

## SISTEM PENCARIAN CITRA DIGITAL MENGUNAKAN *CONTENT-BASED*

Herry Sofyan

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta

E-mail : [mas\\_herrysofyan@yahoo.com](mailto:mas_herrysofyan@yahoo.com)

### Abstrak

Metode berbasis teks (*text-based*) yang biasanya digunakan untuk mencari data berupa citra memerlukan waktu yang lama. Hal tersebut disebabkan karena harus dilakukan berbagai proses seperti proses pengindeksan, pengaturan nama file dan pembuatan katalog secara manual. Permasalahan tersebut dapat diselesaikan dengan memanfaatkan teknik pencarian citra (*image retrieval*) berdasarkan isi citra (*content-based*) yang dikenal dengan *content-based image retrieval (CBIR)*. Dengan cara mencari citra berdasarkan isi (*content*) visual citra seperti warna, bentuk, dan tekstur yang juga tercakup dalam standar MPEG-7 Color Descriptor. Sedangkan Lucene merupakan *Application Programming Interface (API)* untuk metode pencarian informasi (*information retrieval / IR library*) yang dikembangkan oleh Apache untuk menyediakan library pengindeksan dan pencarian berbasis teks (*text-based*). Citra akan dikonversi ke dalam vektor dan disimpan diindeks dalam bentuk teks. Indeks dimanfaatkan untuk proses pencarian dengan membandingkan vektor citra yang dicari dengan vektor-vektor citra yang terdapat di *repository image*. Aplikasi pencarian citra tersebut nantinya dapat dimanfaatkan dalam pengidentifikasian arsip digital image, pencocokan warna, pencarian duplikasi image, bidang desain grafis, dan lain-lain.

**Kata kunci** : *image, retrieval, repository*

### 1. PENDAHULUAN

Citra adalah bagian penting dari bidang multimedia, karena citra dapat menyampaikan banyak informasi yang dapat dievaluasi dan juga menjadi dasar representasi dan pengaturan indeks video. Pesatnya perkembangan data citra digital menimbulkan kesulitan bagi *user* dalam pengaturan, dan pencarian sehingga manfaat dari citra digital tersebut tidak akan berguna kecuali citra digital diorganisir terlebih dahulu sehingga memungkinkan efektifitas dalam melakukan *browsing* dan pencarian.

Kebanyakan pengaturan dan pencarian citra digital dilakukan secara konvensional dengan metode berbasis teks (*text-based*) dan penjelasan dalam bentuk teks (*text-annotated*). Sehingga hasil dari pencarian citra didasarkan pada pencarian dengan keyword nama file atau metadata citra. *Text-annotated* pada citra digital sangat sederhana dan mudah dimanipulasi. Terdapat dua persoalan dalam metode *text-annotated*. Pertama, membuat *keyword* untuk sejumlah besar citra merupakan pekerjaan yang memakan banyak waktu. Kedua, *keyword* yang diberikan *inherently subjective* dan tidak unik. Untuk mengatasi inefisiensi dan keterbatasan metode tersebut, maka dikembangkan metode pengambilan informasi multimedia (*multimedia information retrieval*) seperti citra dan video berdasarkan isi data visual berupa warna, tekstur dan bentuk objek citra. Proses pengindeksan dan pengambilan citra dapat dilakukan secara otomatis berdasarkan pada isi visual citra (*content-based*) dan dapat dikembangkan untuk aplikasi pencarian citra dengan jumlah data yang besar.

*Content-based image retrieval system (CBIRS)* adalah sistem dan teknik untuk melakukan pencarian citra (*image retrieval*) berdasarkan isi visualnya (*content-based*) antara lain seperti warna, tekstur, bentuk, tepian, dan karakteristik citra lainnya. Sedangkan MPEG-7 Color Descriptor digunakan untuk mengekstrak fitur citra secara otomatis berdasarkan deskripsi *content* yang terdapat dalam citra. Untuk proses pengindeksan dilakukan dengan memanfaatkan library Apache Lucene. Lucene menangani pengindeksan dan pencarian berbasis *full-text*. Untuk mengintegrasikan Lucene dengan aplikasi yang akan dibangun hanya perlu memahami perintah dan aturan *class* yang tersedia pada library Lucene yang dibangun dalam bahasa pemrograman Java (Hatcher, 2005).

Ide aplikasi pencarian citra yang dibangun terinspirasi oleh kelebihan Lucene yang mampu menerima segala jenis data untuk diindekskan terlebih dahulu. Untuk itu dalam aplikasi pencarian citra, harus dibuat dahulu suatu cara untuk mengubah citra dengan merepresentasikan citra ke dalam bilangan vektor. Kemudian membandingkan vektor citra dengan data yang telah diindekskan, lalu akan dihitung perbandingannya. Proses selanjutnya adalah mencari relevansi dari *query* citra dengan indeks yang ada. Hasil yang didapat berupa jarak vektor citra yang menunjukkan kesamaan dengan data yang sebelumnya diindeks terlebih dahulu. Semakin kecil jarak yang didapat dari proses perbandingan maka tingkat kesamaan semakin tinggi dengan *query* yang diberikan. Aplikasi pencarian citra ini dapat dimanfaatkan dalam pengidentifikasian arsip digital image, pencocokan warna, pencarian duplikasi image, bidang desain grafis dan lain-lain.

