



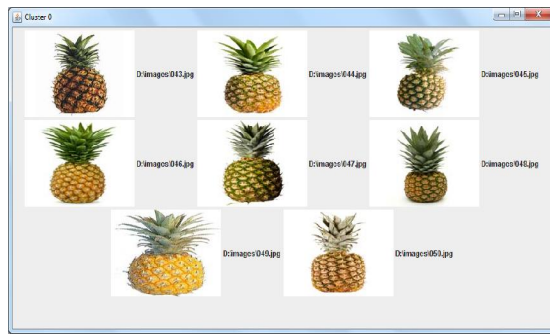
Gambar 6 : Hasil Klastering JE&JP=5; MP=0,1

Pada percobaan ini, hasil kluster didominasi dengan gambar angsa air, dengan tingkat kemiripan 1. Sedangkan hasil kluster yang lain dapat dilihat pada Table 2.

Tabel 2: Hasil Klaster (JE & JP = 5; MP = 0.1)

Klaster	Tingkat Kemiripan
0	1
1	0.54
2	1
3	0.625
4	1
5	1
6	0.54
7	1
8	0.875
9	0.67
<b>Rata-rata</b>	<b>0.825</b>
<b>TWCV</b>	<b>1040010.9376965514</b>
<b>Waktu Komputasi</b>	<b>3949</b>

C. Untuk JE&JP=10



Gambar 7 : Hasil Klastering JE&JP=10; MP=0,1

Pada percobaan ini, hasil klaster didominasi dengan gambar nanas, dengan tingkat kemiripan 1. Sedangkan hasil klaster yang lain dapat dilihat pada Tabel 3.

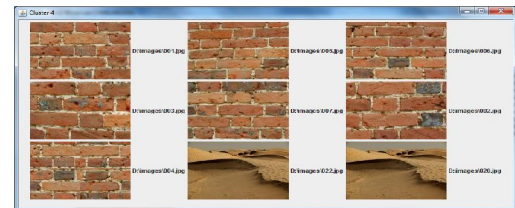
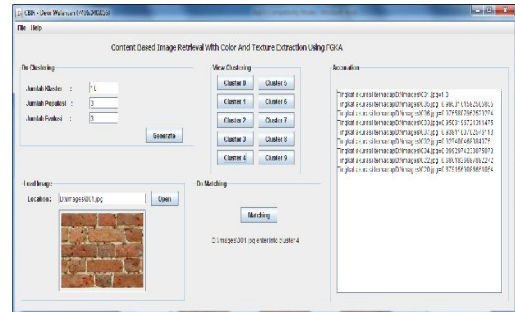
Tabel 3: Hasil Klaster (JE & JP = 10; MP = 0.1)

Klaster	Tingkat Kemiripan
0	1
1	0.83
2	0.64
3	0.67
4	1
5	1
6	1
7	0.5
8	1
9	0.67
<b>Rata-rata</b>	<b>0.831</b>
<b>TWCV</b>	<b>2136136.4494904084</b>
<b>Waktu Komputasi</b>	<b>7441</b>

3.2 Hasil Matching

Berikut ini merupakan contoh hasil matching untuk image query yang sama yaitu batu bata dengan probabilitas mutasi (MP) = 0,1.

A. Untuk JE&JP=3

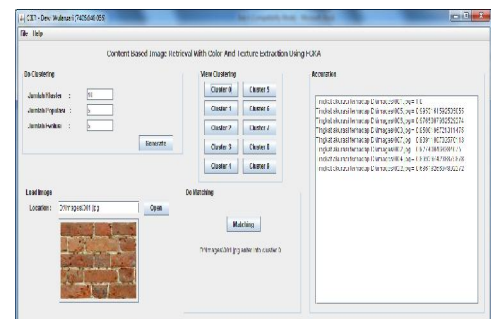


Gambar 8: Hasil Matching JE&JP=3; MP=0,1

Tabel 4: Hasil Matching (JE & JP = 3; MP = 0.1)

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	7.3484692283495345
2	Akurasi Matching	0.78
3	Waktu Komputasi	114

