PENCARIAN GAMBAR BERDASARKAN FITUR WARNA DENGAN GA-KMEANS CLUSTERING

Yanu Widodo[1], Entin Martiana K[2] , Achmad Basuki[2]
Mahasiswa Jurusan Teknologi Informasi[1], Dosen Jurusan Teknologi Informasi[2]
Politeknik Elektronika Negeri Surabaya (PENS), Institut Teknologi Sepuluh Nopember (ITS)
{yanu@student.eepis-its.edu, entin@eepis-its.edu, basuki@eepis-its.edu}

Abstrak

Koleksi-koleksi gambar digital di bidang perdagangan, pemerintahan, akademik, dan rumah sakit jumlahnya semakin banyak. Koleksi tersebut merupakan hasil digitalisasi foto-foto analog, diagram-diagram, lukisan-lukisan, gambar-gambar, dan buku-buku. Cara yang biasa dipakai untuk mencari koleksi tersebut adalah menggunakan metadata (seperti caption atau keywords). Tentu saja cara ini tidak praktis, melelahkan, dan juga mahal (karena masih menggunakan tenaga manusia untuk mendeskripsikan gambar dalam database)

Berangkat dari hal diatas itulah, dewasa ini telah dikembangkan beragam cara untuk melakukan pencarian gambar yang menggunakan image content suatu gambar (yaitu warna, bentuk dan tekstur). Penggunaan centroid hasil pengelompokan HSV histogram dari beberapa gambar menggunakan FGKA, bisa digunakan sebagai acuan untuk melakukan pencarian. Bila dibandingkan dengan tanpa klastering, dari hasil percobaan diperoleh tingkat akurasi sebesar 78 % serta penambahan kecepatan sebesar 23.93 %

Kata Kunci: *Algoritma Genetika, K-Means Clustering, CBIR, Pencarian Gambar.*

1. Pendahuluan

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. Tanpa adanya kemampuan untuk memeriksa *content* sebuah gambar, yang biasa digunakan dalam proses pencarian adalah *metadata* suatu gambar (misalnya, *captions* atau *keywords*) yang dimasukkan secara manual. Tentu saja cara ini tidak praktis, melelahkan, dan juga mahal (karena masih menggunakan tenaga manusia untuk memasukkan deskripsi gambar pada sistem database) [1].

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. Parameter feature gambar yang dapat digunakan untuk retrieval pada system ini seperti histogram, susunan warna, teksture, dan shape, tipe spesifik dari obyek, tipe event tertentu, nama individu, lokasi, emosi [10].

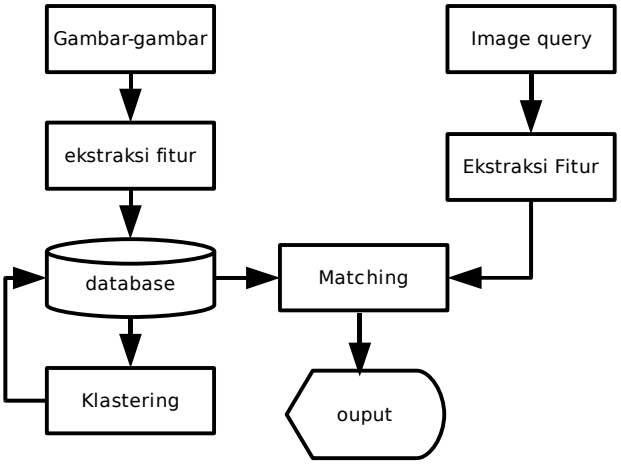
Beragam cara telah diajukan pada sistem CBIR ini. Di PENS-ITS, penulis setidaknya telah menemukan beberapa judul proyek akhir yang ada kaitannya dengan CBIR, lebih khususnya menggunakan fitur warna dan teknik klastering. Diantaranya adalah "Image Clustering Berdasarkan Warna Untuk Identifikasi Buah dengan Metode Hill Climbing", dan "Image Clustering Berdasarkan Warna Untuk Identifikasi Buah dengan Metode Valley Tracing" [5,6].