## 2. TINJAUAN PUSTAKA

### a. Aplikasi

Aplikasi adalah sebuah program komputer yang dibuat khusus untuk menjalankan fungsi-fungsi tertentu sesuai dengan kebutuhan pengguna yang digunakan untuk mempercepat suatu pekerjaan. Dengan menggunakan sistem komputerisasi, diharapkan pekerjaan dapat dilakukan dengan cepat. Aplikasi dapat dibagi menjadi dua bagian, yaitu aplikasi yang digunakan dan terhubung pada jaringan komputer (*client server*) dan aplikasi yang berdiri sendiri tidak terhubung dengan jaringan komputer (*stand alone*).

### b. Rational Unified Process Atau Unified Process (RUP/UP)

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. RUP merupakan iterasi proses perkembangan perangkat lunak yang dibuat oleh *Rational Software*, salah satu divisi IBM semenjak 2003. RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML).

### c. Citra

Secara harafiah citra (*image*) adalah gambar pada bidang dua dimensi. Citra yang dimaksud disini adalah citra diam. Citra diam adalah citra tunggal yang tidak bergerak. Citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. (Munir, 2004).

### d. Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer (Munir, 2004). Sebagai contoh jika gambar tampak agak gelap lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih tajam. Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah ditangkap oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan.

### e. MPEG-7 Color Descriptor

MPEG-7 atau "*Multimedia Content Description Interface*" adalah suatu standar ISO/IEC 15938 yang dikembangkan oleh MPEG (*Moving Picture Experts Group*), yang mendefinisikan tool standarisasi (*standard descriptor*) untuk mendeskripsikan *feature* atau isi (*content*) dari data multimedia (*multimedia content*). MPEG-7 berbeda dengan *standard* MPEG-1, MPEG-2, dan MPEG-4, MPEG-7 bukan *standard* untuk *coding* video. MPEG-7 menyediakan deskripsi standarisasi untuk *streaming* dan *storing multimedia content* yang mencakup data citra, audio, dan video. MPEG-7 adalah suatu *framework* yang *flexible* dan *extensible* untuk menjelaskan data audio dan visual.

### f. Lucene

Lucene adalah *library* untuk *Information Retrieval* (IR) yang berkemampuan tinggi dalam pengindeksan dan pencarian. Lucene bukan aplikasi siap pakai (*ready-to-use application*) seperti program *file-search*, *web crawler*, atau *search engine website*. Tetapi merupakan *Application Programming Interface* (API) sederhana yang menangani pengindeksan dan pencarian berbasis *full-text*. Untuk mengintegrasikan Lucene dengan aplikasi yang akan dibangun hanya perlu memahami perintah dan aturan *class-class* yang tersedia pada *library* Lucene yang dibangun dalam bahasa pemrograman Java. Dan merupakan bagian dari proyek Apache Jakarta dibawah naungan *Apache Software License*. Kelebihan Lucene adalah faktor kemudahan yang ditawarkan karena dibangun dalam bahasa Java sehingga integrasi dengan aplikasi yang akan dibangun lebih mudah diatur dan diawasi (Hatcher, 2005).

### g. Bahasa pemrograman Java

Bahasa Java awalnya bernama Oak, yakni bagian dari proyek Green yang dikembangkan oleh Sun Microsystems untuk memprogram perangkat-perangkat elektronik rumah tangga semacam televisi. Namun dalam perkembangannya, bahasa Oak ini menjadi bahasa yang bisa digunakan untuk pemrograman secara umum dan dikenal dengan bahasa Java saat ini (Wicaksono, 2002).

### h. JDK

JDK adalah software development kit yang digunakan dalam pemrograman bahasa java. Di dalam JDK itu ada sesuatu yg dinamakan JRE (Java Runtime Environment). JRE ini yang akan memproses byte code yang dihasilkan dari interpreter java. Tugas interpreter java adalah mengcompile source java menjadi source class.

Selanjutnya source class inilah yang akan dijalankan oleh JRE. JDK minimal yang harus kita miliki inilah yang dinamakan J2SE yaitu istilah selain J2ME dan J2EE.

### i. Rational Rose

Rational Rose adalah suatu perangkat lunak (*software*) yang digunakan untuk membuat notasi dan diagram standar UML. Rational Rose membantu keakuratan proses analisis dan desain karena dilengkapi dengan notasi sehingga dapat digunakan dengan metode *drag and drop*, software ini juga telah mengotomasi beberapa elemen sehingga saling terhubung dan dapat mengidentifikasi jika terjadi perubahan pada satu elemen karena terdapat perubahan pada elemen lain (Suhendar, 2002).

## 3. METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode RUP/UP yang memiliki beberapa fase dan terdiri dari beberapa *workflow*. Disini hanya akan dibahas mengenai dua fase yaitu fase *inception* dan fase *elaboration*. Pada fase *inception*, *workflows* yang dikerjakan yaitu : *requirements* dan pada fase *elaboration*, *workflows* yang dikerjakan yaitu *analysis* dan *design*.

### a. Fase Inception

Fokus utama fase *inception* adalah untuk memahami proyek yang akan dikerjakan dan mendapat informasi yang cukup untuk menegaskan apa yang harus dilakukan. Kegiatan yang dilakukan dan harus dicapai pada fase ini meliputi: menentukan ruang lingkup proyek, membuat arsitektur sistem secara *global* serta perencanaan dan usaha yang harus dilakukan untuk menghasilkan produk.

