PROYEK AKHIR

PENCARIAN GAMBAR BERDASARKAN FITUR WARNA DENGAN GA-KMEANS CLUSTERING

Yanu Widodo
NRP. 7406 040 064

Dosen Pembimbing

Entin Martiana Kusumaningtyas, M.Kom
NIP. 132 282 692

Achmad Basuki, M.Kom
NIP.132 093 221

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
SURABAYA 2008

Halaman ini sengaja dikosongkan

PENCARIAN GAMBAR BERDASARKAN FITUR WARNA DENGAN GA-KMEANS CLUSTERING

Oleh:
Yanu Widodo
7406040064

Proyek akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains Terapan (S.ST)
di
Politeknik Elektronika Negeri Surabaya,
Institut Teknologi Sepuluh Nopember

Disetujui oleh:

Dosen Penguji I	Dosen Pembimbing I
Setiawardhana, S.T	**Entin Martiana Kusumaningtyas, M.Kom**
NIP.132 310 243	**NIP. 132 282 692**
Dosen Penguji II	Dosen Pembimbing II
Arif Basofi, S.Kom	**Achmad Basuki, M.Kom**
NIP. 132 303 872	**NIP.132 093 221**
Dosen Penguji III	

Yuliana Setiowati, S.Kom
NIP.132 300 678

Mengetahui,
Kepala Jurusan Teknologi Informasi

Arna Fariza, S. Kom, M. Kom
NIP. 132 233 198

Halaman ini sengaja dikosongkan

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.

ABSTRACT

In many areas of commerce, government, academia, and hospitals, large collections of digital images are being created. Many of these collections are the product of digitizing existing collections of analogue photographs, diagrams, drawings, paintings, and prints. Usually, the only way of searching these collections was by using meta data (like caption or keywords). This way is impractical, laborious, and expensive (requires humans to personally describe every image in the database.)

In recent year, have been developed many ways in image retrieval that use image content (color, shape, and texture). The use of centroid produced from clustered HSV Histogram using FGKA, can be used for searching parameter. If compared with non clustering data, from experimental result the accuracy obtained is 48 % and increasing in searching time is 23.93%

Keywords: K-Means Clustering, Genetic Algorithm, CBIR, Image Searching.

KATA PENGANTAR

Segala puji bagi Allah *subhana wa Ta'ala* yang telah mengaruniakan nikmat-Nya, sehingga telah sampai penulis pada bagian paling menyenangkan dan penulis nantikan sejak lama, yaitu menuliskan kata pengantar pada proyek akhir yang berjudul:

PENCARIAN GAMBAR BERDASARKAN FITUR WARNA DENGAN GA-KMEANS CLUSTERING

Seperti tersirat dari judulnya, topik yang dibahas pada buku ini memang sangat terkait dengan *image retrieval* dan *machine learning*. Judul di atas, berawal dari sebuah *handout* yang diberikan Pak Ali Ridho Barakbah berjudul: *Nature Based Image Searching*. Judul itu kemudian diubah hingga seperti sekarang atas masukan dari Pak Achmad Basuki. Karena Pak Ali mendapatkan tugas belajar ke Jepang, proses bimbingan proyek akhir ini kemudian diteruskan oleh Bu Entin dan Pak Basuki.

Dalam proyek akhir ini, Bu Entin sangat berperan membantu penulis dalam memahami kosep FGKA. Sedangkan Pak Basuki sangat berperan membantu penulis dalam memahami konsep CBIR. Dengan kehadiran beliau berdua, penulis seperti disinari dua pelita di tengah ruangan yang gelap.

Akhir kata, tiada gading yang tak retak dan tak ada sesuatu yang tiada cela. Maka demikian juga apa yang telah penulis tuliskan, masih banyak kekurangan di sana-sini. Penulis berharap semoga buku sederhana yang telah penulis susun ini bisa bermanfaat bagi pembacanya.

Surabaya, Agustus 2008

Yanu Widodo

Halaman ini sengaja dikosongkan

UCAPAN TERIMA KASIH

Melalui kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

1. Ibu dan Bapak di rumah yang senantiasa membantu dengan perhatian, doa, dan tenaga yang tidak ternilai harganya.
2. Saudara-saudaraku di rumah yang selalu memberikan dukungan dan semangat.
3. Bapak Dr. Titon Dutono, M.Eng selaku direktur Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember.
4. Ibu Arna Fariza, S. Kom, M. Kom selaku Ketua Jurusan Teknologi Informasi PENS-ITS
5. Ibu Entin Martiana K, S. Kom, M. Kom yang telah memberikan banyak pencerahan tentang konsep *Fast Genetic K-Means Algorithm* dan masukan berharga atas draft laporan yang penulis ajukan.
6. Bapak Ahmad Basuki, M. Kom yang telah memberikan konsep dengan sangat jelas tentang model warna, konsep dan format histogram, serta konsep dasar CBIR.
7. Bapak Ali Ridho Barakbah yang telah memberikan topik proyek akhir yang menarik, bimbingan jarak jauh, motivasi, dan semangat.
8. Bapak Setiawardhana S.T, Bapak Arif Basofi S.Kom, Ibu Yuliana S.Kom selaku dosen penguji
9. Rekan-rekan D4 IT Lintas Jalur 2006 kelas A, yang sering saya mintai bantuan: Fajar, terima kasih atas laptop dan tempat menginapnya. Akmal, terima kasih atas printer dan tintanya. Dan Lika, terima kasih atas sharing info jadwalnya.
10. Rekan-rekan D3 IT 2005 di laboratorium database atas bantuan printernya.

11. Rekan-rekan di Gebang Kidul 31. Amik, Aryo, Mera, Fahru, Ade dan kawan-kawan yang lain.

Serta semua pihak yang tidak tersebutkan di sini. Semoga semuanya dibalas dengan kebaikan yang banyak.

DAFTAR ISI

ABSTRAK.....v
KATA PENGANTAR.....vii
UCAPAN TERIMA KASIH.....ix
DAFTAR ISI.....xi
DAFTAR GAMBAR.....xiii
DAFTAR TABEL.....xv
BAB I PENDAHULUAN.....1
1.1 Latar Belakang.....1
1.2 Tujuan.....2
1.3 Permasalahan 3
1.4 Batasan Masalah.....3
1.5 Metodologi.....3
1.6 Sistematika Pembahasan.....4
BAB II DASAR TEORI.....5
2.1 Fitur Warna.....5
2.1.1 Model Warna.....5
2.1.2 HSV dan RGB.....6
2.1.3 Color Histogram.....7
2.1.3.1 Color quantization.....8
2.1.3.2 Normalisasi.....9
2.2 Content Based Image Retrieval.....10
2.2.1 Pengukuran Jarak Antar Dua Histogram.....10
2.2.2 Tipe Colour Histogram.....11
2.3 K-Means.....13
2.3.1Algoritma K-Means.....13
2.3.2 Karakteristik K-Means.....14
2.4 Algoritma Genetika.....14
2.4.1 Beberapa Definisi Penting Algoritma Genetika.....15
2.4.2 Siklus Algoritma Genetika.....15
2.4.2 Hal-hal yang Harus Diperhatikan dalam Algoritma Genetika. 17
2.6 Fast Genetic K-Means Algorithm 18
2.3.1 Inisialisasi Awal.....19
2.3.2 Fungsi Fitness.....20
2.3.3 Operator Seleksi 21
2.3.4 Operator Mutasi.....21

2.3.5 Operator K-means 22
BAB III PERANCANGAN DAN PEMBUATAN SISTEM.....23
3.1 Blok Diagram Sistem.....23
3.1.1 Ekstraksi Fitur Gambar.....23
3.1.2 Klastering.....27
3.1.2.1 Operator Seleksi.....27
3.1.2.2 Operator Mutasi.....28
3.1.2.3 Operator K-Means.....29
3.1.3 Matching (Pencocokan).....30
3.2 Pembuatan GUI.....31
3.2.1 Tampilan Ketika Runtime.....32
3.2.2 Menampilkan Hasil Klastering.....36
3.2.3 Menampilkan Hasil Pencarian.....36
BAB IV PENGUJIAN DAN ANALISA DATA.....39
4.1 Lingkungan Uji Coba.....39
4.2 Parameter Uji Coba.....39
4.3 Skenario Uji Coba.....39
4.4 Hasil Pengujian.....40
4.4.1 Hasil Klustering.....40
4.4.2 Hasil Pencarian.....50
4.5 Analisa Data 61
4.5.1 Analisa Data Hasil Klastering.....61
4.5.2 Analisa Data Hasil Pencarian.....63
BAB V PENUTUP.....67
5.1 Kesimpulan.....67
5.2 Saran.....67
DAFTAR PUSTAKA.....69
TENTANG PENULIS.....71

DAFTAR GAMBAR

Gambar 2.1: Gambar Berwarna "A" dan Histogramnya.....8
Gambar 2.2: Ukuran beda, tapi distribusi warna sama 9
Gambar 2.3: Tiga gambar yang terkuantisasi menjadi 3 warna.....12
Gambar 2.4: Ilustrasi Algoritma K-Means pada Data 2 Dimensi.....13
Gambar 2.5: Ilustrasi Kelemahan K-Means.....14
Gambar 2.6: Siklus Algoritma Genetika.....16
Gambar 2.8: Mutasi.....17
Gambar 2.7: Cross Over.....17
Gambar 3.1: Diagram Blok System.....23
Gambar 3.2: Diagram Blok Ektraksi Fitur.....24
Gambar 3.3 Ekstraksi HSV.....25
Gambar 3.4 Kuantisasi.....25
Gambar 3.5 Normalisasi.....26
Gambar 3.6: Diagram Blok FGKA.....27
Gambar 3.7: Flow Chart Roulete Wheel.....28
Gambar 3.8: Flow Chart Operator Mutasi.....29
Gambar 3.9: Flow Chart K-Means Operator.....30
Gambar 3.10 : Flow Chart Proses Matching.....31
Gambar 3.11: Tampilan GUI simple CBIR.....32
Gambar 3.13: Open File dengan image preview.....33
Gambar 3.14: Menu Operation.....33
Gambar 3.12: Menu File.....33
Gambar 3.15: View Menu 34
Gambar 3.16: Location Tool Bar.....34
Gambar 3.17: Command Panel.....35
Gambar 3.18: Result Panel.....35
Gambar 3.19: Status Bar.....35
Gambar 3.20: Progess Information Dialog.....35
Gambar 3.21: Logging.....36
Gambar 3.22: Contoh gambar query.....36
Gambar 3.23: Hasil Klastering.....37
Gambar 3.24: Tampilan Hasil Pencarian.....37
Gambar 4.1: Kluster A Hasil Percobaan 1.....40
Gambar 4.2: Klaster A Hasil Percobaan 2.....41
Gambar 4.3: Klaster A Hasil Percobaan 3.....42

Gambar 4.4: Klaster A Hasil Percobaan 4.....43
Gambar 4.5: Klaster A Hasil Percobaan 5.....44
Gambar 4.6: Klaster A Hasil Percobaan 6.....45
Gambar 4.7: Klaster A Hasil Percobaan 7.....46
Gambar 4.8: Klaster A Hasil Percobaan 8.....47
Gambar 4.9: Klaster A Hasil Percobaan 9.....48
Gambar 4.10: Klaster A Hasil Percobaan 10.....49
Gambar 4.11: Hasil Pencarian Percobaan 1.....50
Gambar 4.12: Hasil Pencarian Percobaan 2.....51
Gambar 4.14: Hasil Pencarian Percobaan 4.....53
Gambar 4.16: Hasil Pencarian Percobaan 6.....55
Gambar 4.17: Hasil Pencarian Percobaan 7.....56
Gambar 4.18: Hasil Pencarian Percobaan 8.....57
Gambar 4.19: Hasil Pencarian Percobaan 9.....58
Gambar 4.20: Hasil Pencarian Percobaan 10.....59
Gambar 4.21: Hasil Pencarian Tanpa Clustering.....60
Gambar 4.22: Grafik TWCV-Rata-rata Kesesuaian.....62
Gambar 4.23: Grafik Perbandingan Akurasi.....64
Gambar 4.24: Grafik Perbandingan Waktu.....64

DAFTAR TABEL

Tabel 2.1: Color Histogram gambar "A".....9
Tabel 2.2: GCH Image A, B, dan C.....12
Tabel 4.1: Hasil Klastering Percobaan 1.....40
Tabel 4.2: Hasil Klastering Percobaan 2.....41
Tabel 4.3: Hasil Klastering Percobaan 3.....42
Tabel 4.4: Hasil Klastering Percobaan 4.....43
Tabel 4.5: Hasil Klastering Percobaan 5.....44
Tabel 4.6: Hasil Klastering Percobaan 6.....45
Tabel 4.7: Hasil Klastering Percobaan 7.....46
Tabel 4.8: Hasil Klastering Percobaan 8.....47
Tabel 4.9: Hasil Klastering Percobaan 9.....48
Tabel 4.10: Hasil Klastering Percobaan 10.....49
Tabel 4.11: Hasil Pencarian Percobaan 1.....50
Tabel 4.12: Hasil Pencarian Percobaan 2.....51
Tabel 4.13: Hasil Pencarian Percobaan 3.....52
Tabel 4.14: Hasil Pencarian Percobaan 4.....53
Tabel 4.15: Hasil Pencarian Percobaan 5.....54
Tabel 4.16: Hasil Pencarian Percobaan 6.....55
Tabel 4.17: Hasil Pencarian Percobaan 7.....56
Tabel 4.18: Hasil Pencarian Percobaan 8.....57
Tabel 4.19: Hasil Pencarian Percobaan 9.....58
Tabel 4.20: Hasil Pencarian Percobaan 10.....59
Tabel 4.21: Hasil Pencarian Tanpa Klastering.....60
Tabel 4.23: Hasil Klastering Keseluruhan Percobaan.....61
Tabel 4.24: Statistik Regresi Hasil Klastering.....62
Tabel 4.25: Akurasi dan Jarak Gambar Query-Centroid.....63
Tabel 4.26: Statistik Regresi Hasil Pencarian.....63
Tabel 4.27: Perbandingan Hasil Pencarian.....65

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 Latar Belakang

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 proyek 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. Sedangkan FGKA dipilih karena performanya lebih bagus bila dibandingkan dengan K-Means dan GKA. Pada bab dua nanti akan dijelaskan lebih lanjut beberapa teori singkat tentang *color histogram*, CBIR, dan FGKA.

1.2 Tujuan

Tujuan dari proyek akhir ini adalah mengetahui hasil implementasi teknik klustering, yaitu FGKA, dalam pencarian gambar menggunakan fitur warna *(HSV Histogram)*.

1.3 Permasalahan

Pada proses pencarian dengan membandingkan histogram gambar query dengan histogram gambar-gambar database secara satu-satu, sistem kadangkala mengembalikan gambar-gambar yang secara numerik memiliki jarak antar histogram yang kecil. Namun secara visual, gambar-gambar itu bercampur.

Tugas akhir ini akan membahas permasalahan tentang bagaimana cara agar HSV histogram pada suatu gambar-gambar database tersebut dikelompokkan terlebih dahulu, sehingga hasil yang dikembalikan tidak bercampur dengan gambar jenis lain.