Pada dua penelitian itu, fitur warna yang dipakai adalah RGB. Sedang metode yang digunakan untuk melakukan ekstraksi fitur warna adalah teknik klastering. Pada dua proyek itu, seluruh nilai RGB tiap pixel pada sebuah gambar disegmentasi menjadi tiga bagian menggunakan K-Means. Bagian pertama diasumsikan mendekati warna putih (sebagai background), bagian kedua diasumsikan mendekati warna hitam (sebagai bayangan), dan warna ketiga diasumsikan warna obyek. Dua nilai pertama diabaikan, sedang sisanya diambil nilai rata-ratanya (untuk dijadikan *centroid*). Nilai *centroid* tadi kemudian disimpan dalam sebuah database. Demikian seterusnya hingga sampai beberapa gambar. Data-data itu kemudian disegmentasi lagi menggunakan algoritma klastering yang sudah dimodifikasi. Hasil segmentasi yang terakhir inilah yang dijadikan acuan untuk pencarian gambar.

Dari hasil percobaan, dua proyek akhir itu ternyata masih belum sempurna dan memiliki banyak kelemahan. Diantaranya adalah, bahwa sistem ini ternyata tidak mampu mengidentifikasi gambar-gambar yang memiliki beberapa warna dominan, tidak mampu mengidentifikasi gambar yang memiliki lebih dari satu obyek, tidak mampu menangani gambar yang memiliki ukuran berbeda, dan proses klastering yang sampai 2 kali yang diawali juga dengan pembersihan noise, menyebabkan proses komputasinya menjadi lama.

Dalam makalah ini, akan diuraikan tentang metode pencarian gambar yang menggunakan penanda hasil segmentasi sejumlah data yang didalamnya sudah tersimpan fitur warna (*HSV color histogram)*. Sedangkan teknik segmentasi yang dipakai adalah *Fast Genetic K-Means Algorithm (FGKA)*.

Secara umum, HSV lebih dipilih daripada RGB karena lebih mampu membedakan warna [4]. Sedangkan FGKA dipilih karena performanya lebih bagus bila dibandingkan dengan K-Means dan GKA [2,3].



Gambar 1: Blok Diagram Sistem

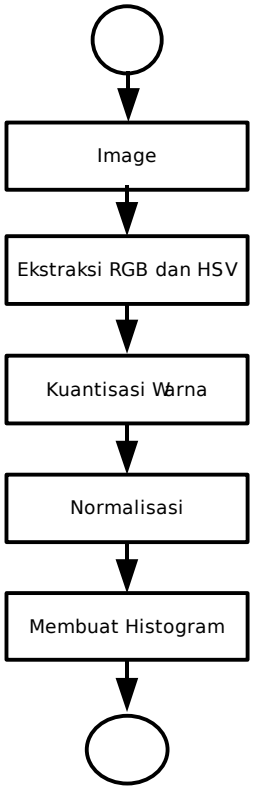
2. Metode Pencarian Gambar

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

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

2.1 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengambilan histogram, baik dari gambar database maupun gambar query.



Gambar 2: Diagram Blok Ektraksi Fitur

Tahap ini terdiri dari beberapa sub tahapan, yaitu:

a. Pengambilan nilai RGB tiap pixel yang kemudian langsung dikonversi ke HSV.

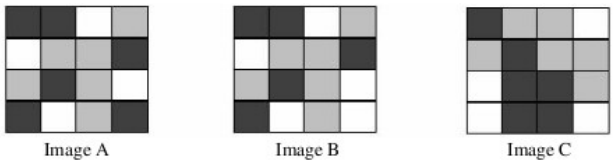
b. Kuantisasi warna dari yang semula berjumlah (360 x 255 x 255) atau 23409000 kemungkinan warna, diubah menjadi (4 x 4 x 4) atau 64 kemungkinan warna. Dengan cara ini, nilai H berkisar antara 0 sampai dengan 3, S berkisar antara 0 sampai dengan 3, dan V berkisar antara 0 sampai dengan 3.

c. Normalisasi.

d. Pembuatan HSV Histogram. Pada langkah ini juga dilakukan pembuatan Thumbnails yang berguna untuk menampilkan hasil pencarian dalam bentuk icon.