B. Untuk JE&JP=5

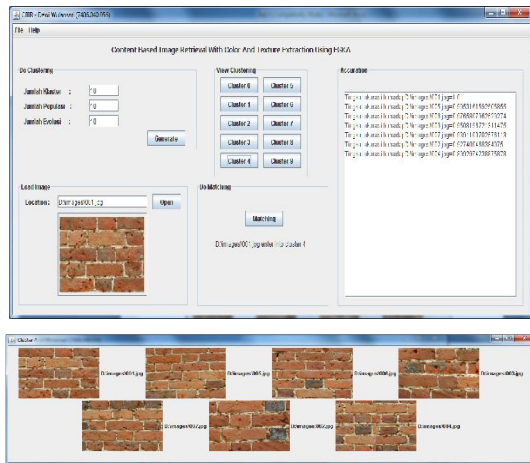


Gambar 9: Hasil Matching JE&JP=5; MP=0,1

Tabel 5: Hasil Matching (JE & JP = 5; MP = 0.1)

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	14.7648230602334
2	Akurasi Matching	0.88
3	Waktu Komputasi	156

C. Untuk JE&JP=10



Gambar 10: Hasil Matching JE&JP=10; MP=0,1

Tabel 6: Hasil Matching (JE & JP = 10; MP = 0.1)

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	15.511933384829668
2	Akurasi Matching	1
3	Waktu Komputasi	254

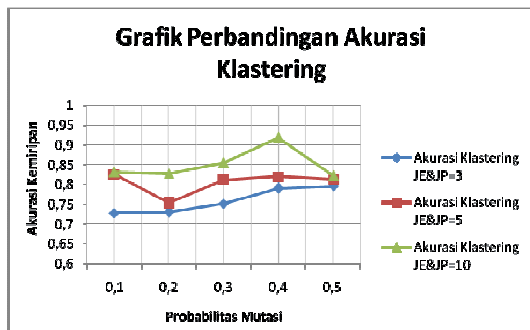
3.3 Perbandingan Hasil Klastering

Tabel 6: Perbandingan Hasil Klastering antara JE&JP=3 ; JE&JP=5 ; JE&JP=10

MP	JE&JP=3		JE&JP=5		JE&JP=10	
	Kemiripan	Waktu	Kemiripan	Waktu	Kemiripan	Waktu
0.1	0,728	1232	0.825	3949	0.831	7441
0.2	0,73	1170	0.7525	3120	0.827	7566
0.3	0,751	1326	0.811	2403	0.855	8018
0.4	0,79	1311	0.8195	2839	0.918	7971
0.5	0,795	1248	0.8125	2465	0.8225	8081
Total	3,794	6287	4.0205	14776	4.2535	39077
Rata	0,76	1257.4	0.8041	2955.2	0.8507	7815.4

Berdasarkan Tabel 6 di atas, dapat disimpulkan bahwa :

- ✓ Klastering berdasarkan fitur Warna-Tekstur dengan JE&JP=10 membutuhkan waktu komputasi yang paling lama dibandingkan dengan menggunakan JE&JP=3 dan JE&JP=5.



Gambar 11: Grafik Perbandingan Hasil Klastering untuk JE&JP=3 ; JE&JP=5 ; JE&JP=10

Berdasarkan grafik 11 di atas, dapat disimpulkan bahwa :

- ✓ Klastering berdasarkan fitur Warna-Tekstur dengan JE&JP=10 mempunyai rata-rata kemiripan tertinggi dibandingkan dengan menggunakan JE&JP=3 dan JE&JP=5.