### b. Fase Elaboration

Fase ini merupakan fase untuk pembuatn arsitektur yang menjadi dasar dari sistem. *Workflow* yang banyak dijalankan pada fase ini adalah analisis dan Perancangan.

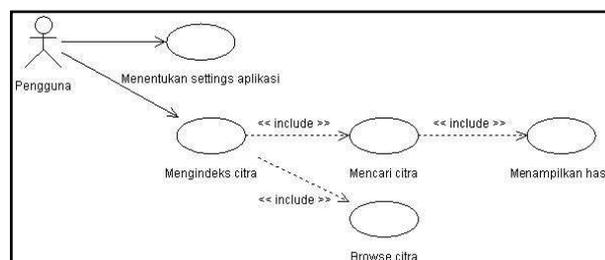
#### b.1. Analisis

Tujuan dari analisis adalah untuk mengubah kebutuhan sistem ke dalam bentuk yang menampilkan gambaran ideal dari sistem yang dibangun. Proses pencarian citra selama ini menggunakan sistem pencarian berbasis teks yaitu menerangkan citra dengan kata kuncinya. Dalam sistem pencarian citra berbasis teks, kata kunci (*keyword*) dari informasi semantik disertakan ke dalam citra. Sehingga pencarian dilakukan dengan pengetikan manual atau dengan ekstraksi *caption* citra. Cara tersebut sangat efisien untuk *database* citra sederhana dan berjumlah sedikit, karena keseluruhan *database* dapat dideskripsikan hanya dengan beberapa *keyword*. Namun, sistem tersebut tidak bekerja efektif pada pencarian citra yang berjumlah besar, dan citra-citra yang memiliki banyak *content* yang berbeda yang tidak cukup dapat dideskripsikan dengan beberapa *keyword* saja. *Keyword* juga sangat tergantung pada minat *observer* yang cenderung subjektif. *Caption* tidak selalu tepat dalam mendeskripsikan gambar. Pengindeksan (*indexing*) dan pencarian (*searching*) dalam suatu basisdata citra yang besar dengan *keyword* sangat memakan waktu dan tidak efisien.

#### b.2. Perancangan

Proses perancangan dimulai dengan pembuatan model sistem yang akan dibangun dengan menggunakan program bantu (tool) UML yang meliputi pembuatan diagram *use case*, diagram *activity* dan diagram *class*.

##### b.2.1. Diagram Use Case



Gambar 3.1. Diagram *use case* pencarian citra

Dari gambar 3.1, terdapat aktor yaitu pengguna yang menjalankan program. Berikut penjelasan untuk setiap *use case* dalam *use case specification*.

1. *Use case* menentukan *settings* aplikasi  
*Use case* menentukan *settings* aplikasi menunjukkan fungsi untuk melakukan *setting* aplikasi seperti metode pencarian, nama indeks dan jumlah hasil pencarian.
2. *Use case* mengindeks citra

Use case mengindeks citra menunjukkan fungsi untuk melakukan pengindeksan citra-citra *digital* yang tersimpan pada direktori yang tersimpan dalam repositori komputer pengguna. Jika tidak ditemukan direktori berisi citra-citra *digital*, maka program akan memberi pesan informasi mengenai hal tersebut.

3. Use case mencari citra

Use case mencari citra menunjukkan fungsi untuk melakukan pencarian citra dengan menggunakan citra sebagai *query* untuk menjalankan proses pencarian. Metode pencarian disediakan dalam tiga macam metode dan dapat diubah sesuai kebutuhan dalam menu *settings*.

4. Use case browse citra

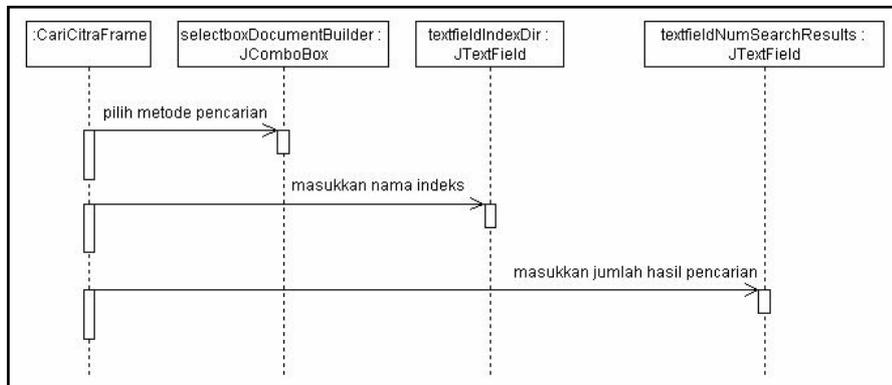
Use case browse citra menunjukkan fungsi untuk menampilkan citra-citra *digital* yang telah terindeks sebelumnya. Pengguna dapat melakukan *browsing* dengan spinner yang terdapat dalam halaman browse dan melihat citra-citra yang diinginkan.