1.4 Batasan Masalah

Batasan masalah pada proyek akhir ini adalah:

1. Type gambar: jpeg, png, gif.
2. Ukuran maksimal: 400 x 400 pixel.
3. *Image Content* yang digunakan: HSV Color Histogram.
4. Algoritma Klastering: FGKA
5. *Operating System*: Linux Ubuntu 7.10 - *the Gutsy Gibbon - released in October 2007.*

1.5 Metodologi

Metodologi yang diterapkan dalam makalah ini adalah:

1. Studi Literatur
   Tahapan ini dilakukan untuk mendapatkan segala informasi yang terkait dengan topik tugas akhir ini: hasil penelitian sebelumnya dan teori penunjangnya.
2. Perencanaan Sistem
   Tahapan ini dilakukan untuk merencakan sistem yang akan dibangun. Meliputi penyusunan proses ektraksi fitur warna, proses segmentasi nilai-nilai yang didapat, *query* gambar, proses pencarian, dan proses penampilan gambar.
3. Penerapan Algoritma
   Tahapan ini dilakukan untuk melakukan testing awal algoritma yang sudah direncanakan. Pada tahap ini, tampilan hasil akhir masih berupa teks-teks terminal.
4. Pembuatan Sistem
   Tahapan ini dilakukan untuk melakukan pengembangan aplikasi

yang dihasilkan pada tahapan nomer tiga. Hasil akhir dari tahapan ini berupa aplikasi berbasis GUI yang lebih mudah untuk dibaca.

5. Pengujian dan Analisa Data
   Tahapan ini dilakukan untuk mengetahui performa dan hasil akhir sistem yang telah dikembangkan. Hasil dari tahapan ini akan digunakan untuk mengambil kesimpulan tahap selanjutnya.

1.6 Sistematika Pembahasan

Adapun sistematika pembahasan buku ini adalah:

1. Bab Pendahuluan
   Berisi tentang latar belakang, tujuan, batasan masalah proyek akhir ini.
2. Bab Teori Penunjang
   Berisi tentang uraian teori-teori singkat yang berkaitan dengan Fitur Warna, CBIR, K-Means, Algoritma Genetika, dan FGKA.
3. Bab Perencanaan dan Pembuatan Sistem
   Berisi tentang hal-hal yang terkait dengan proses perencanaan dan pembuatan sistem yang meliputi proses mendapatkan *hsv color histogram,* implementasi algoritma FGKA, pencarian gambar, serta cara menampilkan hasil klustering dan pencarian.
4. Bab Pengujian dan Analisa
   Berisi tentang hal-hal yang terkait dengan hasil pengujian sistem dengan sekian banyak data hingga beberapa kali. Hasilnya kemudian dianalisa untuk menjadi pertimbangan dalam pengambilan kesimpulan.
5. Bab Penutup
   Berisi tentang kesimpulan akhir dari keseluruhan percobaan. Apakah sistem ini berhasil seperti yang diharapkan, adakah yang perlu diperbaiki dan sebaagainya. Porsi terbesar bagian terakhir ini diambil dari bab pengujian dan analisa.

BAB II
DASAR TEORI

Pada bab ini akan dibahas secara ringkas dasar teori yang menunjang dalam pembuatan tugas akhir, yaitu: Fitur Warna, CBIR, Algoritma K-means, Algoritma Genetika, dan *Fast Genetic K-Means Algorithm*.

2.1 Fitur Warna

Selain bentuk dan textur, warna merupakan salah satu *image contents* yang sering digunakan pada kebanyakan sistem CBIR.

2.1.1 Model Warna

Model warna *(color model)* adalah sejumlah cara untuk merepresentasikan warna yang diindera manusia. Model warna yang digunakan saat ini dapat digolongkan ke dalam dua kategori: *hardware-oriented* dan *user-oriented*.

Model warna *hardware-oriented* banyak digunakan untuk warna alat-alat. Misalnya model warna RGB (red, green, blue), biasa digunakan untuk warna monitor dan kamera. Model warna CMY (cyan, magenta, yellow), digunakan untuk warna printer; dan warna YIQ digunakan untuk penyiaran tv warna. Sedangkan model warna yang u*ser-oriented* termasuk HLS, HCV, HSV, MTM, dan CIE-LUV, didasarkan pada tiga persepsi manusia tentang warna, yaitu *hue* (keragaman warna), *saturation* (kejenuhan), dan *brightness* (kecerahan) [7].

Berikut penjenjelasan ringkas tentang berbagai macam model atau format warna:

- Format warna RGB
  Format ini digunakan untuk menghasilkan warna di monitor dan televisi tabung yang menggunankan gelombang elektromagnetik. Sebuah titik ditembak dengan spektrum R, G dan B.
- Format warna HSV atau HSI atau HSL
  Format ini merupakan format warna alamiah dengan mempertimbangkan bahwa spektrum warna adalah sebuah koordinat polar seperti warna pantulan yang jatuh di mata manusia.

Format ini sangat baik untuk membedakan warna-warna yang 'terlihat'.

- Format warna CIE
  Format warna ini adalah varians dari RGB dengan normalisasi spektrum, sehingga sifat orthogonalitas dari masing-masing komponen warna lebih dijamin. FOrmat ini merupakan standard dalam QBIC
- Format warna YCrCb
  Format warna ini disebut juga dengan warna chrominant. Format ini banyak digunakan dalam skin-detection.
- Format warna CMYK
  Format warna ini adalah penghasil warna pada cat atau tinta. Format warna ini yang digunakan oleh mesin cetak.

2.1.2 HSV dan RGB

Model warna RGB merupakan yang paling banyak digunakan pada sistem CBIR. Pada model ini, warna direpresentasikan menjadi tiga warna primer, yaitu: red, green, dan blue [8]. Nilai masing-masing warna primer itu berkisar antara 0 - 255. Sedangkan HSV (hue, saturation, value) merupakan model warna yang diturunkan dari RGB.

Berikut ini rumus mengkonversi nilai-nilai RGB menjadi HSV:

Rumus untuk menentukan h:

$$h = \begin{cases} 0 & \text{if } \max = \min \\ 60° \times \frac{g-b}{\max - \min} + 0°, & \text{if } \max = r \text{ and } g \geq b \\ 60° \times \frac{g-b}{\max - \min} + 360°, & \text{if } \max = r \text{ and } g < b \\ 60° \times \frac{b-r}{\max - \min} + 120°, & \text{if } \max = g \\ 60° \times \frac{r-g}{\max - \min} + 240°, & \text{if } \max = b \end{cases}$$

Rumus untuk menentukan s:

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

Rumus untuk menentukan v:

v = max

Literatur [4] menunjukkan bahwa performa HSV ternyata lebih baik dalam membedakan warna jika dibandingkan dengan RGB.

2.1.3 *Color Histogram*

Color histogram adalah representasi distribusi warna dalam sebuah gambar yang didapatkan dengan menghitung jumlah pixel dari setiap bagian range warna, secara tipikal dalam dua dimensi atau tiga dimensi [8].

Misalnya ada sebuah gambar berukuran 3x3 pixel dengan nilai RGB sebagai berikut:

(1,1,1) (1,2,0) (1,2,0)

(1,1,0) (2,1,0) (2,3,1)

(3,2,1) (2,2,1) (2,1,0)

Bila yang digunakan adalah format H(r,g,b) dimulai dari H(0,0,0) s/d H(3,3,3) , maka histogram gambar tersebut adalah sebagai berikut :

H(0,0,0)=0 H(0,0,1)=0 H(0,0,2)=0 H(0,0,3)=0

H(0,1,0)=0 H(0,1,1)=0 H(0,1,2)=0 H(0,1,3)=0

H(0,2,0)=0 H(0,2,1)=0 H(0,2,2)=0 H(0,2,3)=0

H(0,3,0)=0 H(0,3,1)=0 H(0,3,2)=0 H(0,3,3)=0

H(1,0,0)=0 H(1,0,1)=0 H(1,0,2)=0 H(1,0,3)=0

H(1,1,0)=1 H(1,1,1)=1 H(1,1,2)=0 H(1,1,3)=0

H(1,2,0)=1 H(1,2,1)=0 H(1,2,2)=0 H(1,2,3)=0

H(1,3,0)=0 H(1,3,1)=0 H(1,3,2)=0 H(1,3,3)=0

H(2,0,0)=0 H(2,0,1)=0 H(2,0,2)=0 H(2,0,3)=0

H(2,1,0)=2 H(2,1,1)=0 H(2,1,2)=0 H(2,1,3)=0

H(2,2,0)=0 H(2,2,1)=1 H(2,2,2)=0 H(2,2,3)=0

H(2,3,0)=0 H(2,3,1)=1 H(2,3,2)=0 H(2,3,3)=0

H(3,0,0)=0 H(3,0,1)=0 H(3,0,2)=0 H(3,0,3)=0

H(3,1,0)=0 H(3,1,1)=0 H(3,1,2)=0 H(3,1,3)=0

H(3,2,0)=0 H(3,2,1)=1 H(3,2,2)=0 H(3,2,3)=0

H(3,3,0)=0 H(3,3,1)=0 H(3,3,2)=0 H(3,3,3)=0

Jika ditulis, histogram dari data-data diatas adalah:

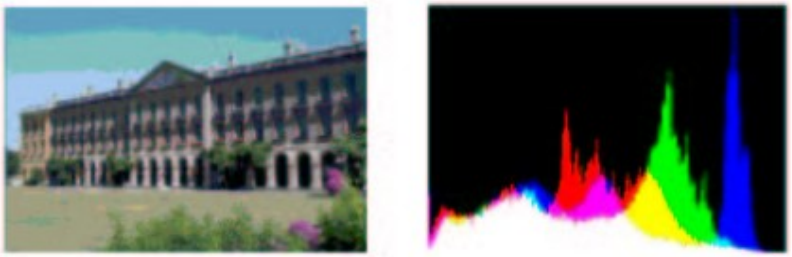
H = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}

2.1.3.1 *Color quantization*

Dalam pembuatan histogram, nilai RGB yang punya range dari 0 sampai 255 akan punya kemungkinan kombinasi warna sebesar 16777216 (didapat dari: 255 x 255 x 255). Pada proses komputasi, tentu saja ini proses yang *time consuming*.

Masalah tersebut dapat diatasi dengan *color quantization* (kuantisasi warna), yaitu suatu prosedur untuk mengurangi kemungkinan jumlah warna. Dengan cara ini, jumlah warna yang besar tadi bisa dikurangi, sehingga proses yang dibutuhkan akan semakin mudah [7].

Misalnya, nilai sebuah pixel RGB adalah (260, 200, 150). Maka setelah melalui kuantisasi menjadi 64 warna, misalnya, range H: 0-3, range S: 0-3, dan range V: 0-3, nilai itu menjadi (260 * 4/255, 200 * 4/255, 150 * 4/255) atau (3,3,2).



Gambar 2.1: Gambar Berwarna "A" dan Histogramnya

Colour	Number of Pixels
(0,0,0)	234
(0,0,1)	23
(0,0,2)	478
...	...
...	...
...	...
(3,3,3)	3429

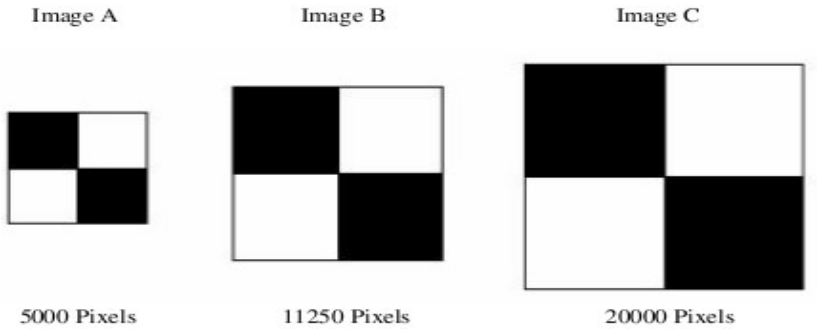
Tabel 2.1: Color Histogram gambar "A"

Gambar 2.1 dan tabel 2.1 menunjukkan bahwa gambar "A" yang telah melalui proses kuantisasi menjadi 64 warna. Sesuai dengan distribusi warna pada tiap pixel, *color histogram* gambar "A" adalah sebagai berikut: $H^A = \{234, 23, 478, ..., 3429\}$.

2.1.3.2 Normalisasi

Penggunaan nilai-nilai aktual distribusi warna pada distogram, membuat untuk dipahami. Namun pemakaian dengan cara ini akan menimbulkan masalah jika diterapkan pada gambar yang mempunyai ukuran berbeda namun seebenarnya mempunyai distribusi warna yang sama.

Sebagai contoh, misalnya ada 3 gambar dengan ukuran berbeda yang terkuantisasi menjadi 2 warna (hitam dan putih) - lihat gambar 2.



Gambar 2.2: Ukuran beda, tapi distribusi warna sama

Histogram 3 gambar ini adalah:

$H^A = \{2500,2500\}$

$H^B = \{5625, 5625\}$

$H^C = \{10000,1000\}$

Seperti dilihat, meskipun ketiga gambar tadi mempunyai distribusi warna yang sama, tapi mempunyai histogram yang berbeda. Ini dikarenakan perbedaan ukuran gambar (dan tentu saja jumlah pixel). Oleh karena itu, untuk membuat histogram tetap sama pada gambar yang mempunyai kesamaan distribusi warna, maka diperlukan suatu normalisasi histogram. Alih-alih menggunakan jumlah aktual, lebih baik menggunakan persentase pembagian jumlah aktual dengan jumlah total pixel gambar yang digunakan *color histogram.*

Dengan cara ini, selama distribusi warna pada gambar sama, histogram warnanya akan sama, tidak tergantung lagi pada ukuran gambar. Berikut adalah hasil histogram warna pada gambar 2 yang sudah ternormalisasi:

$$H^A = \{50\%, 50\%\}$$

$$H^B = \{50\%, 50\%\}$$

$$H^C = \{50\%, 50\%\}$$

2.2 Content Based Image Retrieval

Content Based Image Retrieval System (CBIR) merupakan suatu teknik pencarian kembali gambar yang mempunyai kemiripan karakteristik atau *content* dari sekumpulan gambar. Proses umum dari CBIR adalah gambar yang menjadi query dilakukan proses ekstraksi fitur, begitu halnya dengan gambar yang ada pada sekumpulan gambar juga dilakukan proses seperti pada gambar query. Fitur gambar yang dapat digunakan untuk retrieval pada system ini misalnya histogram, susunan warna, teksture, dan shape, tipe spesifik dari obyek, tipe event tertentu, nama individu, lokasi, emosi [10]. Fokus pembahasan pada bagian ini adalah teknik *image retrieval* yang menggunakan *color histogram*.

2.2.1 Pengukuran Jarak Antar Dua Histogram

Fitur warna merupakan fitur yang paling banyak digunakan pada

sistem CBIR. Banyak diantaranya mengunakan *image color histogram*. *Color histogram* antara dua gambar tadi kemudian dihitung jaraknya. Gambar yang memiliki jarak paling kecil, merupakan solusinya.