Tipe HSV Histogram yang dipakai pada makalah ini adalah GCH (Global Colour Histogram).Pada penggunaan GCH, distribusi warna global suatu gambar diambil dan digunakan sebagai metada. Jika pengguna mencari gambar dengan yang dalam sistem databasenya hanya memperhatikan distribusi warna global suatu gambar, memang, GCH adalah pilihan terbaik. Walaupun demikian, karena GCH hanya mengambil distribusi warna global suatu gambar sebagai pertimbangan untuk membandingkan gambar, ini bisa mengembalikan hasil yang tidak sesuai dengan persepsi visual [7].

Misalkan ada tiga gambar yang telah dikuantisasi menjadi tiga warna: hitam, abu-abu, dan putih (gambar 4.3). Misalkan gambar A adalah query image, sedangkan gambar B dan C adalah gambar-gambar dalam database.



Gambar 3: Tiga gambar yang terkuantisasi menjadi 3 warna

Image	Hitam	Abu-abu	Putih
A	37.5%	37.5%	25%
B	31.25%	37.5%	31.25%
C	37.5%	37.5%	25%

Tabel 1: GCH *Image* A, B, dan C

Sedangkan Distribusi warna (GCH) tiga gambar diatas adalah seperti pada tabel. Maka, jarak antara gambar A dengan gambar B dan C adalah:

$d(A,B) = |0.375 - 0.3125| + |0.375 - 0.375| + |0.25-0.3125|$

$= 0.125$

$d(A,C) = |0.375 - 0.375| + |0.375 - 0.375| + |0.25 - 0.25|$

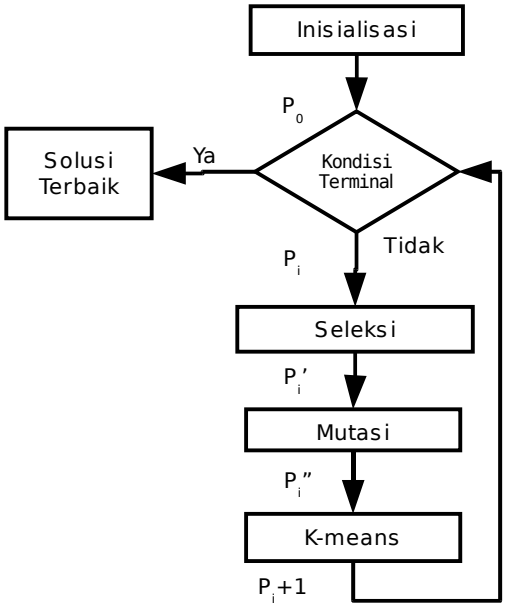
$= 0$

Dari hasil pembandingan, gambar C ternyata ditemukan lebih mirip daripada gambar B (karena jarak C lebih kecil). Padahal, sesuai dengan persepsi, yang lebih mirip dengan gambar A sebenarnya adalah gambar B [7].

GCH merepresentasikan keseluruhan bagian gambar dengan satu histogram. Sedangkan LCH membagi gambar menjadi beberapa bagian dan kemudian mengambil histogram warna tiap bagian tadi. LCH memang berisi lebih banyak informasi tentang gambar, namun metode ini membutuhkan lebih banyak proses komputasi [11, 12].

2.2 Klastering

Tahap ini merupakan implementasi dari algoritma FGKA untuk melakukan klasterisasi terhadap sejumlah HSV histogram, sesuai dengan kedekatan jarak (kemiripan) antara gambar-gambar.



Gambar 4: Diagram Blok FGKA

Tahap klastering di awali 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.