5. Use case menampilkan hasil

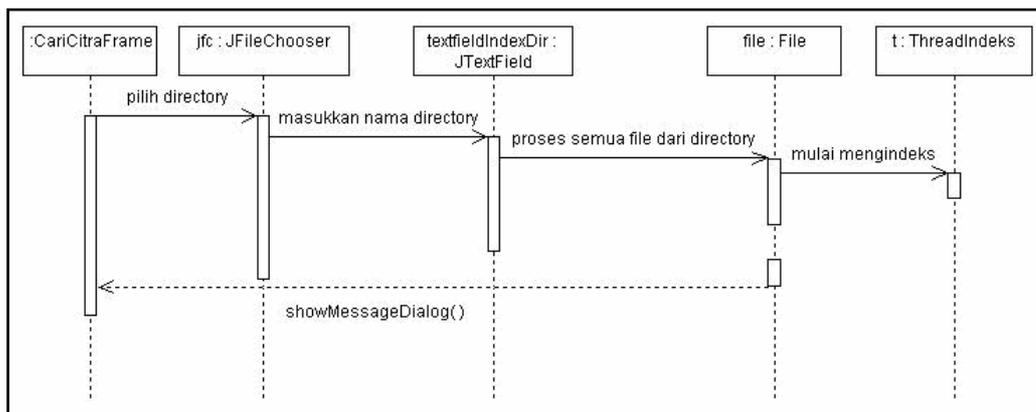
Use case menampilkan hasil menunjukkan fungsi untuk menampilkan hasil pencarian. Pengguna juga dapat melakukan pencarian dengan menggunakan citra hasil pencarian sebagai *query* citra pencarian.

**b.2. Diagram Sequence**

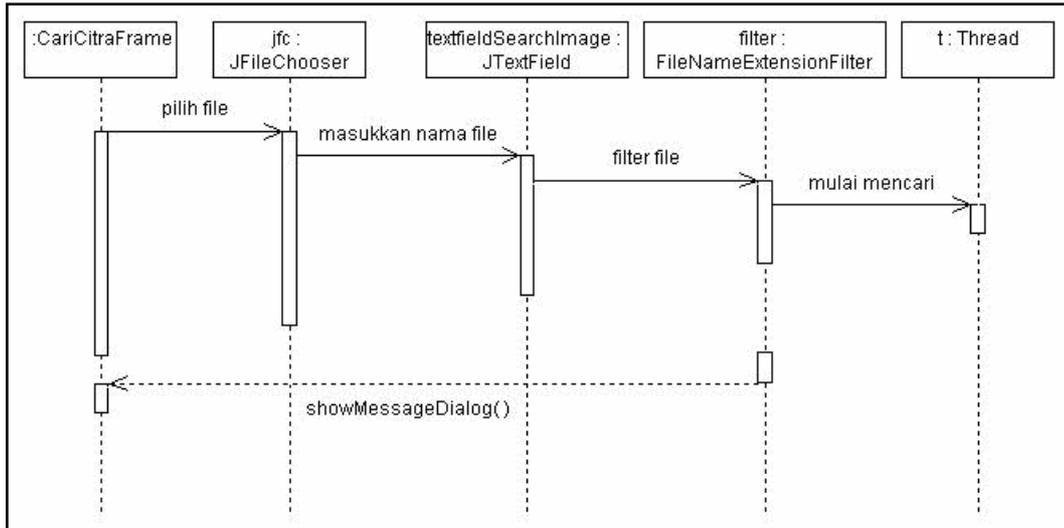
Diagram *sequence* diperoleh berdasarkan *use case* pada diagram *use case*. Diagram ini menunjukkan urutan *method* dan *object* yang terlibat dalam suatu *use case*. Berdasarkan diagram *use case* pada gambar 3.1. maka dapat dibuat sebanyak lima diagram *sequence* sebagai berikut:



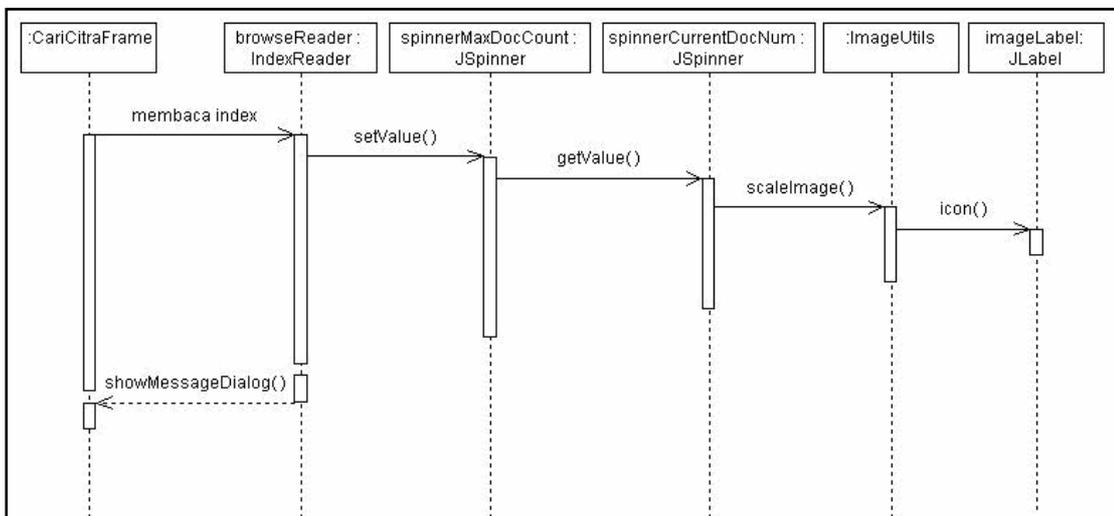
Gambar 3.2. Diagram *sequence* menentukan *settings* aplikasi.