Sebagai penjelasan, dimisalkan ada dua gambar dengan histogram 4 warna yang sudah terkuantisasi sebagai berikut:

$$H^A = \{20\%, 30\%, 10\%, 40\%\}$$

$$H^B = \{10\%, 10\%, 50\%, 30\%\}$$

Literatur [7] menyebutkan cara termudah untuk menghitungnya, yaitu dengan dengan menggunakan rumus:

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

Jika nilai 2 histogram tersebut dimasukkan ke dalam rumus diatas, maka hasilnya adalah sebagai berikut:

$$d(A,B) = |0.2 - 0.1| + |0.3 - 0.1| + |0.1 - 0.5| + |0.4 - 0.3| = 0.8$$

Cara lain untuk melakukan perhitungan jarak antar dua histogram adalah menggunakan rumus jarak Euclidan. Rumusnya:

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

Jika nilai dua histogram diatas dimasukkan ke dalam rumus, maka hasilnya adalah sebagai berikut:

$$d(A,B) = \sqrt{\sum_{j=1}^{n} (0.2+0.1)^2+(0.3+0.1)^2+(0.1-0.5)^2+(0.4-0.3)^2} = 0.47$$

2.2.2 Tipe *Colour Histogram*

Histogram warna terdiri dari dua tipe, *Global colour histograms (**GCHs**)* dan *Local colour histograms (**LCHs**).* 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.

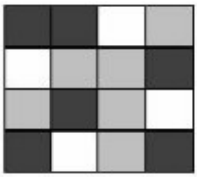


Image A

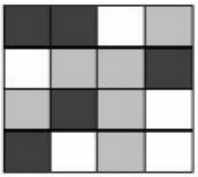


Image B

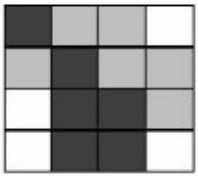


Image C

Gambar 2.3: 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 2.2: 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].

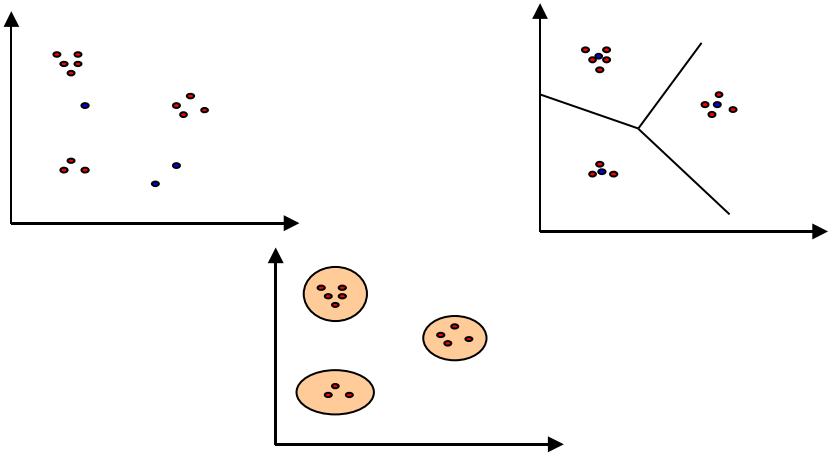
GCH merepresentasikan keseluruhan bagian gambar dengan satu histogram. Sedangkan LCH membagi gambar menjadi beberapa bagian dan

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].

2.3 K-Means

K-means merupakan *partitioning clustering* yang memisahkan data ke k daerah bagian yang terpisah. Cara ini sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan data outlier dengan sangat cepat.

Pada K-Means, setiap data harus termasuk ke cluster tertentu. Pada suatu tahapan proses sangat dimungkinkan bagi setiap data yang termasuk cluster tertentu, tapi pada tahapan berikutnya berpindah ke cluster yang lain [14].



Gambar 2.4: Ilustrasi Algoritma K-Means pada Data 2 Dimensi

2.3.1Algoritma K-Means

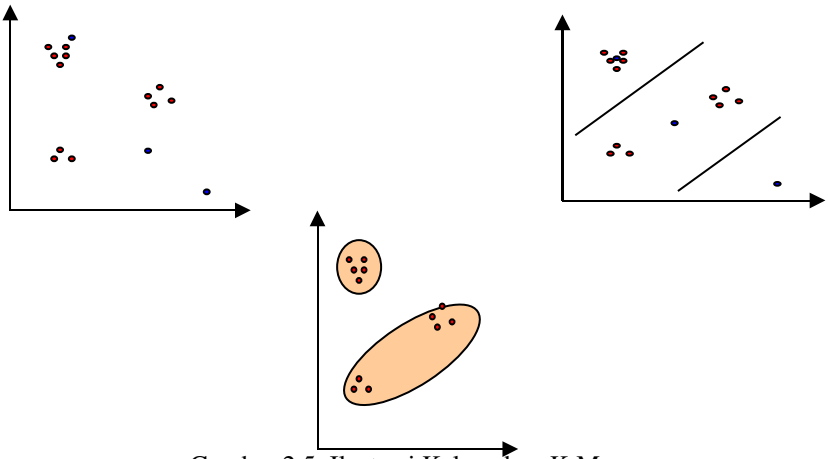
Berikut ini merupakan langkah-langkah yang harus dilalui untuk menggunakan K-Means:

- Tentukan k sebagai jumlah cluster yang ingin dibentuk
- Bangkitkan k centroids (titik pusat cluster) awal secara random

- Hitung jarak setiap data ke masing-masing centroids
- Setiap data memilih centroids yang terdekat
- Tentukan posisi centroids baru dengan cara menghitung nilai rata-rata dari data-data yang memilih pada centroid yang sama
- Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.

2.3.2 Karakteristik K-Means

- K-means sangat cepat dalam proses clustering
- K-means sangat sensitif pada pembangkitan centroids awal secara random
- Memungkinkan suatu cluster tidak mempunyai anggota
- Hasil clustering dengan K-means bersifat tidak unik (selalu berubah-ubah) - terkadang baik, terkadang jelek.
- K-means sangat sulit untuk mencapai global optimum



Gambar 2.5: Ilustrasi Kelemahan K-Means

2.4 Algoritma Genetika

Algoritma Genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. “Hanya individu-individu

yang kuat yang mampu bertahan".

Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembang- biakan. Dalam algoritma genetika ini, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: "Bagaimana mendapatkan keturunan yang lebih baik".

Algoritma genetika ini ditemukan oleh John Holland dan dikembangkan oleh muridnya David Goldberg [13].

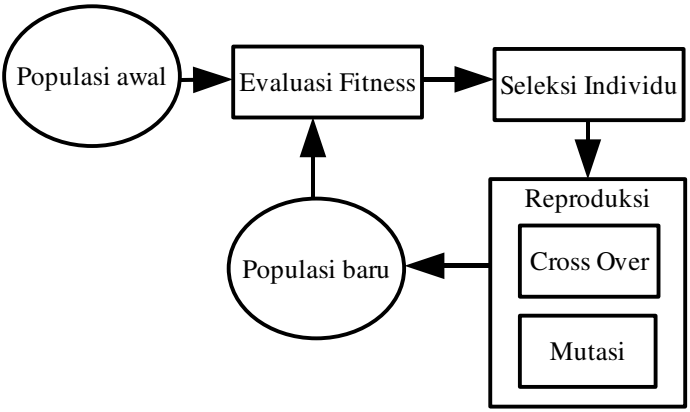
2.4.1 Beberapa Definisi Penting Algoritma Genetika

- Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter, atau kombinatorial.
- Allele, nilai dari gen.
- Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
- Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
- Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
- Generasi, menyatakan satu-satuan siklus proses evolusi.
- Nilai Fitness, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.[13]

2.4.2 Siklus Algoritma Genetika

- Membangkitkan Populasi Awal
  Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya [13].
- Seleksi
  Seleksi dilakukan untuk mendapatkan calon induk yang baik. "Induk yang baik akan menghasilkan keturunan yang baik". Semakin tinggi nilai fitness suatu individu , semakin besar

kemungkinannya untuk terpilih. Seleksi dapat dilakukan dengan menggunakan dua macam teknik, yaitu mesin roullete, dan turnamen [13].



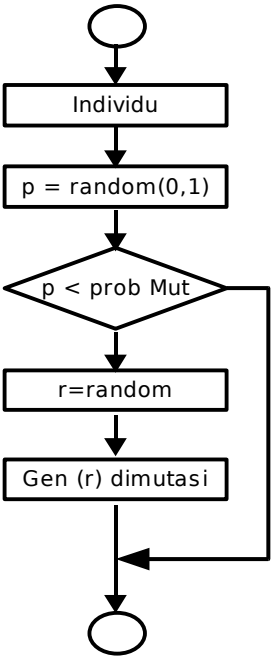
Gambar 2.6: Siklus Algoritma Genetika

- Cross Over

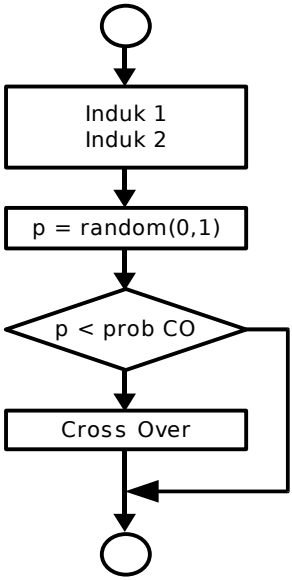
  Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan Induk 2 dua induk untuk menghasilkan keturunan yang baru. Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak. Macam-macam Cross-Over yang banyak digunakan antara lain: pertukaran gen secara langsung dan pertukaran gen secara aritmatika. Proses cross over dilakukan pada setiap individu dengan probabilitas *cross-over* yang ditentukan [13].

- Mutasi

  Mutasi Gen merupakan operator yang menukar nilai gen dengan nilai inversinya, misalnya gennya bernilai 0 menjadi 1. Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan. Mutasi dilakukan dengan memberikan nilai inversi atau menggeser nilai gen pada gen yang terpilih untuk dimutasikan [13].



Gambar 2.8: Mutasi



Gambar 2.7: Cross Over

2.4.2 Hal-hal yang Harus Diperhatikan dalam Algoritma Genetika

Algoritma Genetika adalah algoritma yang dikembangkan dari proses pencarian solusi menggunakan pencarian acak, ini terlihat pada proses pembangkitan populasi awal yang menyatakan sekumpulan solusi yang dipilih secara acak.

Berikutnya pencarian dilakukan berdasarkan proses-proses teori genetika yang memperhatikan pemikiran bagaimana memperoleh individu yang lebih baik, sehingga dalam proses evolusi dapat diharapkan diperoleh individu yang terbaik.

Algoritma Genetika sangat handal untuk melakukan proses pencarian dan optimasi. Salah satu hal yang menggembirakan adalah kecepatan komputasi algoritma genetika untuk proses optimasi bisa dikatakan sangat baik. Proses-proses optimasi yang membutuhkan kecepatan komputasi dengan kompleksitas yang tinggi membutuhkan

algoritma genetika [13].

2.6 Fast Genetic K-Means Algorithm

FGKA merupakan pengembangan dari Genetic K-means Algorithm (GKA) yang diusulkan oleh Khrisna dan Murty pada tahun 1999. Tentang performanya, dari percobaan dilaporkan bahwa dengan FGKA selalu didapatkan konvergensi pada optimal global, yang jika memakai algoritma K-means dimungkinkan didapatkan konvergensi pada optimal lokal. Dari eksperimen, juga dilaporkan bahwa pada saat K-means konvergen pada wilayah lokal, FGKA dan GKA *(Genetic K-Means Algorithm)* selalu konvergen pada wilayah global. Selain itu, jika dibandingkan dengan GKA, FGKA ternyata berjalan 20 kali lebih cepat [2, 3].

FGKA memproses satu populasi yang merupakan kumpulan Z solusi, dimana Z adalah sebuah parameter yang dimasukkan oleh pengguna. Masing-masing solusi dinamakan kromosom, yang dikodekan dengan string $a_1\ a_2\ ...\ a_N$ dengan panjang N, dimana a_n dinamakan gen yang berisi nilai {1,2,...,K} yang merepresentasikan nomor dari klaster yang merupakan memiliki gen tersebut. Sebagai contoh $a_1\ a_2\ a_3\ a_4\ a_5$ = "33212" mengkodekan 5 data dimana data X_1 dan X_2 milik klaster 3 dan X_3 dan X_5 milik klaster 2 dan X_4 milik klaster 1.

String yang tidak legal merepresentasikan sebuah solusi di mana terdapat klaster yang tidak mempunyai anggota. Sebagai contoh, jika jumlah K = 3, string $a_1\ a_2\ a_3\ a_4\ a_5$ = "23232" tidak legal karena klaster 1 tidak mempunyai anggota. Untuk sebuah kemungkinan solusi $S_z = a_1\ a_2\ ...\ a_N$, maka $e(S_z)$, dinamakan *legality ratio,* adalah angka yang menunjukkan jumlah klaster yang tidak punya anggota dalam S_z dibagi dengan K. Sehingga kita dapat mengatakan S_z adalah legal jika $e(S_z) = 1$, selainnya adalah tidak legal.

FGKA dimulai dari fase inisialisasi, dimana pada fase ini dibangkitkan populasi awal P_0. Populasi dalam generasi berikutnya P_{i+1} didapatkan menggunakan operator genetika dari populasi sebelumnya P_i. Evolusi akan berhenti jika kondisi akhir telah dicapai.

Operator genetika yang digunakan dalam algoritma FGKA ini adalah : seleksi, mutasi dan operator K-means. Di dalam merepresentasikan FGKA, digunakan notasi:

Z : jumlah kromosom (kemungkinan solusi dalam masing-masing populasi, dimana jangkauan z antara [1..Z];

N : jumlah data(obyek) yang akan diklaster, jangkauan n antara [1..N]

$X_1,\cdots,X_N$: data(obyek) yang diklaster, sejumlah N;

D : jumlah dimensi(atribut) dalam masing-masing data(obyek), jangkauan d antara [1..D];

K : jumlah klaster, sebuah parameter yang dimasukkan oleh pengguna, jangkauan k antara [1..K];

G_k: klaster yang ke-k;

SF_{kd}: jumlah dimensi ke-d untuk semua data dalam klaster G_k;

Z_k : jumlah data(obyek) dalam G_k;

a_n : gen ke-n dari sebuah kromosom(kemungkinan solusi) S;

Sz: solusi ke-z dalam populasi P, digunakan Sz untuk menggambarkan gen dari Sz dan TWCV dari Sz.

2.3.1 Inisialisasi Awal

Tahap inisialisasi adalah proses pembangkitan populasi awal P_0 yang berisi Z solusi, dimana Z adalah parameter yang ditentukan oleh pengguna. Masing-masing gen dalam kromosom menunjukkan nomor klaster yang dipilih secara random dengan distribusi uniform dalam jangkauan {1,...,K}. Dari inisialisasi awal ini dimungkinkan didapatkannya string yang tidak legal. Pada awalnya, untuk populasi awal, string yang tidak legal dihindari, artinya dalam populasi awal harus didapatkan string yang legal. Untuk itu, algoritma GKA membuat usaha siginifikan untuk mengeliminasi string ilegal tersebut. Namun, dalam FGKA string ilegal tadi diperbolehkan. Akan tetapi solusi ilegal tersebut akan diberi nilai TWCV $+\infty$, sehingga nantinya akan didapatkan nilai fitness rendah yang akan menyebabkan solusi ilegal tersebut akan mempunyai kemungkinan kecil untuk bertahan. Kompleksitas dari proses inisialisasi awal ini adalah $O(Z)$ [2,3].

2.3.2 Fungsi Fitness

Permasalahan klastering terdiri dari *N* gen yang merepresentasikan N data(obyek). Masing-masing data merupakan vektor *D* yang menggambarkan jumlah dimensi(atribut). Tujuan dari algoritma FGKA adalah membagi *N* data ke dalam *K* klaster, dimana klaster yang bagus adalah klaster yang memiliki *Total Within-Cluster Variation* (TWCV, yang juga dinamakan square-error dalam literatur) yang dijelaskan sebagai berikut.