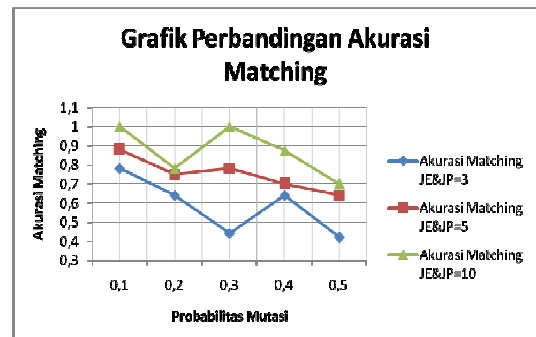
3.4 Perbandingan Hasil Matching

Tabel 7: Perbandingan Hasil Matching antara JE&JP=3 ; JE&JP=5 ; JE&JP=10

MP	JE&JP=3		JE&JP=5		JE&JP=10	
	Akurasi	Waktu	Akurasi	Waktu	Akurasi	Waktu
0.1	0.78	114	0.88	156	1	254
0.2	0.64	124	0.75	172	0.78	273
0.3	0.44	112	0.78	186	1	281
0.4	0.64	109	0.70	159	0.875	269
0.5	0.42	125	0.64	168	0.7	248
Total	2.92	584	3.75	841	4.355	1325
Rata	0.584	116.8	0.75	168.2	0.871	265

Berdasarkan Tabel 7 di atas, dapat disimpulkan bahwa :

- ✓ Matching berdasarkan fitur Warna-Tekstur dengan JE&JP=10 membutuhkan waktu komputasi yang paling lama dibandingkan dengan menggunakan JE&JP=3 dan JE&JP=5.



Gambar 12: Grafik Perbandingan Hasil Matching untuk JE&JP=3 ; JE&JP=5 ; JE&JP=10

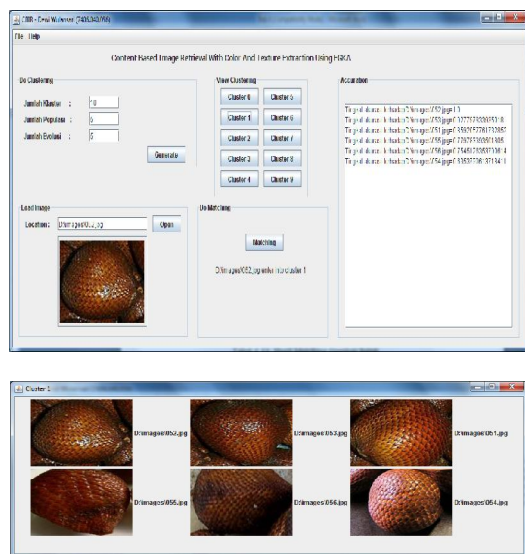
Berdasarkan grafik 12 di atas, dapat disimpulkan bahwa :

- ✓ Matching berdasarkan fitur Warna-Tekstur dengan JE&JP=10 mempunyai rata-rata kemiripan tertinggi dibandingkan dengan menggunakan JE&JP=3 dan JE&JP=5.

3.5 Perbandingan Hasil Matching Fitur Warna-Tekstur, Warna saja, Tekstur saja dan Tanpa Klastering

Pengujian kali ini berguna untuk melihat perbandingan hasil matching terhadap fitur Warna-Tekstur, Warna saja, Tekstur saja dan tanpa klastering dengan menggunakan gambar query yang berbeda-beda. Pengujian dilakukan dengan menggunakan jumlah evolusi dan populasi = 5 serta probabilitas mutasi sama dengan 0.5 .

### A. Matching Warna-Tekstur Dengan Gambar Query Salak

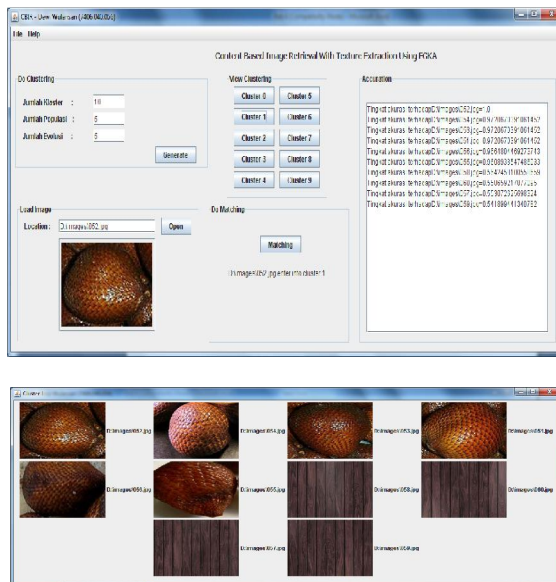


Gambar 13: Matching Warna-Tekstur Salak

Tabel 8: Hasil Matching Warna-Tekstur Salak

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	24.56339438742778
2	Akurasi Matching	1
3	Waktu Komputasi	163