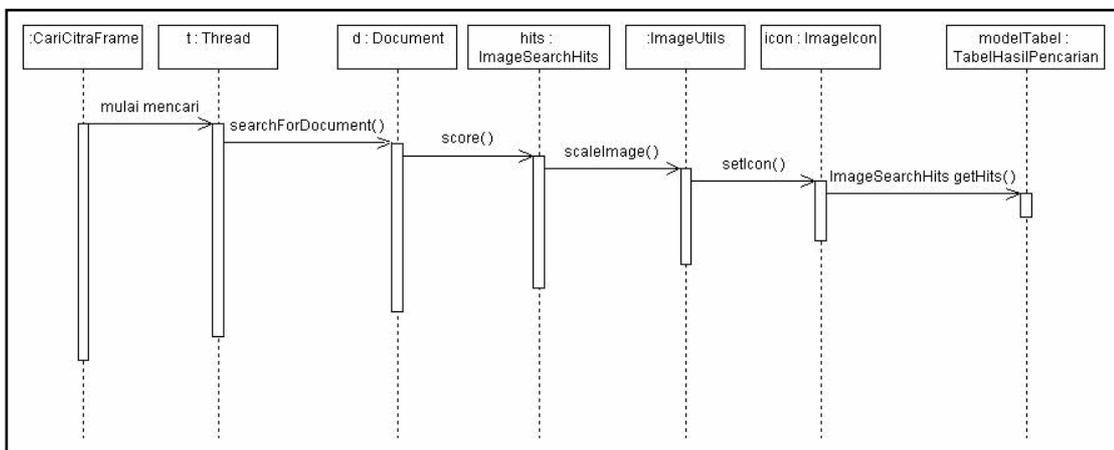
Gambar 3.3. Diagram *sequence* mengindeks citra.



Gambar 3.4. Diagram *sequence* mencari citra



Gambar 3.5. Diagram *sequence* browse citra



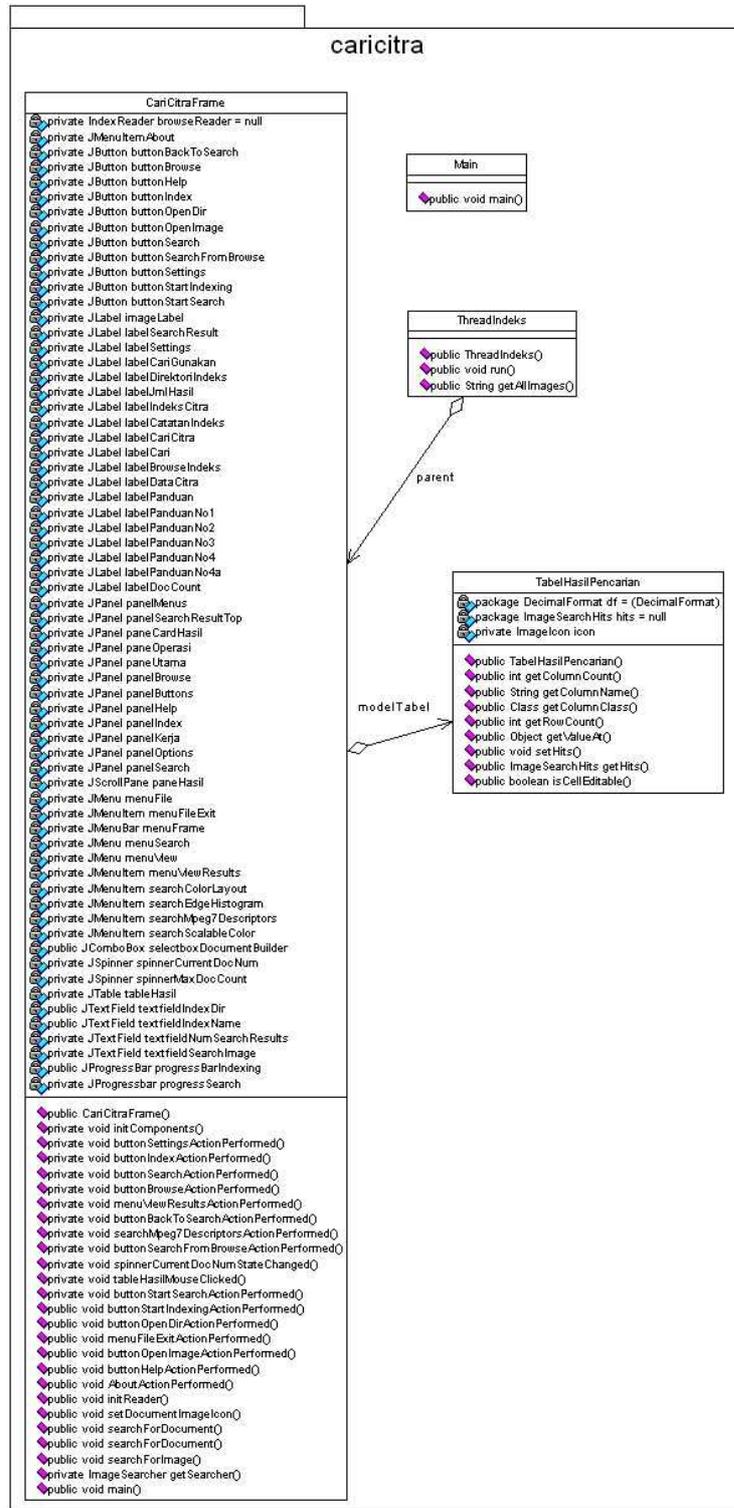
Gambar 3.6. Diagram *sequence* menampilkan hasil