Titik tengah $c_k = (c_{k1}, c_{k2}, \ldots, c_{kD})$ dari klaster G_k didefinisikan sebagai berikut:

$$c_{kd} = \frac{SF_{kd}}{Z_k} = \frac{\sum_{\vec{X_n} \in G_k} X_{nd}}{Z_k}, \ (d = 1, 2, \cdots, D) \tag{1}$$

Sedangkan *Within-Cluster Variation* dari klaster G_k didefinisikan sebagai berikut:

$$WCV(G_k) = \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} (X_{nd} - c_{kd})^2 \tag{2}$$

Dan *Total Within-Cluster Variation* (TWCV) didefinisikan sebagai berikut:

$$\begin{aligned}
TWCV &= \sum_{k=1}^{K} WCV(G_k) \\
&= \sum_{k=1}^{K} \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} (X_{nd} - c_{kd})^2 \\
&= \sum_{k=1}^{K} \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} (X_{nd}^2 + c_{kd}^2 - 2 * X_{nd} * c_{kd}) \\
&= \sum_{k=1}^{K} \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} X_{nd}^2 + \sum_{k=1}^{K} \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} c_{kd}^2 - \sum_{k=1}^{K} \sum_{\vec{X_n} \in G_k} \sum_{d=1}^{D} 2 * X_{nd} * c_{kd} \\
&= \sum_{n=1}^{N} \sum_{d=1}^{D} X_{nd}^2 + \sum_{k=1}^{K} Z_k \sum_{d=1}^{D} c_{kd}^2 - \sum_{k=1}^{K} \sum_{d=1}^{D} 2 * SF_{kd} * c_{kd} \\
&= \sum_{n=1}^{N} \sum_{d=1}^{D} X_{nd}^2 + \sum_{k=1}^{K} \sum_{d=1}^{D} \frac{SF_{kd}^2}{Z_k} - \sum_{k=1}^{K} \sum_{d=1}^{D} \frac{2 * SF_{kd}^2}{Z_k} \\
&= \sum_{n=1}^{N} \sum_{d=1}^{D} X_{nd}^2 - \sum_{k=1}^{K} \frac{1}{Z_k} \sum_{d=1}^{D} SF_{kd}^2
\end{aligned} \tag{3}$$

Definisi TWCV di atas diturunkan dari makalah GKA [1] akan tetapi dengan pengurangan kompleksitas waktu, dari $O(K*N*D)$ menjadi $O(N*D)$. Pengurangan ini tidak hanya mempengaruhi kecepatan dari algoritma FGKA, tapi diharapkan akan memberi kontribusi pada algoritma lain yang menggunakan kriteria *square-error*.

Fungsi fitness dalam literatur [9] dikatakan menggambarkan kemampuan suatu kromosom untuk bertahan dalam generasi berikutnya. Dalam algoritma FGKA ini tujuannya adalah meminimumkan *Total Within-Cluster Variation (TWCV)*. Sehingga solusi dengan nilai TWCV yang paling kecil merupakan kromosom yang mempunyai nilai fitness yang paling besar, dan kromosom yang tidak legal mempunyai nilai yang kecil. Dalam rumus 4 didefinisikan nilai fitness untuk S_z, $F(S_z)$ adalah:

$$F(S_z) = \begin{cases} 1.5 * TWCV_{max} - TWCV(S_z), \; if \; S_z \; is \; legal \\ e(S_z) * F_{min}, \; otherwise \end{cases} \tag{4}$$

Dengan rumus rumus di atas, maka kromosom dengan nilai TWCV yang paling kecil akan mempunyai nilai F yang paling besar, sementara untuk kromosom yang tidak legal secara otomatis akan mempunyai nilai F yang kecil, sehingga kromosom ini tidak akan bertahan dalam proses seleksi untuk diikutkan dalam proses selanjutnya [2,3].

2.3.3 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:

$$p_z = \frac{F(S_z)}{\sum_{z=1}^{Z} F(S_z)} \; (z = 1 \cdots Z) \tag{5}$$

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].

2.3.4 Operator Mutasi

Jika tiap kromosom dikodekan dengan a_1 a_2 ... 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:

$$p_k = \frac{1.5 * d_{max}(\vec{X_n}) - d(\vec{X_n}, \vec{c_k}) + 0.5}{\Sigma_{k=1}^{K}(1.5 * d_{max}(\vec{X_n}) - d(\vec{X_n}, \vec{c_k}) + 0.5)} \quad (6)$$

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. Mutasi ini didefinisikan sebagai berikut:

1. X_n dipilih secara random, hal ini bertujuan untuk mencapai optimal pada wilayah global.
2. Probabilitas a_n akan besar ke klaster k, jika X_n dekat dengan klaster k, hal ini diharapkan akan menjamin X_n untuk masuk ke klaster yang tepat dengan probabilitas yang tinggi.
3. X_n bisa jadi masuk ke dalam klaster kosong, diharapkan akan terdapat konversi dari string tidak legal menjadi legal.

Operator mutasi penting untuk mendapatkan solusi yang lebih baik. Dengan operator mutasi akan membawa algoritma keluar dari nilai optimum lokal, dan menuju optimum global [2,3].

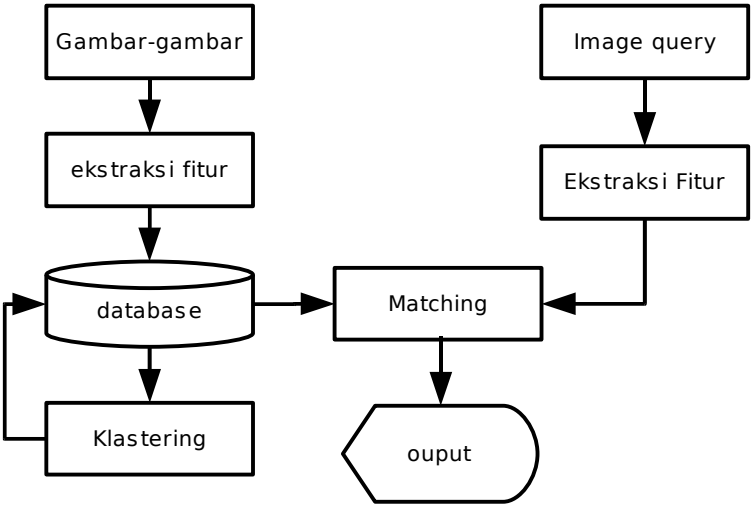
2.3.5 Operator K-means

Untuk mempercepat konvergensi, langkah klasik K-means algoritma digunakan dalam algoritma FGKA ini. Dengan solusi yang dikodekan dengan a1 a2 ... a_N, operator K-means 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].

$$d(\vec{X_n}, \vec{c_{a_n'}}) = \min_k \{d(\vec{X_n}, \vec{c_k})\} \quad (7)$$

BAB III
PERANCANGAN DAN PEMBUATAN SISTEM

3.1 Blok Diagram Sistem

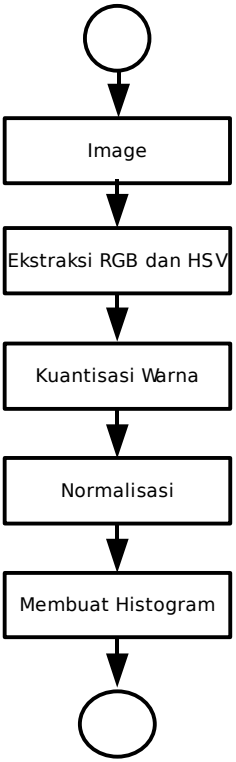


Gambar 3.1: Diagram Blok System

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*. Sedangkan *matching* (pencocokan), adalah proses pembandingan antara gambar *query* dengan gambar dalam database.

3.1.1 Ekstraksi Fitur Gambar

Seperti disebutkan diatas bahwa ekstraksi fitur adalah proses pengambilan histogram, baik dari gambar database maupun gambar query.

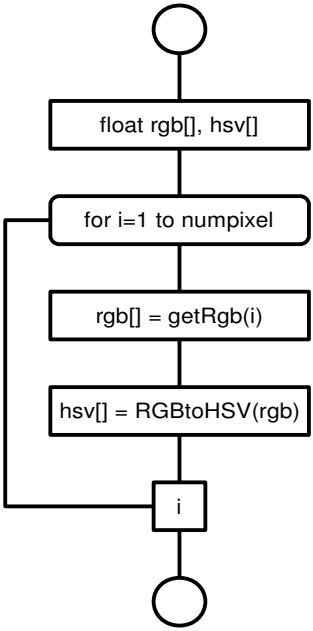


Gambar 3.2: Diagram Blok Ektraksi Fitur

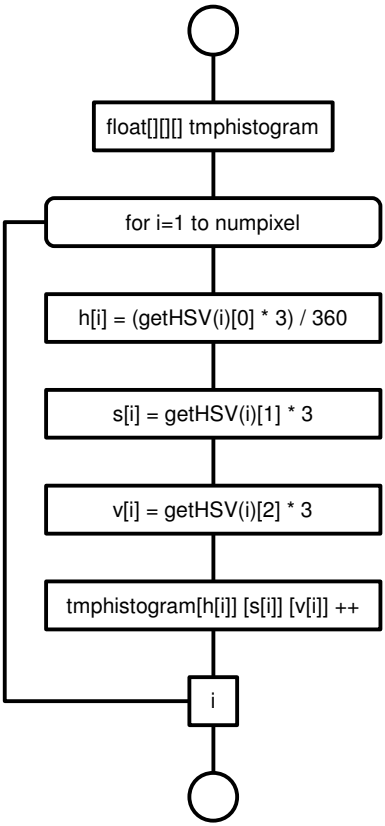
Tahap ini terdiri dari beberapa sub tahapan, yaitu:

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.

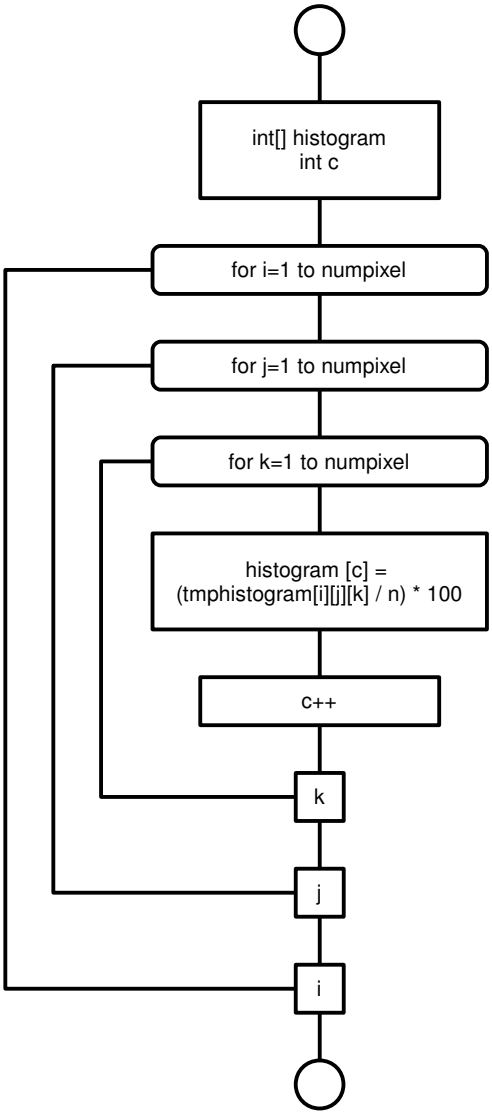
c. Normalisasi
d. Pembuatan HSV Histogram. Pada langkah ini juga dilakukan pembuatan Thumbnails yang berguna untuk menampilkan hasil pencarian dalam bentuk icon. Tipe histogram yang dipakai GCH.



Gambar 3.3 Ekstraksi HSV



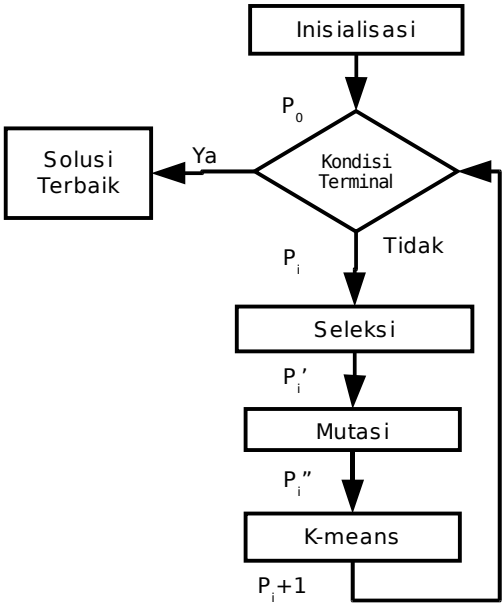
Gambar 3.4 Kuantisasi



Gambar 3.5 Normalisasi

3.1.2 Klastering

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



Gambar 3.6: 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.

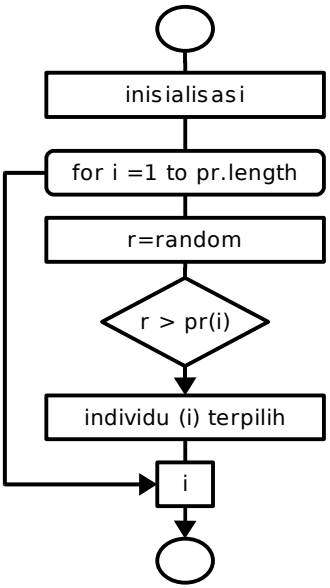
3.1.2.1 Operator Seleksi

Operator seleksi yang digunakan 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:

$$p_z = \frac{F(S_z)}{\sum_{z=1}^{Z} F(S_z)} \, (z = 1 \cdots Z) \tag{5}$$

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.



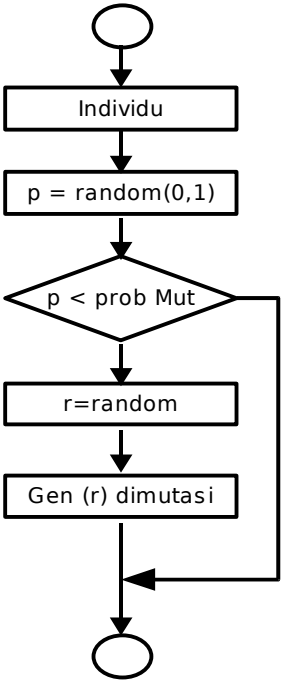
Gambar 3.7: Flow Chart Roulete Wheel

3.1.2.2 Operator Mutasi

Pada operator ini, tiap kromosom dikodekan dengan a_1 a_2 ... 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:

$$p_k = \frac{1.5 * d_{max}(\vec{X_n}) - d(\vec{X_n}, \vec{c_k}) + 0.5}{\Sigma_{k=1}^{K}(1.5 * d_{max}(\vec{X_n}) - d(\vec{X_n}, \vec{c_k}) + 0.5)}$$

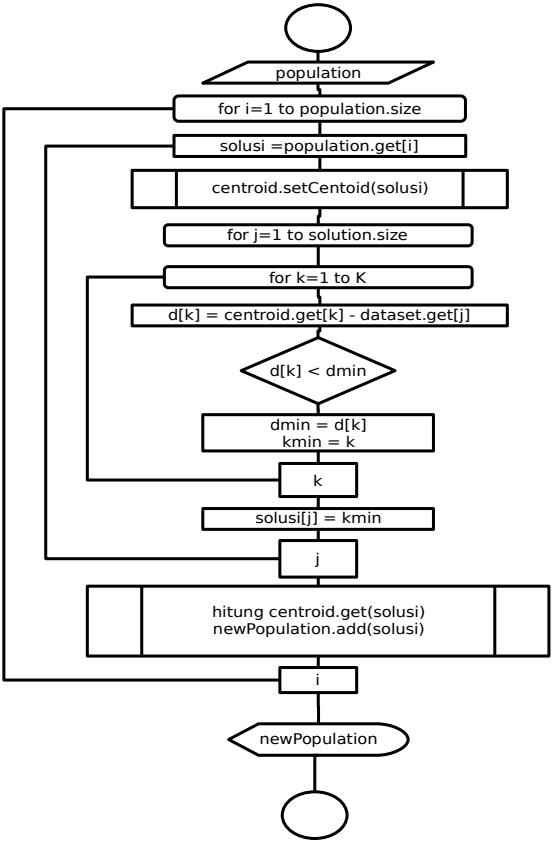
dimana $d(\vec{X_n}, \vec{c_k})$ adalah jarak Euclidean antara data $\vec{X_n}$ dan titik pusat $\vec{c_k}$ dari klaster ke-k.