### C. Matching Tekstur Dengan Gambar Query Salak

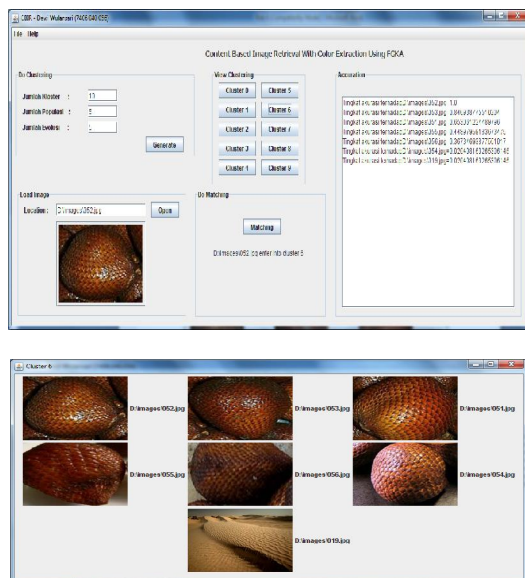


Gambar 15: Matching Tekstur Salak

Tabel 10: Hasil Matching Tekstur Salak

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	4.56339438742778
2	Akurasi Matching	0.6
3	Waktu Komputasi	123

### B. Matching Warna Dengan Gambar Query Salak

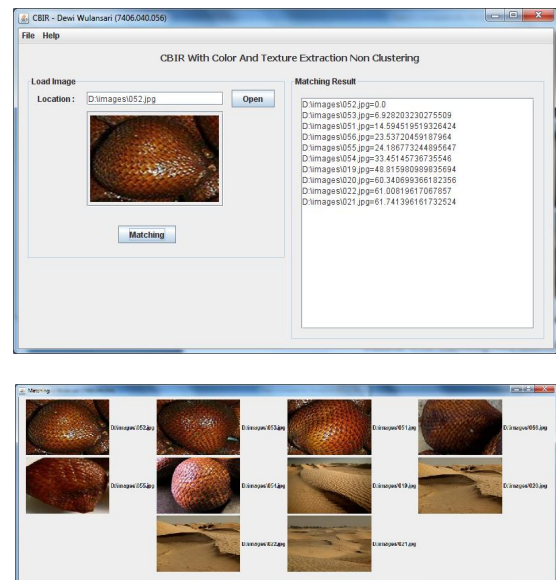


Gambar 14: Matching Warna Salak

Tabel 9: Hasil Matching Warna Salak

No.	Item	Hasil
1	Jarak Gambar Query dengan Centroid Terdekat	14.56339438742778
2	Akurasi Matching	0.86
3	Waktu Komputasi	106

### D. Matching Non-Klastering Dengan Gambar Query Salak



Gambar 16: Matching Non Klastering Salak

Tabel 11: Hasil Matching Non Klastering Salak

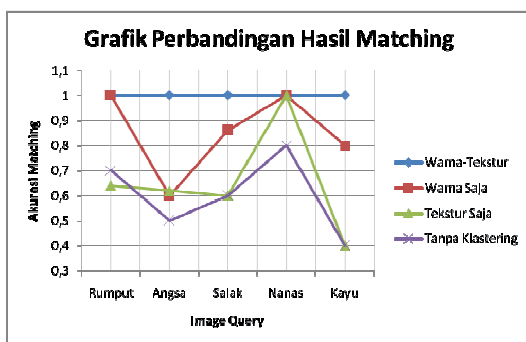
No.	Item	Hasil
1	Akurasi Matching	0.6
2	Waktu Komputasi	341

Tabel 8: Perbandingan Hasil Matching

	Warna-Tekstur		Warna Saja		Tekstur Saja		Tanpa Klastering	
	Akurasi	Waktu	Akurasi	Waktu	Akurasi	Waktu	Akurasi	Waktu
Rumput	1	140	1	102	0.64	125	0.7	327
Angsa	1	147	0.6	114	0.62	120	0.5	340
Salak	1	163	0.86	106	0.6	123	0.6	341
Nanas	1	152	1	118	1	121	0.8	338
Kayu	1	125	0.8	105	0.4	125	0.4	356
Total	5	727	4.26	545	3.26	614	3	1702
Rata	1	145.4	0.852	109	0.652	122.8	0.6	340.4