**b.3. Diagram Kelas**

Pada tahap analisis dengan diagram kelas (*class diagram*) yang menjelaskan hubungan yang terjadi antar kelas dan program ini, dibagi ke dalam tiga diagram kelas, yaitu, diagram kelas mencari citra, diagram kelas dari *library package* net.semanticsimilarity dan *library package* org.apache.lucene sebagai kelas impor yang mendukung aplikasi utama.

1. Diagram kelas mencari citra terdiri dari empat kelas yaitu:

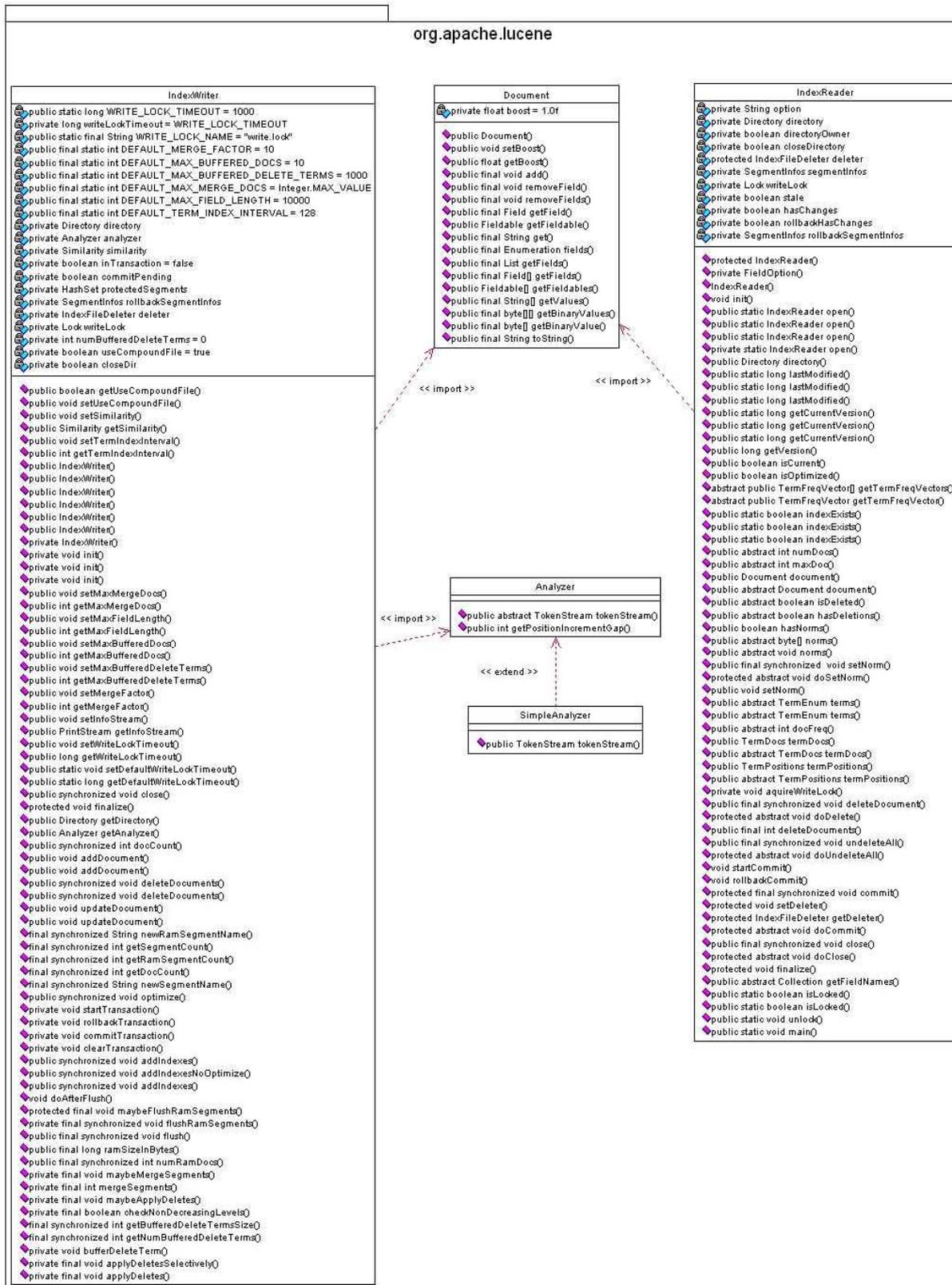
- Kelas CariCitraFrame
- Kelas ThreadIndeks
- Kelas TabelHasilPencarian
- Kelas Main