Gambar 3.8: Flow Chart Operator Mutasi

3.1.2.3 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]



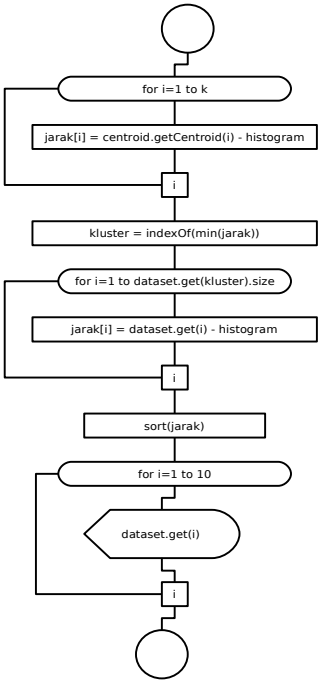
Gambar 3.9: Flow Chart K-Means Operator

3.1.3 *Matching* (Pencocokan)

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.

Setelah centroid yang memiliki jarak paling dekat tadi ditemukan, seluruh *HSV histogram* anggota centroid tersebut kemudian diukur jaraknya

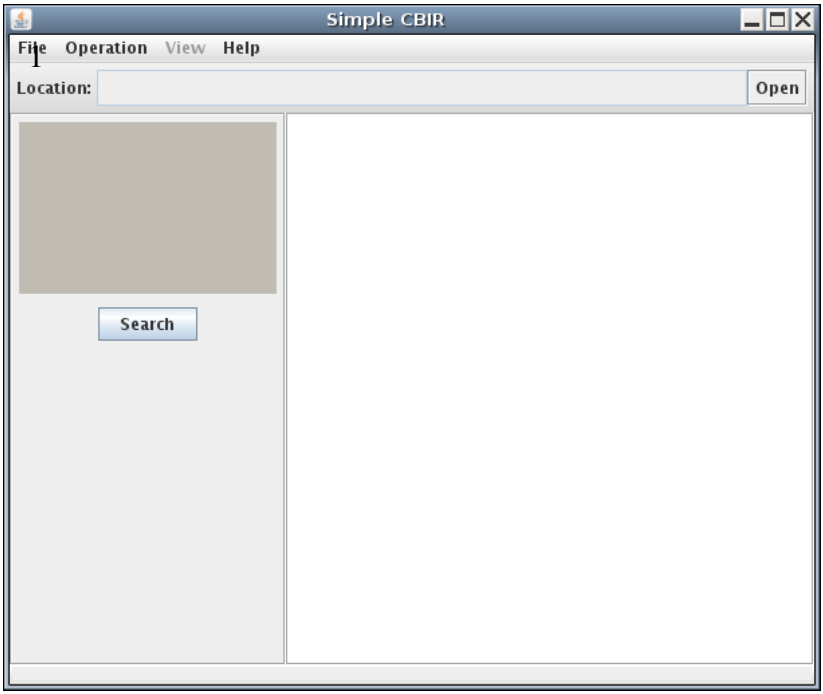
dengan *HSV histogram* gambar query. Menggunakan rumus euclidan distance. Hasilnya kemudian diurutkan. Hanya 10 gambar dengan selisih paling kecil saja yang ditempatkan pada posisi teratas.



Gambar 3.10 : Flow Chart Proses Matching

3.2 Pembuatan GUI

Pembuatan GUI dan implementasi algoritma terkait pada proyek akhir ini menggunakan bahasa java sedang penyimpanan database permanen menggunakan XML. Untuk memudahkan user dalam melakukan pencarian, aplikasi ini dibuat dalam bentuk GUI dengan beberapa satu frame utama, satu menu bar, satu tool bar, dua panel, dan satu status bar.

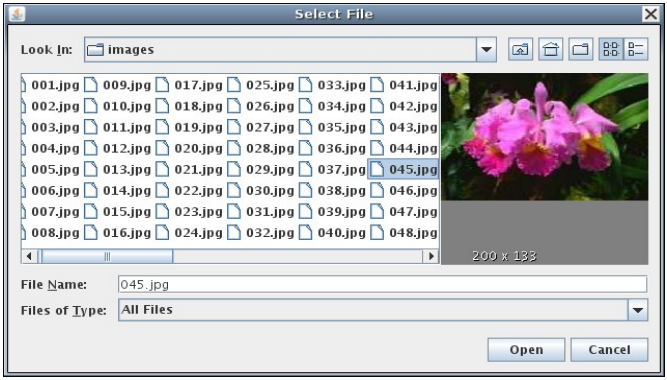


Gambar 3.11: Tampilan GUI simple CBIR

3.2.1 Tampilan Ketika Runtime

Di bawah ini adalah keterangan singkat beberapa fungsi menu tombol yang terdapat pada aplikasi GUI yang telah dibuat.

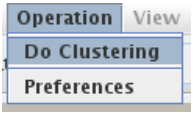
1. Menu File
   Terdiri dari 4 menu item
   a. Open File untuk membuka gambar yang akan dijadikan acuan dalam pencarian. Dilengkapi dengan *image preview*.



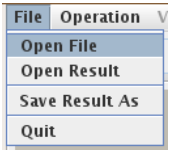
Gambar 3.13: Open File dengan image preview

b. Open Result untuk membuka hasil klastering
c. Save Result untuk menyimpan hasil klastering
d. Quit untuk keluar dari sistem

2. Menu Operation
   Terdiri dari dua menu item, yaitu:
   a. Do Operation: untuk melakukan klastering.
   b. Preferences: untuk pengaturan klastering.

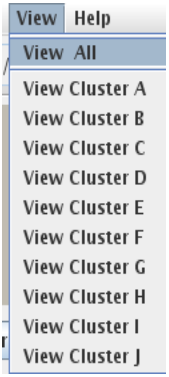


Gambar 3.14: Menu Operation

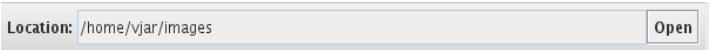


Gambar 3.12: Menu File

3. Menu View
   Keadaan awalnya disable kecuali jika database telah dibuat. Menu ini terdiri dari 2 bagian utama, yaitu
   a. View All: fungsinya untuk menampilkan seluruh isi gambar dalam suatu direktori.
   b. View Cluster : fungsinya untuk menampilkan gambar-gambar yang sudah tersegmentasi.



Gambar 3.15: View Menu



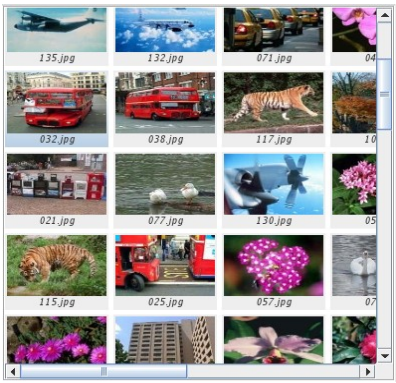
Gambar 3.16: Location Tool Bar

4. Toolbar
   Berisi satu textfield dan satu button untuk membuka direktori yang hanya berisi gambar.

5. Command Panel (bagian kiri),
   Terdapat area kecil untuk menampilkan icon gambar yang akan dicari. Dibawahnya terdapat tombol "search" yang berguna untuk mengawali pencarian.

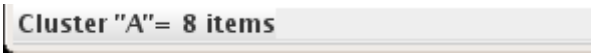
6. Result Panel (bagian kanan),
   Digunakan untuk menampilkan hasil pencarian atau hasil klastering.

Gambar 3.17: Command Panel



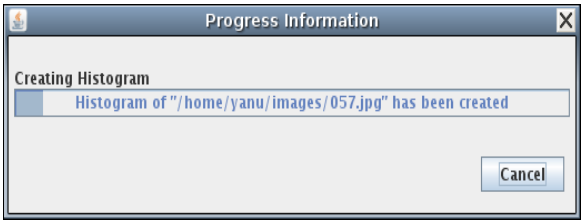
Gambar 3.18: Result Panel

7. Status bar (bagian paling bawah)
   Digunakan untuk memberikan keterangan hasil pencarian.

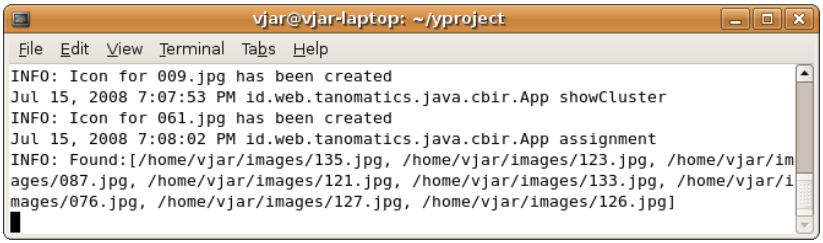


Gambar 3.19: Status Bar

8. Progess Information
   Digunakan sebagai indikator proses apa yang sedang dijalankan dan sejauh mana proses itu telah berjalan.



Gambar 3.20: Progess Information Dialog



Gambar 3.21: Logging

9. Sistem Logging
   Digunakan untuk menampilkan hasil logging operasi-operasi yang telah dilakukan. Logging ini ditampilkan dalam terminal.

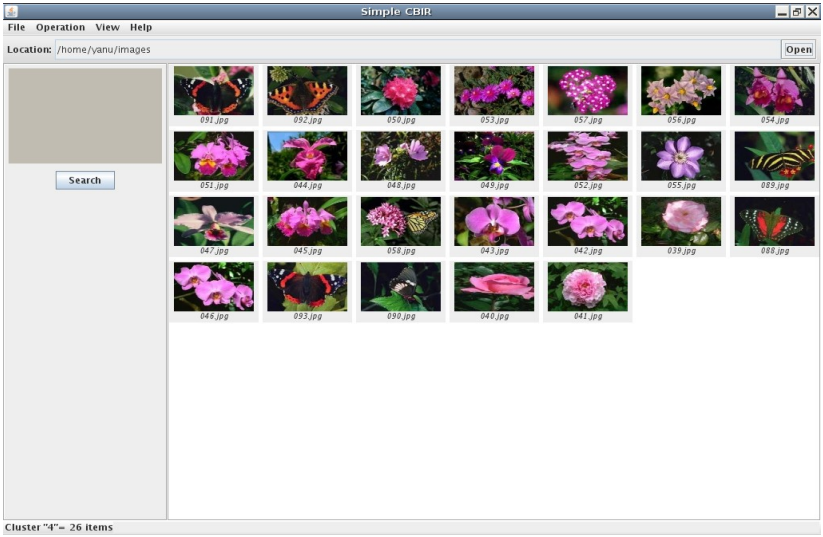
3.2.2 Menampilkan Hasil Klastering

Misalkan user sudah menentukan direktori tempat gambar-gambar berada dan ingin menampilkan hasil klastering yang telah dilakukan. Maka user memilih pada menu bar, View > View Cluster [no kluster]. Hasilnya akan tampak seperti gambar 3.23.

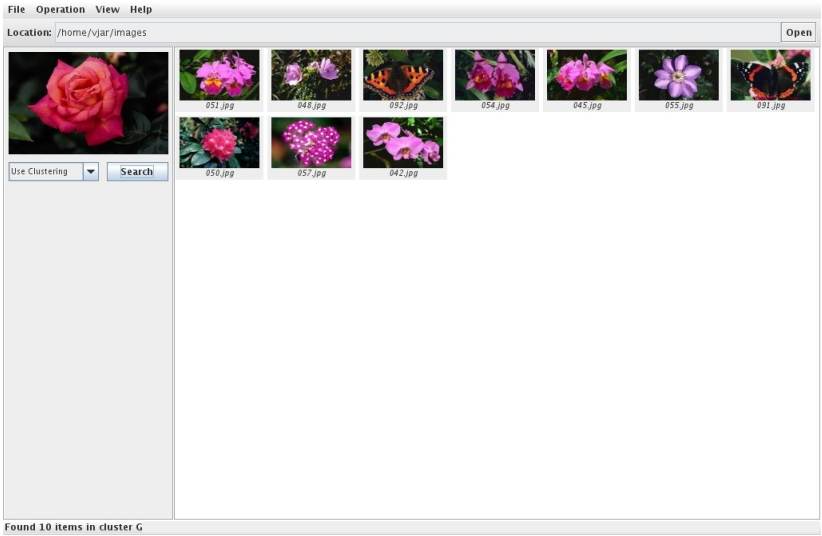
3.2.3 Menampilkan Hasil Pencarian

Misalkan user ingin gambar-gambar yang mirip dengan gambar 15, maka user tinggal membuka gambar yang ingin dicari dengan membuka menu File > Open File. Sebelumnya, direktori tempat gambar yang akan dicari gambarnya harus ditentukan dahulu (misalnya seperti gambar 3.19). Setelah image preview muncul pada *command pane*, tombol search kemudian ditekan. Hasilnya akan nampak seperti gambar 3.24.

Gambar 3.22: Contoh gambar query



Gambar 3.23: Hasil Klastering



Gambar 3.24: Tampilan Hasil Pencarian

Halaman ini sengaja dikosongkan

BAB IV
PENGUJIAN DAN ANALISA DATA

Untuk mengetahui hasil implementasi FGKA pada pencarian gambar, pada bab ini akan dipaparkan hal-hal yang berkaitan dengan lingkungan uji coba, parameter uji coba, skenario uji coba, hasil pengujian dan analisa hasil pengujian.

4.1 Lingkungan Uji Coba

Lingkungan uji merupakan bagian penting dari percobaan yang menentukan kecepatan komputasi. Pada percobaan ini, lingkungan uji cobanya adalah sebagai berikut:

- prosesor: 833 MHz; memori: 512 MB
- sistem operasi: ubuntu linux 7.10
- bahasa pemrograman:java
- sistem database: xml

4.2 Parameter Uji Coba

Parameter uji coba merupakan bagian penting dari percobaan yang menentukan hasil akhir pencarian. Pada percobaan ini, parameter uji coba yang digunakan adalah sebagai berikut:

- tipe gambar: jpg;
- *image content:* HSV histogram; tipe histogram: GCH
- jumlah gambar DB: 136 (dapat diunduh di http://yanuwid.googlepages.com).
- metode klastering: FGKA; *mutation probability:* 0.1; banyak klaster: 10; jumlah generasi:10; dan jumlah populasi: 10.