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,..., S_Z) yang mempunyai probabilitas (p_1, p_2, ..., p_Z) dengan definisi sebagai berikut:

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

Dari probabilitas ini, kemudian dilakukan penyeleksian menggunakan *Roullete Wheel*, yang dengan cara itu, kromosom dengan probabilitas yang tinggi akan bertahan untuk ikut diproses dalam operator selanjutnya [2,3]

2.2.2 Operator Mutasi

Pada operator ini, tiap kromosom dikodekan dengan $a_1\ a_2 \ldots\ a_N$ dan operator mutasi melakukan mutasi pada suatu gen a_n (n = 1...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, .. ,K) dengan distribusi (p_1, p_2, ..., 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 klaster 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 ... a_N. Operator ini akan mengganti isi dari gen a_n (n = 1...N) dengan nilai baru a_n', dimana nilai yang baru merupakan klaster dengan jarak terpendek dari data a_n yang dihitung menggunakan rumus Euclidean [2,3]

2.3 Matching

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

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

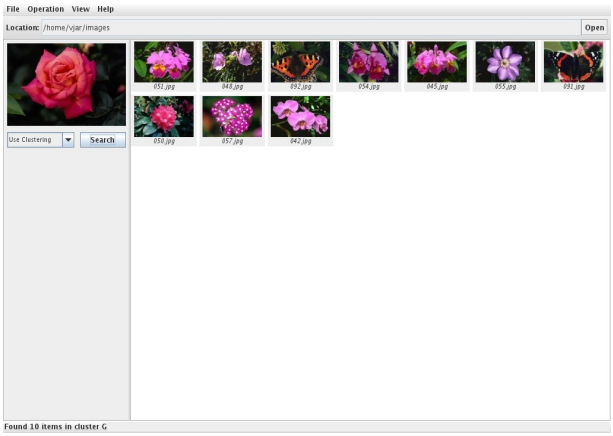
$$d(A,B) = \sqrt{\sum_{j=1}^{n} \left(H_j^A - H_j^B\right)^2}$$

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 euclidan 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 searching.

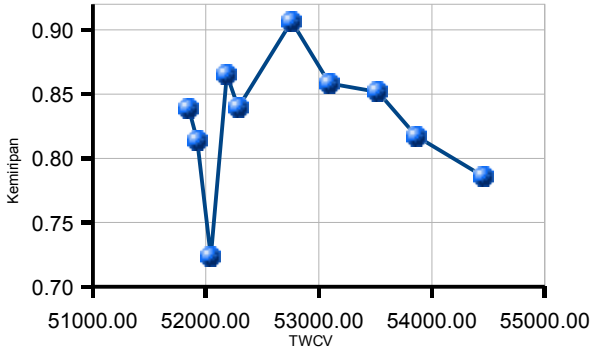
Gambar 5: Gambar Query



Gambar 6: Hasil Pencarian

3.1 Hasil Klastering

Dari percobaan, diperoleh bahwa besarnya TWCV tidak berkorelasi dengan tingkat keseuaian antar data dalam satu klaster.



Gambar 7: Grafik Hasil Klastering

Dari hasil perhitungan korelasi, diperoleh bahwa nilai TWCV hanya mempunyai pengaruh sebesar 0.03% saja terhadap tingkat kemiripan dan sisanya sebesar 99.97%, nilai kemiripan ini dipengaruhi oleh faktor bukan TWCV.