Gambar 3.7. Diagram kelas mencari citra



3. Diagram kelas *library package org.apache.lucene*

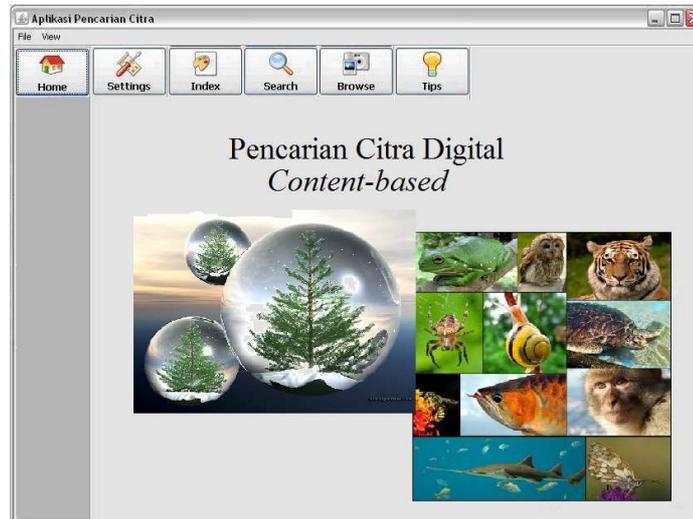


Gambar 3.9. Diagram kelas *library package org.apache.lucene*

#### 4. HASIL DAN PEMBAHASAN

##### a. Tampilan Antarmuka Halaman Home

Antarmuka pada gambar 4.1 berikut ini merupakan tampilan utama yang akan ditunjukkan kepada pengguna saat pertama kali aplikasi pencarian citra dibuka. Halaman Home ini juga berperan sebagai halaman default dari aplikasi. Operasi untuk membuka halaman Home dilakukan dengan menggunakan method `buttonHome.ActionPerformed` yang dikenakan pada button Home.

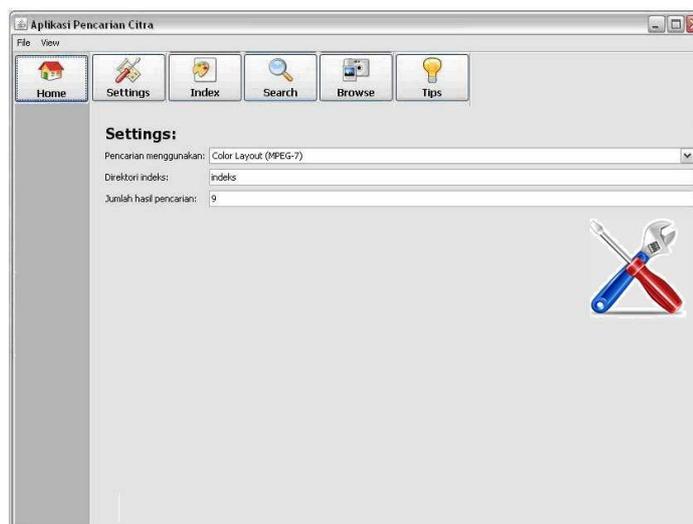


Gambar 4.1 Antarmuka halaman *home*

##### b. Tampilan Antarmuka Halaman Setting

Antarmuka pada gambar 4.2 ini digunakan untuk menentukan *settings* aplikasi pencarian citra, dimana fungsi untuk melakukan *setting* aplikasi seperti metode pencarian, nama indeks, dan jumlah hasil pencarian.

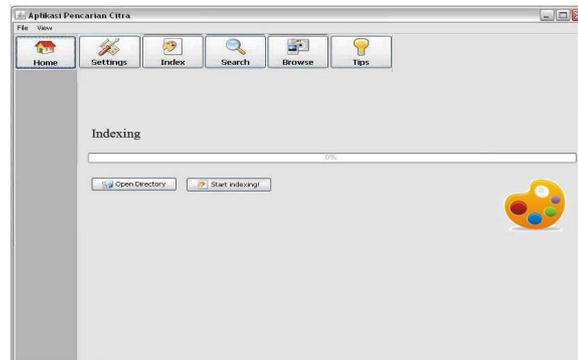
Proses untuk menentukan pilihan metode pencarian dimulai dengan membuka menu settings, yang dijalankan dengan metode `buttonSettings.ActionPerformed` dimana operasi ini akan membuka panel halaman settings. Kemudian pada tool selectbox yaitu `selectboxDocumentBuilder` untuk memilih metode pencarian, yang dijalankan dengan metode `searchMpeg7Descriptors.ActionPerformed` dimana pengguna dapat melakukan pull-down untuk memilih metode pencarian, jika `selectboxDocumentBuilder` bernilai 1 maka metode pencarian citra yang akan berjalan adalah `scalable color descriptor (command.equals("sc"))`, jika pilihan bernilai 2 adalah `edge histogram descriptor (command.equals("eh"))`, dan jika pilihan bernilai 3 adalah `color layout descriptor (command.equals("cl"))`.