4.3 Skenario Uji Coba

Skenario uji coba merupakan urut-urutan proses yang dilakukan pada saat pengujian. Skenarion uji coba pada pengujian ini adalah:

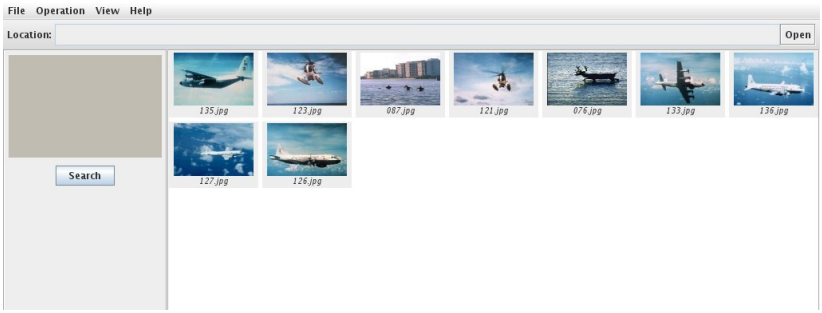
- Direktori yang berisi gambar-gambar (136 gambar) ditentukan.
- Sistem akan secara otomatis melakukan klastering
- Klastering dilakukan lagi sampai 10 kali.
- Setiap hasil klastering disimpan dalam file xml
- Pada tiap hasil percobaan (klastering), nilai-nilai kesesuaian klaster dan akurasi hasil pencarian dicatat.

4.4 Hasil Pengujian

Hasil pengujian pada percobaan ini terdiri dari dua. Pertama adalah hasil klastering dan kedua adalah hasil pencarian. Dua jenis pengujian itu masing-masing dilakukan sebanyak sepuluh kali.

4.4.1 Hasil Klustering

Percobaan 1



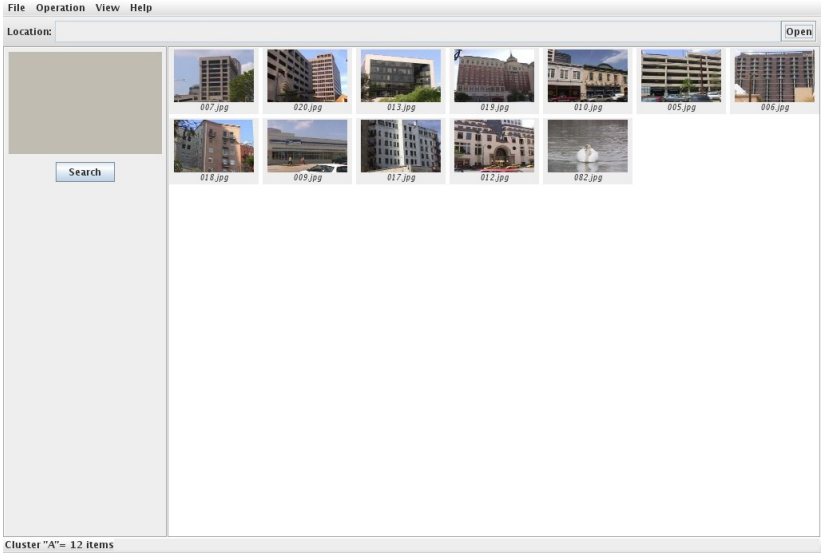
Gambar 4.1: Kluster A Hasil Percobaan 1

Pada percobaan ini, gambar klaster A didominasi dengan gambar langit (warna biru), dengan tingkat kemiripan 0.78. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.1

Cluster	Tingkat Kemiripan
A	0.78
B	0.4
C	0.64
D	1
E	0.92
F	0.57
G	0.89
H	0.67
I	1
J	1
Rata-rata	0.79
TWCV	54459.25

Tabel 4.1: Hasil Klastering Percobaan 1

Percobaan 2



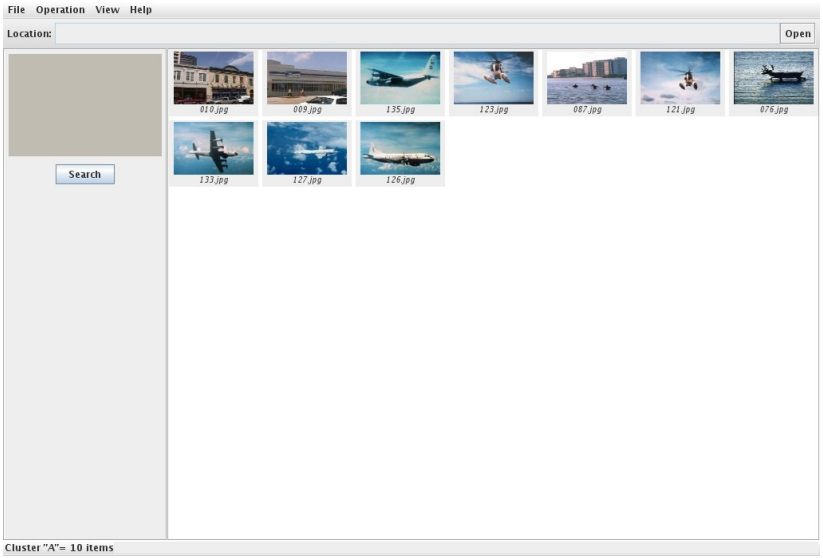
Gambar 4.2: Klaster A Hasil Percobaan 2

Pada percobaan ini, gambar klaster A didominasi dengan gambar gedung dengan tingkat kemiripan 0.92. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.2

Cluster	Tingkat Kemiripan
A	0.92
B	0.53
C	0.71
D	1
E	1
F	0.73
G	0.75
H	0.75
I	1
J	1
Rata-rata	0.84
TWCV	52287.03

Tabel 4.2: Hasil Klastering Percobaan 2

Percobaan 3



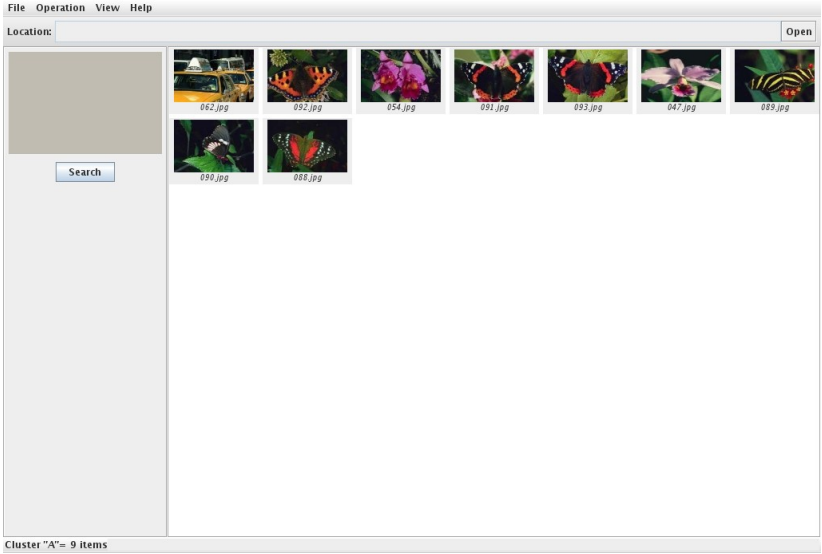
Gambar 4.3: Klaster A Hasil Percobaan 3

Pada percobaan ini, gambar klaster A didominasi dengan gambar pesawat dengan tingkat kemiripan 0.6. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.3

Cluster	Tingkat Kemiripan
A	0.6
B	0.52
C	0.67
D	0.5
E	0.75
F	0.88
G	1
H	0.77
I	0.56
J	1
Rata-rata	0.72
TWCV	520445.51

Tabel 4.3: Hasil Klastering Percobaan 3

Percobaan 4



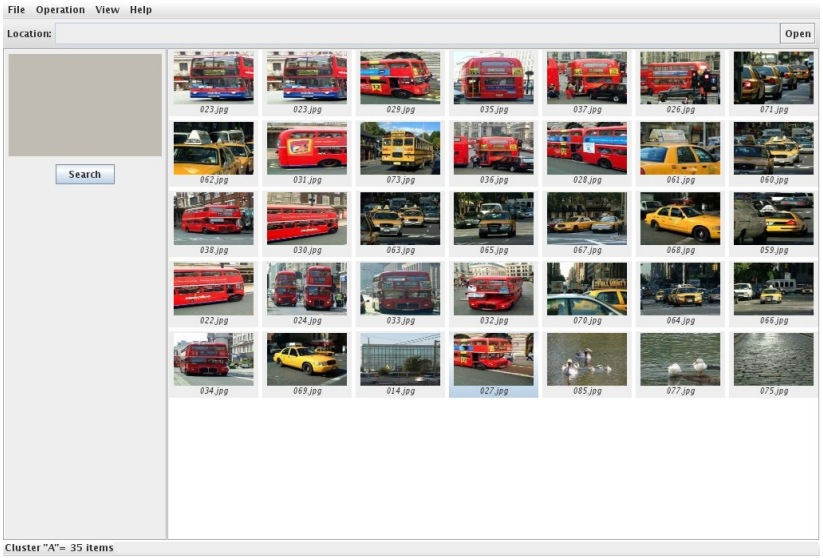
Gambar 4.4: Klaster A Hasil Percobaan 4

Pada percobaan ini, gambar klaster A didominasi dengan gambar kupu-kupu dengan tingkat kemiripan 067. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.4

Cluster	Tingkat Kemiripan
A	0.67
B	0.53
C	1
D	0.75
E	0.87
F	1
G	0.91
H	0.67
I	1
J	1
Rata-rata	0.84
TWCV	51848.08

Tabel 4.4: Hasil Klastering Percobaan 4

Percobaan 5



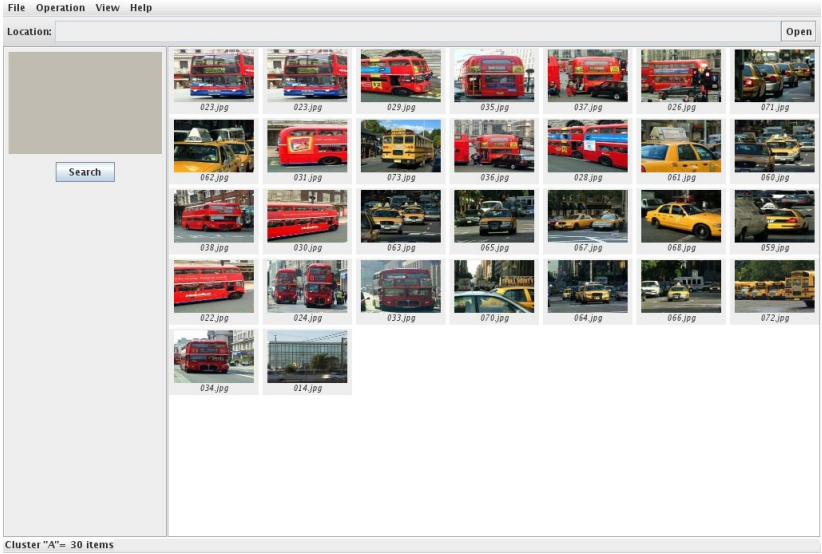
Gambar 4.5: Klaster A Hasil Percobaan 5

Pada percobaan ini, gambar klaster A didominasi dengan gambar mobil dengan tingkat kemiripan 0.49. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.5

Cluster	Tingkat Kemiripan
A	0.49
B	1
C	1
D	0.88
E	0.81
F	0.81
G	0.79
H	1
I	0.75
J	0.85
Rata-rata	0.85
TWCV	53523.29

Tabel 4.5: Hasil Klastering Percobaan 5

Percobaan 6



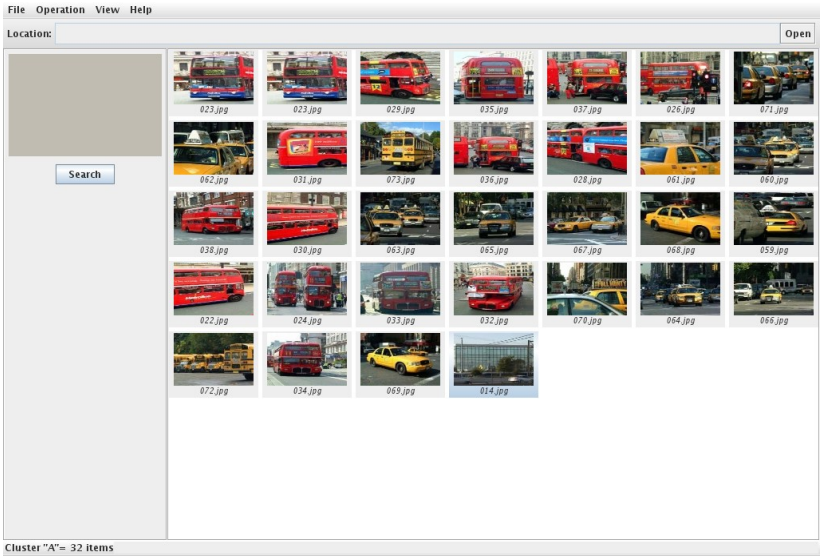
Gambar 4.6: Klaster A Hasil Percobaan 6

Pada percobaan ini, gambar klaster A didominasi dengan gambar mobil dengan tingkat kemiripan 0.5. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.6

Cluster	Tingkat Kemiripan
A	0.5
B	0.87
C	0.86
D	0.33
E	0.75
F	1
G	0.86
H	1
I	1
J	1
Rata-rata	0.82
TWCV	53864.59

Tabel 4.6: Hasil Klastering Percobaan 6

Percobaan 7



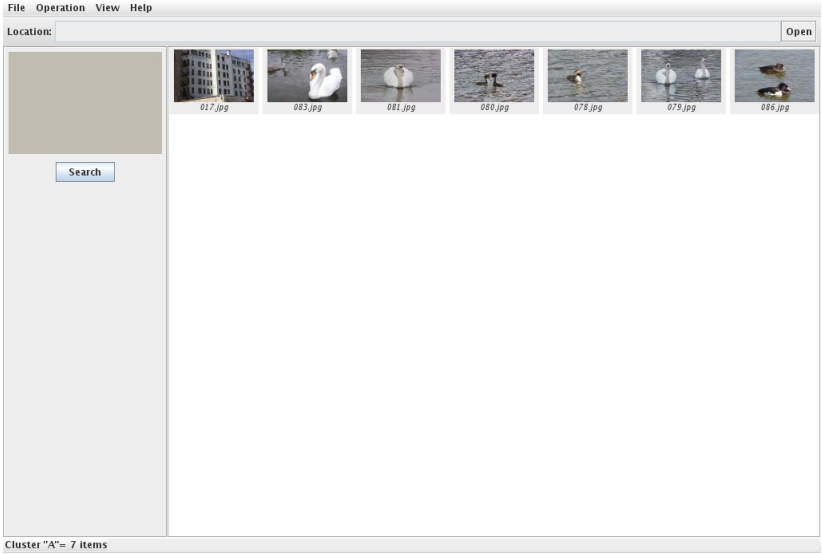
Gambar 4.7: Klaster A Hasil Percobaan 7

Pada percobaan ini, gambar klaster A didominasi dengan gambar mobil dengan tingkat kemiripan 0.5. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.7