Berdasarkan Tabel 8 di atas, dapat disimpulkan bahwa :

- ✓ Hasil matching berdasarkan fitur Warna-Tekstur memiliki tingkat akurasi tertinggi dibandingkan hanya berdasarkan fitur warna saja atau tekstur saja.
- ✓ Hasil matching tanpa klastering memiliki tingkat akurasi paling rendah dan membutuhkan waktu komputasi paling lama.



Gambar 13: Perbandingan Hasil Matching

#### 4. Kesimpulan dan Saran

Dari hasil pengujian dan analisa data yang telah dipaparkan tadi, dapat disimpulkan bahwa:

1. Dengan metode ini, gambar dalam klaster yang didapatkan pada beberapa kali running program berbeda.
2. FGKA dengan JE&JP = 10 memiliki tingkat akurasi tertinggi. Hasil klastering dan matching memiliki tingkat akurasi lebih dari 85% dibandingkan dengan menggunakan JE&JP = 3 dan 5.
3. FGKA dengan JE&JP = 10 membutuhkan waktu komputasi hampir 2kali lebih lama dibandingkan dengan menggunakan JE&JP = 3 dan 5.
4. Hasil matching berdasarkan fitur Warna-Tekstur memiliki tingkat akurasi tertinggi dibandingkan dengan berdasarkan fitur warna saja atau tekstur saja.
5. Penggunaan klastering ternyata mampu menghasilkan tingkat akurasi yang lebih tinggi yaitu rata-rata akurasi diatas 80% daripada non-klastering yang hanya mencapai rata-rata akurasi 60%, sedangkan dari sisi waktu komputasi non-klastering membutuhkan waktu 2 kali lebih lama dibandingkan dengan menggunakan klastering.

Untuk lebih meningkatkan akurasi hasil pencarian pada system CBIR, perlu dipertimbangkan penggunaan ekstraksi fiturnya serta data training yang digunakan.

Karena keduanya memiliki pengaruh terhadap hasil pengklasteran dan pencariannya.

Selain itu juga disarankan agar dilakukan ekstraksi fitur tambahan yang merupakan penggabungan dari ketiga ekstraksi fitur yaitu warna, bentuk dan tekstur untuk lebih memaksimalkan akurasi hasil.

#### Daftar Pustaka

- [1] Anonym, "Content-based image retrieval", [http://en.wikipedia.org/wiki/Content-based\\_image\\_retrieval](http://en.wikipedia.org/wiki/Content-based_image_retrieval)
- [2] Widodo, Yanu, "Pencarian Gambar Berdasarkan Fitur Warna Dengan GA-KMeans Clustering", Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2008.
- [3] Zakiyah, Farah, "Pengelompokan Gambar Berdasarkan Warna dan Bentuk Menggunakan FGKA Clustering", Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, Juli 2010.
- [4] Ferdian Alfatah, Edwin, "Klasifikasi Ikan Dengan Menggunakan Hierarkikal Klastering", Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2009.
- [5] Satriya Wardhana, Whisnu, "Ekstraksi Fitur Tekstur Image Diatom Dengan Menggunakan Filter Gabor", Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2008.
- [6] Sukma Wantara, Danang, "Studi Analisa Perbandingan Teknik Ekstraksi Fitur Dalam CBIR Gambar Ikan", Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2008.
- [7] Martiana, Entin, "Perbaikan Kinerja Algoritma Klasterisasi K-Means Genetika", FTIF-ITS.
- [8] Ramadjanti, Nana, "Image Processing", Laboratorium Computer Vision, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2008.
- [9] Anonym, "Color histogram", [http://en.wikipedia.org/wiki/Color\\_histogram](http://en.wikipedia.org/wiki/Color_histogram).
- [10] Firman, "Algoritma Genetika dan Contoh Aplikasinya", Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember.
- [11] Basuki, Achmad, "Suatu Alternatif Penyelesaian Permasalahan Optimasi dan Machine Learning Algoritma Genetika", Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2003.