Gambar 4.2 Antarmuka halaman *setting*

### c. Tampilan Antarmuka Halaman Indeks

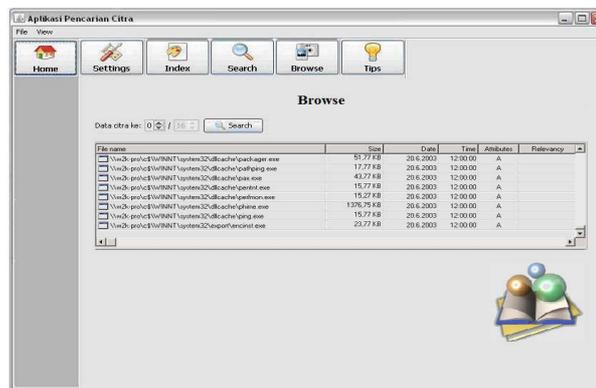
Antarmuka pada gambar 4.3 ini digunakan untuk mengindeks citra melakukan fungsi untuk melakukan pengindeksan citra-citra *digital* yang tersimpan pada direktori yang tersimpan dalam repositori komputer pengguna.



Gambar 4.3 Antarmuka halaman indeks

### d. Tampilan Antarmuka Halaman Browse

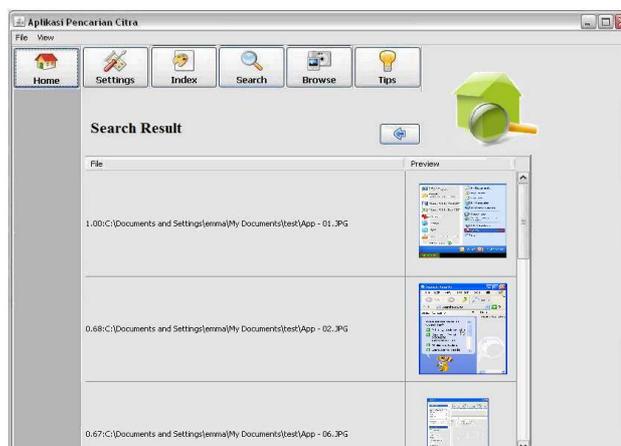
Antarmuka halaman *browse* citra ini menunjukkan fungsi untuk menampilkan citra-citra *digital* yang telah terindeks sebelumnya.



Gambar 4.4 Antarmuka halaman *browse*

### e. Tampilan Antarmuka Halaman Hasil

Antarmuka halaman menampilkan hasil ini berfungsi untuk menampilkan hasil pencarian. Pengguna juga dapat melakukan pencarian dengan menggunakan citra hasil pencarian sebagai *query* citra pencarian.



Gambar 4.5 Antarmuka halaman hasil

## 5. KESIMPULAN

Berdasarkan hasil analisis dan perancangan telah berhasil dibuat suatu aplikasi pencarian citra digital berbasis isi (*content-based*) dengan teknik query by example sehingga dapat mempercepat proses pencarian citra digital dalam repositori pengguna. Teknik ini lebih efisien dari segi waktu bila dibandingkan dengan proses pencarian berbasis teks (*text-based*).

## 6. DAFTAR PUSTAKA

- Berg, Daniel, 2000, *Search and Delivery of Multimedia Information Using MPEG-7 Technology*.  
<<ftp://ftp.csd.uu.se/pub/papers/masters-theses/0172-berg.pdf>>, (5 Des 2000, accessed 2 Sep 2007).
- Chang, Phang, 2007, *Modified Fast and Exact Algorithm for Fast Haar Transform*.  
<<http://www.waset.org/pwaset/v26/v26-96.pdf>>, (12 Desember 2007, accessed 26 Mei 2008).
- Deb, Sagarmay, 2004, *Multimedia Systems and Content-Based Image Retrieval*, Idea Group Publishing.
- Eakins, John, 1999, *Content Based Image Retrieval*.  
<<http://www.jisc.ac.uk/jtap/htm/jtap-039.html>>, (Desember 2002, accessed 2 Sep 2007).
- Hartati, Sri, 2007, *Pemrograman GUI Swing Java dengan Netbeans 5*, Andi Offset, Yogyakarta.
- Hatcher, Erik, 2005, *Lucene in Action*, Manning Publications.
- Kosch, Harald, 2004, *Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21*, Auerbach Publications.
- Manjunath<sup>1</sup>, 2001, *Color and Texture Descriptors*.  
<<http://vision.ece.ucsb.edu/publications/01IEEEManjunath.pdf>>, (6 Juli 2001, accessed 2 Sep 2007).
- Manjunath<sup>2</sup>, 2001, *MPEG-7 Visual Descriptors*. <[www.ece.ucsb.edu/~manj/ece181bS04/L18\(MPEG7\).ppt](http://www.ece.ucsb.edu/~manj/ece181bS04/L18(MPEG7).ppt)>, (Juni 2001, accessed 26 Mei 2008).
- Munir, Rinaldi, 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika Bandung, Bandung.
- Nugroho, Adi, 2005, *Rational Rose untuk Pemodelan Berorientasi Objek*, Informatika Bandung, Bandung.
- Schmuller, Joseph, 1999, *Teach Yourself UML in 24 Hours*, Sams Publishing.
- Suhendar, dkk, 2002, *Visual Modelling Menggunakan UML dan Rational Rose*, Informatika Bandung, Bandung.
- Suryani, Emma, 2008, *Aplikasi Pencarian Citra Memanfaatkan Apache Lucene dan Mpeg-7 Color Descriptor*, Teknik Informatika - UPN "Veteran" Yogyakarta.