Cluster	Tingkat Kemiripan
A	0.5
B	1
C	0.85
D	0.75
E	0.92
F	0.77
G	1
H	1
I	0.69
J	0.67
Rata-rata	0.81
TWCV	51924.18

Tabel 4.7: Hasil Klastering Percobaan 7

Percobaan 8



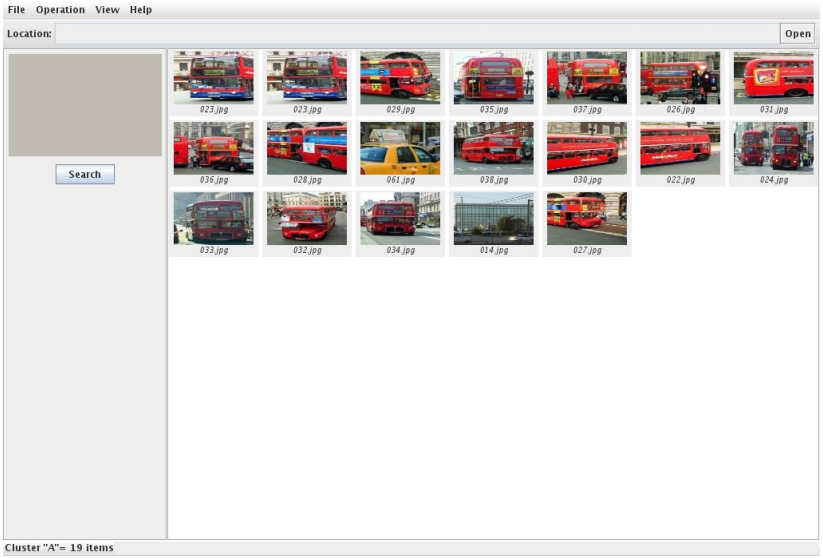
Gambar 4.8: Klaster A Hasil Percobaan 8

Pada percobaan ini, gambar klaster A didominasi dengan gambar unggas di danau dengan tingkat kemiripan 0.86. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.8

Cluster	Tingkat Kemiripan
A	0.86
B	0.78
C	0.9
D	1
E	1
F	1
G	0.76
H	0.75
I	0.67
J	0.87
Rata-rata	0.86
TWCV	53093.59

Tabel 4.8: Hasil Klastering Percobaan 8

Percobaan 9



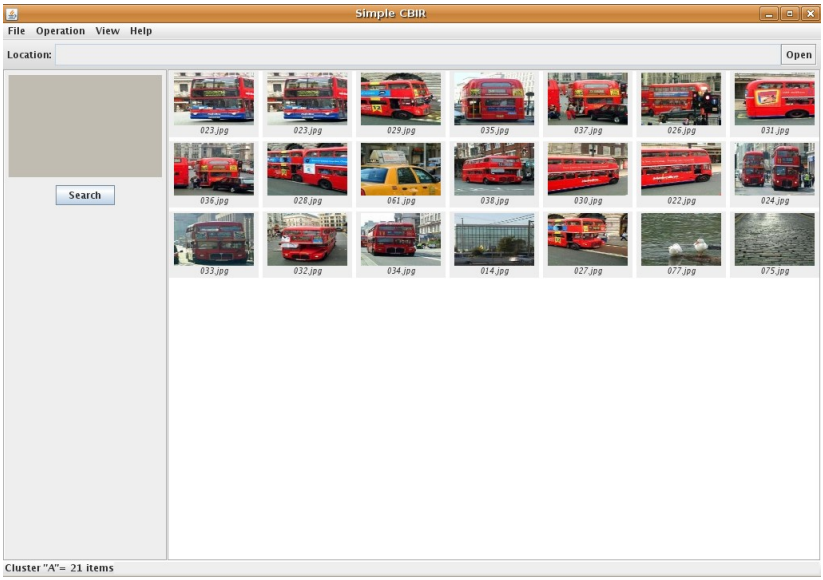
Gambar 4.9: Klaster A Hasil Percobaan 9

Pada percobaan ini, gambar klaster A didominasi dengan gambar mobil dengan tingkat kemiripan 0.89. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.9

Cluster	Tingkat Kemiripan
A	0.89
B	0.75
C	1
D	0.67
E	0.87
F	1
G	0.61
H	0.86
I	1
J	1
Rata-rata	0.87
TWCV	52184.82

Tabel 4.9: Hasil Klastering Percobaan 9

Percobaan 10



Gambar 4.10: Klaster A Hasil Percobaan 10

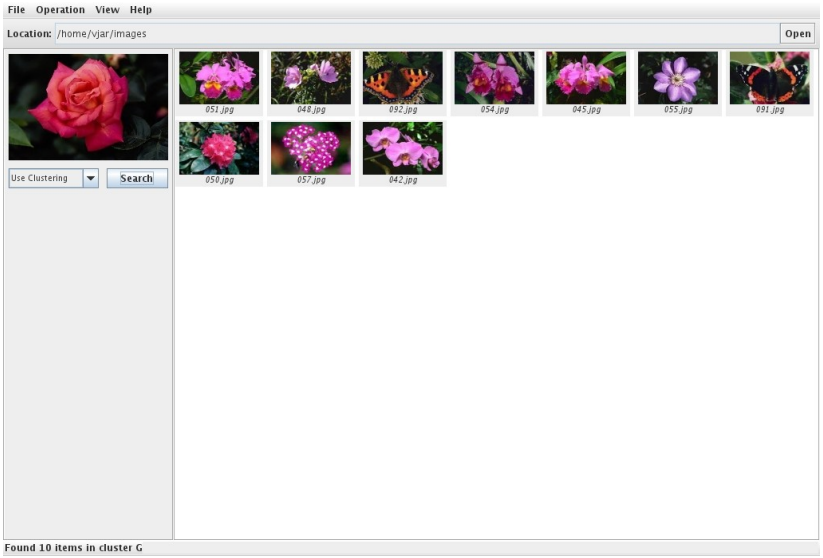
Pada percobaan ini, gambar klaster A didominasi dengan gambar mobil dengan tingkat kemiripan 0.81. Sedangkan hasil klaster yang lain dapat dilihat pada tabel 4.10

Cluster	Tingkat Kemiripan
A	0.81
B	0.75
C	1
D	0.93
E	0.81
F	1
G	1
H	1
I	0.76
J	1
Rata-rata	0.91
TWCV	52760.41

Tabel 4.10: Hasil Klastering Percobaan 10

4.4.2 Hasil Pencarian

Percobaan 1



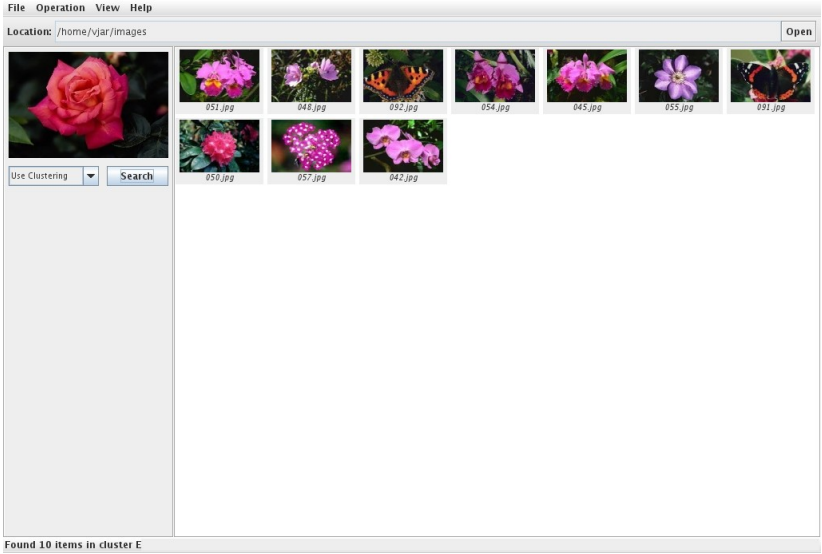
Gambar 4.11: Hasil Pencarian Percobaan 1

Gambar 4.11 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	389

Tabel 4.11: Hasil Pencarian Percobaan 1

Percobaan 2



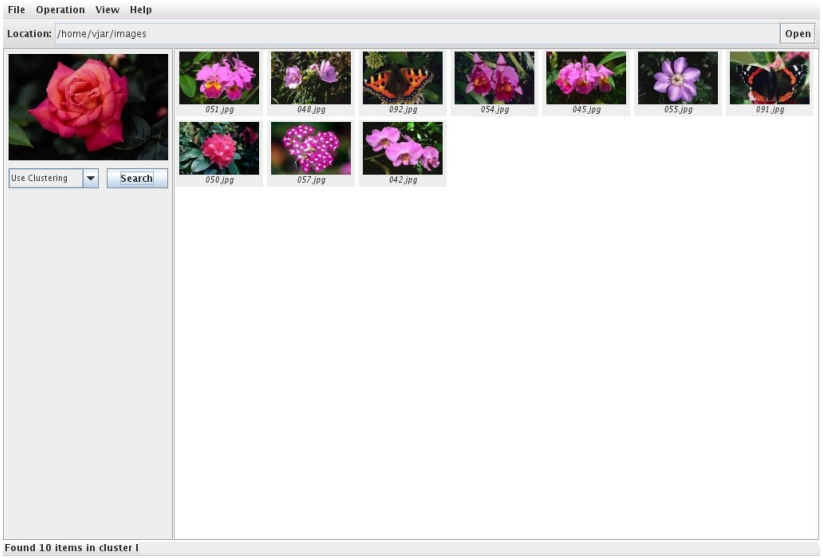
Gambar 4.12: Hasil Pencarian Percobaan 2

Gambar 4.12 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	316

Tabel 4.12: Hasil Pencarian Percobaan 2

Percobaan 3



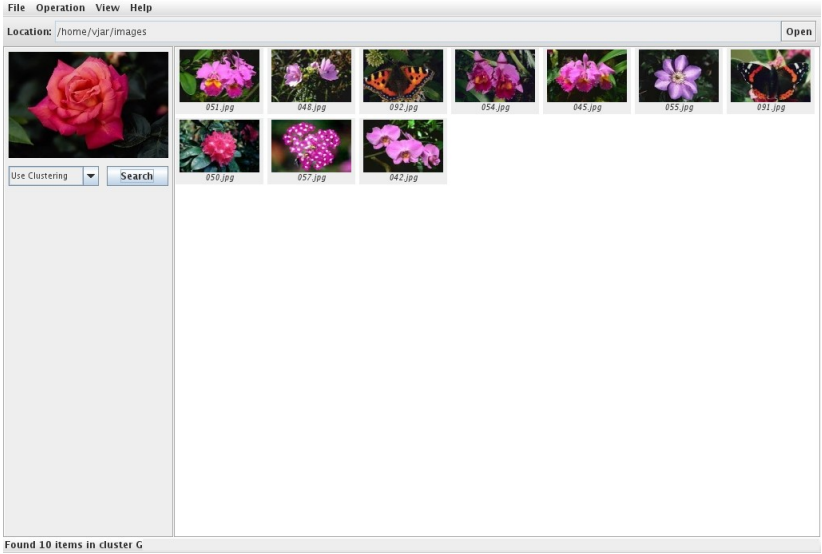
Gambar 4.13: Hasil Pencarian Percobaan 3

Gambar 4.13 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	284

Tabel 4.13: Hasil Pencarian Percobaan 3

Percobaan 4



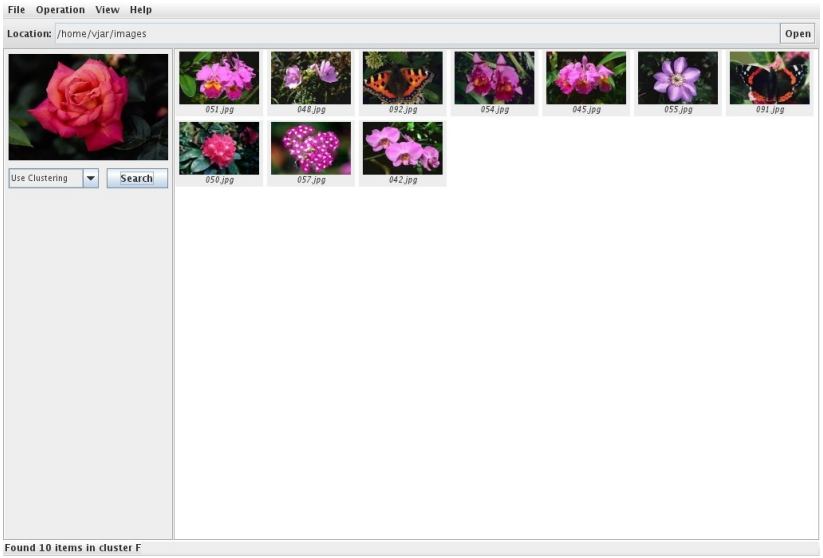
Gambar 4.14: Hasil Pencarian Percobaan 4

Gambar 4.14 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	317

Tabel 4.14: Hasil Pencarian Percobaan 4

Percobaan 5



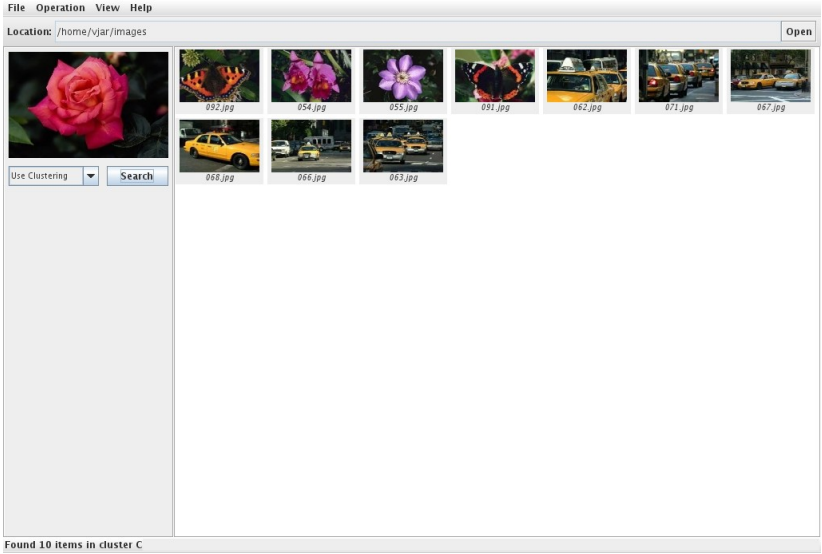
Gambar 4.15: Hasil Pencarian Percobaan 5

Gambar 4.15 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	265

Tabel 4.15: Hasil Pencarian Percobaan 5

Percobaan 6



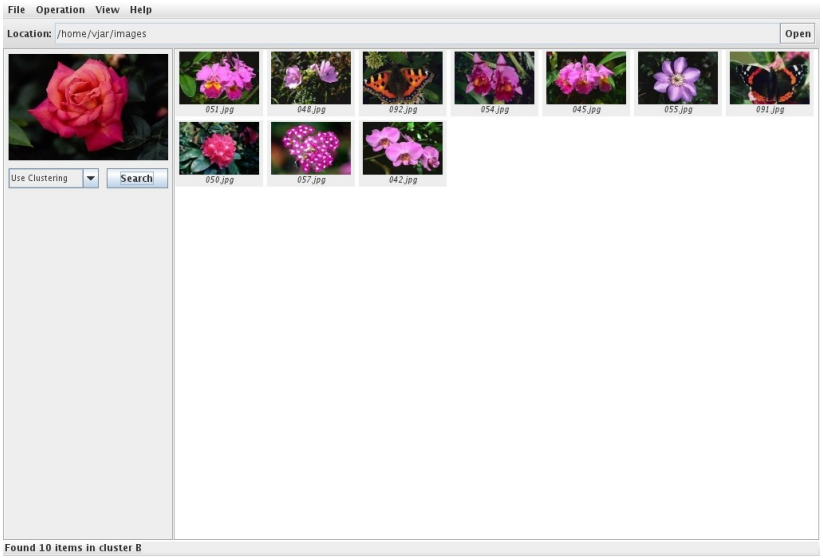
Gambar 4.16: Hasil Pencarian Percobaan 6