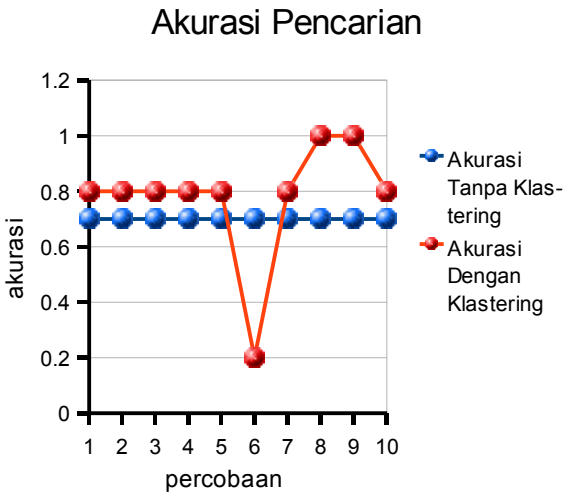
Dari percobaan juga diperoleh bahwa hasil yang dikembalikan kadang-kadang tidak sesuai dengan persepsi visual. Ini terjadi karena informasi yang disimpan dalam sistem merupakan *HSV histogram* yang merepresentasikan keseluruhan gambar (GCH) tanpa memperhatikan komposisi tiap bagian-bagian gambar (misalnya bagian pojok-pojok dan tengah gambar). Jadi, meskipun secara numerik jarak antar histogram sudah dikelompokkan dengan nilai TWCV kecil, secara visual belum tentu sama.

3.2 Hasil Searching

Pembahasan hasil searching ini ada dua yaitu Akurasi Klastering dan Waktu pencarian menggunakan klastering dan tanpa menggunakan klastering.

3.2.1 Akurasi Klastering vs Tanpa Klastering

Dari perhitungan korelasi hasil 10 kali percobaan, diperoleh bahwa dekatnya jarak centroid dengan gambar query, hanya mempunyai pengaruh sebesar 0.33% saja terhadap tingkat akurasi dan sisanya sebesar 99.67%, nilai akurasi dipengaruhi oleh faktor bukan jarak. Dari hasil percobaan, juga diperoleh bahwa hasil yang dikembalikan sistem kadang-kadang tidak sesuai dengan persepsi visual. Seperti pada hasil klastering, ini terjadi karena informasi yang disimpan dalam sistem merupakan *HSV histogram* yang merepresentasikan keseluruhan gambar (GCH) tanpa memperhatikan komposisi tiap bagian-bagian gambar (misalnya bagian pojok-pojok dan tengah gambar). Jadi, meskipun secara numerik jarak antara gambar query dengan gambar database sudah minimal, secara visual belum tentu mirip.

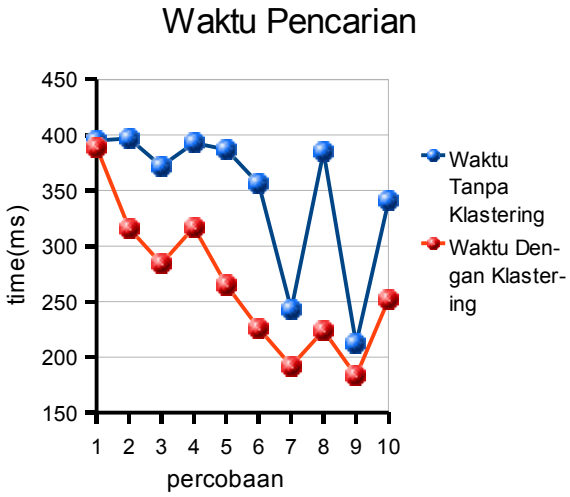


Gambar 8: Grafik Perbandingan Akurasi

Terkait dengan tingkat akurasi dengan dan tanpa klastering, dari hasil percobaan, dapat dilihat bahwa tingkat akurasi tanpa klastering pada sepuluh kali percobaan selalu tetap dengan nilai sebesar 0.7.

Bila dibandingkan dengan pencarian dengan klastering, hasilnya masih lebih rendah dengan tingkat akurasi rata-rata pengujian pencarian dengan klastering yang mencapai 0.78. Ini sesuai dengan yang diharapkan bahwa gambar yang dikembalikan pada sistem pencarian dengan klastering adalah gambar-gambar yang sudah terkelompok dan tidak tercampur.