Gambar 4.16 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.2

No	Item	Hasil
1	Akurasi	0.2
2	Jarak gambar query dan centroid terdekat	26.35
3	Waktu yang dibutuhkan	226

Tabel 4.16: Hasil Pencarian Percobaan 6

Percobaan 7



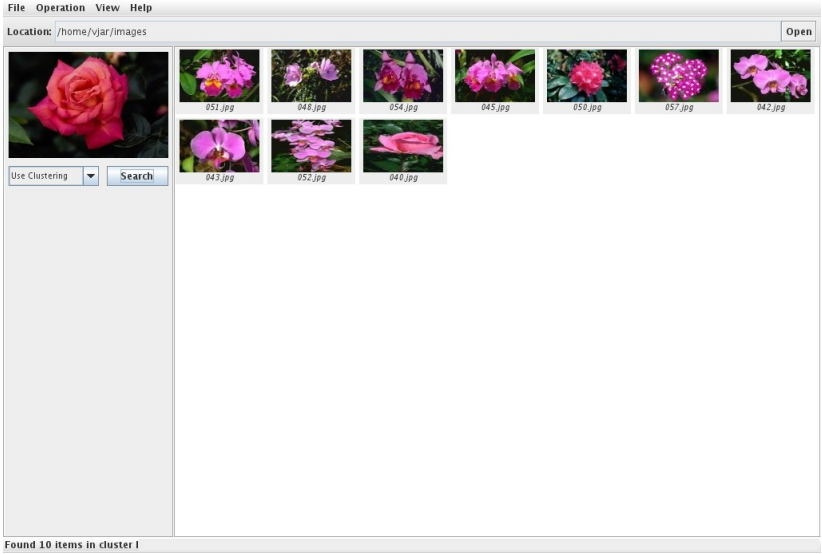
Gambar 4.17: Hasil Pencarian Percobaan 7

Gambar 4.17 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	192

Tabel 4.17: Hasil Pencarian Percobaan 7

Percobaan 8



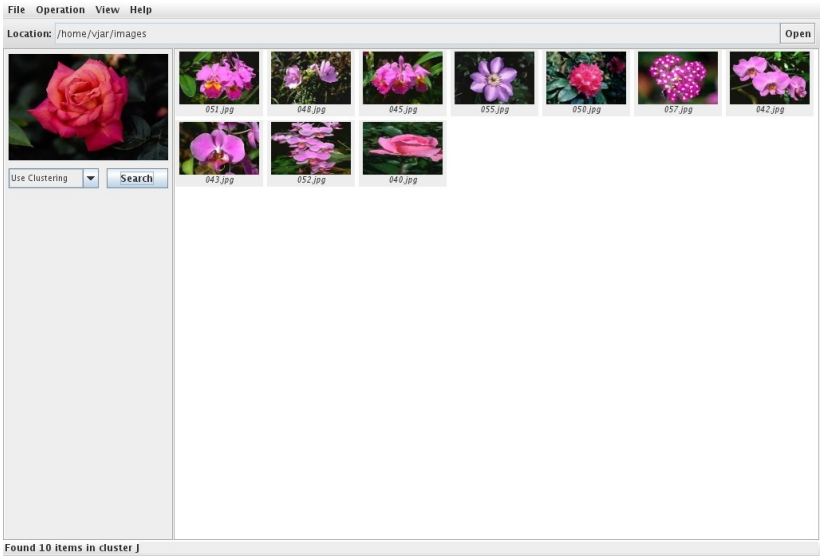
Gambar 4.18: Hasil Pencarian Percobaan 8

Gambar 4.18 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 1

No	Item	Hasil
1	Akurasi	1
2	Jarak gambar query dan centroid terdekat	26.55
3	Waktu yang dibutuhkan	224

Tabel 4.18: Hasil Pencarian Percobaan 8

Percobaan 9



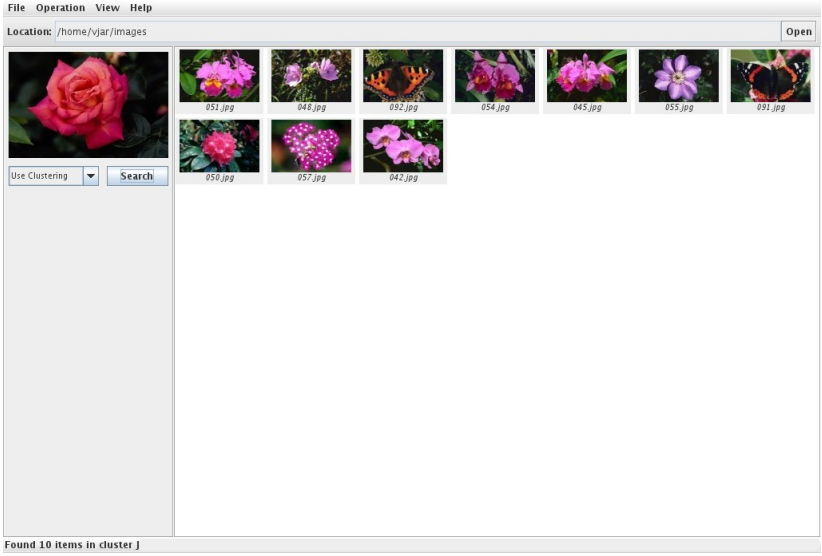
Gambar 4.19: Hasil Pencarian Percobaan 9

Gambar 4.19 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 1

No	Item	Hasil
1	Akurasi	1
2	Jarak gambar query dan centroid terdekat	26.55
3	Waktu yang dibutuhkan	183

Tabel 4.19: Hasil Pencarian Percobaan 9

Percobaan 10



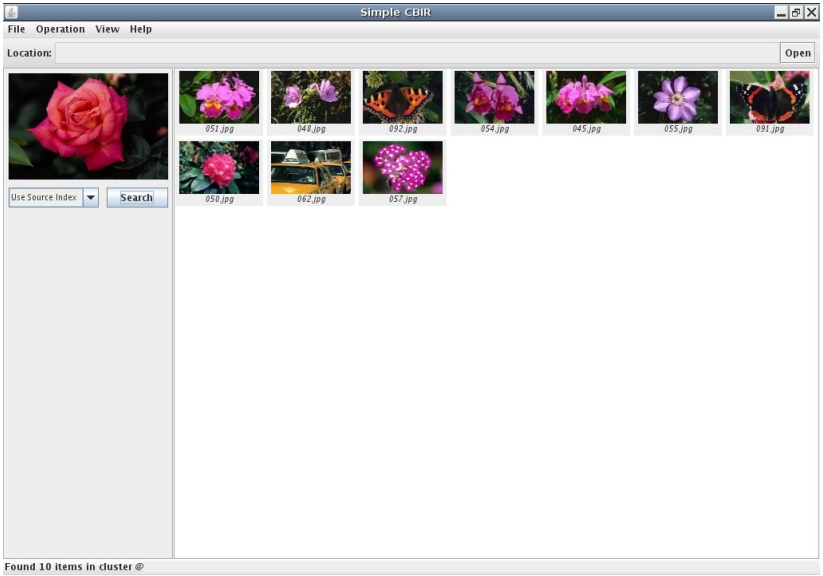
Gambar 4.20: Hasil Pencarian Percobaan 10

Gambar 4.20 menunjukkan hasil pencarian gambar menggunakan gambar query bunga. Tingkat akurasinya 0.8

No	Item	Hasil
1	Akurasi	0.8
2	Jarak gambar query dan centroid terdekat	25.23
3	Waktu yang dibutuhkan	252

Tabel 4.20: Hasil Pencarian Percobaan 10

Hasil Percobaan Tanpa Klastering



Gambar 4.21: Hasil Pencarian Tanpa Clustering

Pada gambar 4.21 ditunjukkan hasil pencarian gambar tanpa menggunakan metode klastering. Terlihat tingkat akurasinya selalu konstan.

No	Akurasi	Waktu yang dibutuhkan
1	0.7	395
2	0.7	397
3	0.7	372
4	0.7	393
5	0.7	387
6	0.7	356
7	0.7	243
8	0.7	385
9	0.7	212
10	0.7	341

Tabel 4.21: Hasil Pencarian Tanpa Klastering

4.5 Analisa Data

Analisa data pada bab ini terdiri dari dua bagian. Bagian pertama adalah analisa hasil klastering. Sedang yang kedua merupakan analisa data hasil pencarian.

4.5.1 Analisa Data Hasil Klastering

Berikut ini adalah tabel TWCV dan kemiripan dari sepuluh kali percobaan klastering:

No	TWCV (x)	Kemiripan (y)
1	54459.25	0.79
2	52287.03	0.84
3	52045.51	0.72
4	51848.09	0.84
5	53523.29	0.85
6	53864.59	0.82
7	51924.18	0.81
8	53093.59	0.86
9	52184.82	0.87
10	52760.41	0.91
Total	527990.75	8.3

Tabel 4.23: Hasil Klastering Keseluruhan Percobaan

Berdasarkan data-data diatas, dengan fungsi `correl()` pada OpenOffice.org Calc (versi 2.3), didapatkan nilai koefisien korelasi sebagai berikut:

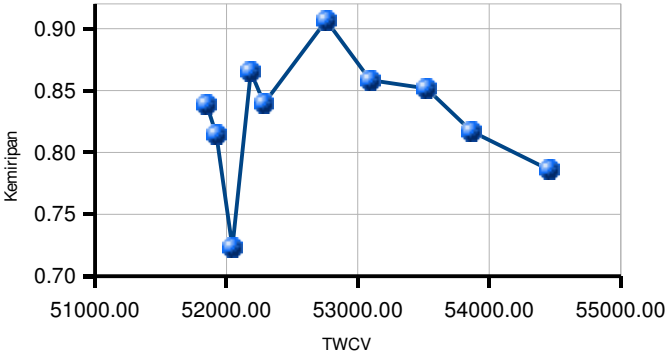
Statistik Regresi	
Koefisien Korelasi	-0.017
Koefisien Determinasi	0.03%

Tabel 4.24: Statistik Regresi Hasil Klastering

Dari hasil perhitungan diatas, dapat ditunjukkan 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. Dapat dilihat dari data-data pada tabel 4.2.1 pula bahwa nilai TWCV yang kecil, ternyata tidak selalu menghasilkan tingkat kemiripan yang tinggi.

Dari tabel juga dapat dilihat 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.



Gambar 4.22: Grafik TWCV-Rata-rata Kesesuaian

4.5.2 Analisa Data Hasil Pencarian

Berikut ini (tabel 4.25) adalah tabel hubungan jarak antara centroid dan gambar query dengan tingkat akurasi.

No	jarak gambar query dan centroid terdekat	Akurasi
1	25.23	0.80
2	25.23	0.80
3	25.23	0.80
4	25.23	0.80
5	25.23	0.80
6	26.35	0.20
7	25.23	0.80
8	26.55	1.00
9	26.55	1.00
10	25.23	0.80

Tabel 4.25: Akurasi dan Jarak Gambar Query-Centroid

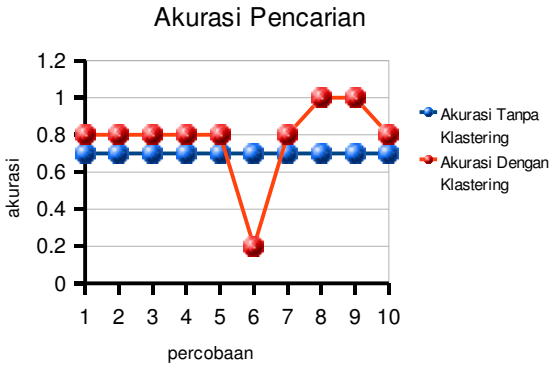
Berdasarkan data-data diatas, dengan fungsi `correl()` pada OpenOffice.org Calc (versi 2.3), didapatkan nilai koefisien korelasi sebagai berikut:

statistik regresi	
koefisien korelasi	-0.06
koefisien determinasi	0.33%

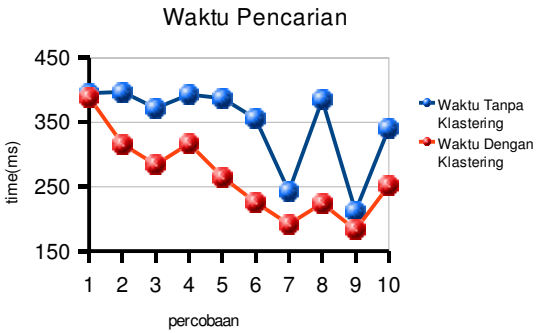
Tabel 4.26: Statistik Regresi Hasil Pencarian

Dari tabel 4.26, dapat dilihat 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 tabel 4.25, juga terlihat bahwa hasil yang dikembalikan 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 4.23: Grafik Perbandingan Akurasi



Gambar 4.24: Grafik Perbandingan Waktu

Sedangkan tabel berikut ini (tabel 4.27) adalah tabel yang menunjukkan perbedaan performa (akurasi dan kecepatan) antara pencarian tanpa klastering dengan tanpa klastering.

No	Akurasi		Waktu yang dibutuhkan	
	Tanpa Klastering	Dengan Klastering	Tanpa Klastering	Dengan Klastering
1	0.7	0.8	395	389
2	0.7	0.8	397	316
3	0.7	0.8	372	284
4	0.7	0.8	393	317
5	0.7	0.8	387	265
6	0.7	0.2	356	226
7	0.7	0.8	243	192
8	0.7	1.0	385	224
9	0.7	1.0	212	183
10	0.7	0.8	341	252
	rata-rata=0.7	rata-rata=0.78	rata-rata=348.1	rata-rata=264.8

Tabel 4.27: Perbandingan Hasil Pencarian

Dari hasil percobaan tanpa klastering, dapat dilihat bahwa tingkat akurasi pada sepuluh kali percobaan selalu tetap dengan nilai sebesar 0.7. Bila dibandingkan dengan pencarian dengan klastering (tabel 4.27), ternyata 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.

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.

Halaman ini sengaja dikosongkan

BAB V
PENUTUP

5.1 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.

5.2 Saran

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.

Halaman ini sengaja dikosongkan

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," tanpa tahun.

[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] Ahmad 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.

TENTANG PENULIS

Penulis yang punya nama lengkap Yanu Widodo ini lahir di Tulungagung -sebuah kabupaten kecil di Jawa Timur- pada awal tahun 1981. Dia adalah anak pertama dari empat bersaudara. Kegemarannya makan dan yang paling dia suka adalah buah jeruk. Dia tinggal di sebuah desa yang terletak di pinggiran pusat kota. Nama desa itu Ketanon. Obsesinya setelah mendapatkan gelar sarjana dari PENS-ITS adalah *travelling*, keliling dunia sambil membawa *backpack*. Blognya dapat diakses melalui http://yanuwid.blogspot.com.