3.2.2 Waktu Pencarian Klastering vs Tanpa Klastering



Gambar 4.9: Grafik Perbandingan Waktu

Sedangkan terkait dengan waktu, jika dua percobaan itu dibandingkan, didapatkan bahwa ternyata waktu yang diperlukan pada proses pencarian menggunakan klasterning lebih cepat hingga 23.93 %. Ini sesuai dengan yang diharapkan bahwa waktu yang diperlukan untuk melakukan pencarian dengan menggunakan klastering akan lebih cepat bila dibandingkan dengan tanpa klastering karena data-data sudah tersegmentasi.

4. Kesimpulan

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

1. FGKA dapat digunakan pada proses pencarian gambar dengan terlebih dahulu mengelompokkan gambar-gambar yang memiliki nilai *HSV histogram* yang berdekatan. Dengan cara ini, pada beberapa pengujian, hasil yang dikembalikan ternyata tidak tercampur.
2. Karena GCH *(Global Colour Histogram)* hanya mengambil distribusi warna global suatu gambar sebagai pertimbangan untuk membandingkan gambar, hasil yang dikembalikan pada sistem pencarian proyek ini kadang tidak sesuai dengan persepsi visual.
3. Hasil klastering yang berubah-ubah mengakibatkan hasil pencarian juga ikut berubah. Akibatnya, meskipun jarak terdekat gambar query dan centroid telah didapatkan, hal itu tidak menjamin bahwa gambar-gambar pada klaster centroid tersebut itu punya gambar-gambar yang mirip dengan gambar query.

Untuk lebih meningkatkan akurasi hasil pencarian pada sistem CBIR (khususnya fitur warna), perlu dipertimbangkan penggunaan LCH *(Local Colour Histogram)*, karena secara teoritis, informasi yang di dapat dari gambar dengan histogram tipe ini lebih banyak.

Daftar Pustaka

[1] Anonym, *"Content-based image retrieval,"* http://en.wikipedia.org/wiki/Content-based_image_retrieval

[2] Entin Martiana, "Perbaikan Kinerja Algoritma Klusterisasi K-Means Genetika," FTIF-ITS.

[3] Yi Lu, Shiyong Lu, Farshad Fotouhi, Youping Deng, and Susan Brown, "*Fast Genetic K-means Algorithm and its Application in Gene Expression Data Analysis*," Technical Report TR-DB-06-2003, Department of Computer Science, Wayne State University, June, 2003.

[4] Wen Chen, Yun Q. Shi, and Guorong Xuan, "Identifying Computer Graphics Using HSV Color Model and Statistical Moments of Characteristic Functions."

[5] Gedhe Wiryana Wardana, "Image Clustering Berdasarkan Warna Untuk Identifikasi Buah dengan Metode Hill Climbing,"Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2007.

[6] Helmy Hasniawati, "Image Clustering Berdasarkan Warna Untuk Identifikasi Buah dengan Metode Valley Tracing," Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2007.

[7] Yue Zhang, "On the use of CBIR in Image Mosaic Generation," Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada , 2002.

[8] Anonym, "Color histogram," http://en.wikipedia.org/wiki/Color_histogram.

[9] D. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, MA: Addison-Wesley, 1989.

[10] Bayu Bagus, "Image Database Menggunakan Sistem Content Based Image Retrieval dengan Ekstraksi Fitur Terstuktur," Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, 2007.

[11] Shengjiu Wang, "A Robust CBIR Approach Using Local Color Histograms," Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, Tech. Rep. TR 01-13, October 2001

[12] Rami Al-Tayeche dan Ahmed Khalil, "CBIR: Content Based Image Retrieval," Department of Systems and Computer Engineering, Faculty of Engineering, Carleton University, 2003.

[13] Achmad Basuki, "Algoritma Genetika: Suatu Alternatif Penyelesaian Permasalahan, Optimasi, dan Machine Learning", *Handout* Mata Kuliah, Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember.

[14] Ali Ridho Barakbah, "Clustering", *Handout* di workshop data mining, Jurusan Teknologi Informasi, Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember, Juli 2006.