

Rancang Bangun Aplikasi Pengelompokan dan Pemberi Rekomendasi Berita Lomba Online Menggunakan Klasifikasi Fuzzy Berbasis Kerangka Kerja Spring

Febri Fernanda, Umi Laili Yuhana, dan Diana Purwitasari

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: yuhana@its-sby.edu

Abstrak—Saat ini sebagian besar orang mendapatkan berita yang diinginkan melalui internet. Sebaliknya, orang yang ingin mempublikasikan berita yang ditulis juga melalui internet. Salah satu berita yang beredar dan banyak dibutuhkan adalah berita lomba. Pada umumnya, untuk mendapatkan berita lomba setiap orang akan melakukan pencarian pada mesin pencari. Hasil yang didapatkan dari pencarian masih terlalu umum dan beberapa menghasilkan data yang tidak relevan. Untuk itu bermunculan beberapa portal lomba yang memuat berita-berita lomba yang tersebar di internet. Namun, pengguna masih harus tetap membuka halaman dan melakukan pencariannya pada masing-masing portal. Dikarenakan setiap portal mempunyai data berita lomba yang berbeda-beda, maka waktu yang diperlukan untuk mendapatkan berita lomba yang sesuai kurang efektif. Pada penelitian ini dikembangkan suatu sistem berbasis web yang secara berkala mampu memperbarui dan mengelompokkan kategori data berita lomba dari beberapa portal lomba. Pengambilan judul dan rangkuman data lomba memanfaatkan layanan *Really Simple Syndication* (RSS) sedangkan untuk mendapatkan konten data lomba memanfaatkan teknologi web *crawler*. Sistem berupa sebuah mesin pencari berita lomba dengan beberapa fungsi penyaringan. Dengan tambahan fitur pemberi rekomendasi berita berdasarkan profil pengguna dan sejarah pencarian, maka sistem ini dapat memudahkan pengguna mendapatkan berita lomba yang diinginkan secara cepat. Sistem dibangun menggunakan kerangka kerja Spring MVC agar memudahkan dalam pembangunan dan penggunaan ulang. Pengelompokkan data berita lomba menggunakan metode *Fuzzy Similarity K Nearest Neighbors* (FSKNN) yang mampu mengelompokkan berita lomba ke dalam beberapa kategori sekaligus. Untuk membangun fitur mesin pencari dan pemberi rekomendasi berita lomba sistem memanfaatkan pustaka Lucene.

Kata Kunci—*Fuzzy Similarity, K Nearest Neighbors, Klasifikasi Berita, Lucene, Spring*

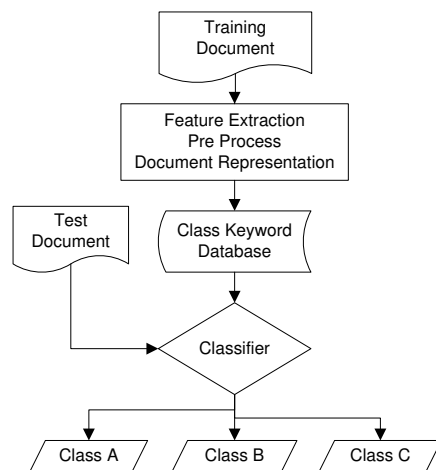
I. PENDAHULUAN

Saat ini penyebaran berita lewat internet berkembang dengan sangat cepat. Salah satunya adalah berita tentang lomba yang diselenggarakan oleh suatu organisasi atau institusi. Permasalahan yang muncul adalah pengguna tidak mudah mengunjungi situs yang menampilkan informasi lomba tersebut. Dengan hanya memanfaatkan situs mesin pencari, situs yang dimaksud tidak mudah untuk ditemukan. Beberapa hasil pencarian berita lomba pada situs pencari memang tepat menampilkan situs-situs yang merupakan

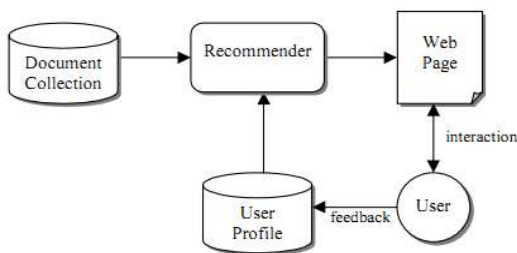
promosi lomba, namun dari hasil pencarian ditampilkan juga situs yang hanya memuat artikel berita tentang lomba yang sudah dilaksanakan, situs yang berupa sebuah iklan, situs cerita pribadi atau blog, situs video *sharing*, situs jejaring sosial dan memungkinkan juga ditampilkan situs ‘tipuan’ yang hanya ingin mengeruk keuntungan dengan banyaknya lalu lintas pengunjung yang mengunjungi situsnya.

Untuk menangani masalah tersebut, saat ini telah bermunculan portal-portal yang khusus menampilkan informasi tentang lomba-lomba yang sedang diselenggarakan. Portal tersebut mendapatkan berita lomba dari pencarian sendiri oleh pengelolaanya, info dan kiriman berita dari pengguna lain serta kerjasama dengan berbagai institusi terkait sehingga portal tersebut dapat dianggap mewakili persebaran berita lomba yang ada di internet. Pada portal tersebut, pengguna harus membuka tautan pada portal satu persatu lalu baru bisa melakukan pencarian dan penyaringan jenis lomba yang sesuai dengan keinginan. Walaupun ada label pada masing-masing berita yang bisa dianggap sebagai pengelompokan, namun masing-masing portal tidak mempunyai standard yang sama dalam memberikan label, sehingga portal satu dengan yang lain berbeda-beda cara pemanfaatannya. Hal itu membuat waktu akses pengguna kurang efektif, terlebih jika ditambah dengan waktu untuk membuka satu persatu portal lomba yang tersedia.

Dari permasalahan di atas, pada penelitian ini akan dikembangkan sistem yang berkala dapat memperbarui dan mengelompokkan kategori data berita lomba dari berbagai portal lomba yang ada. Sistem mengambil data melalui RSS [1] *feed* portal lalu melakukan proses *crawling* ke URL berita lomba. Sistem juga dapat memberikan rekomendasi lomba kepada pengguna yang didasarkan atas data profil pengguna dan sejarah pencarian pada sistem. FSKNN digunakan untuk melakukan pengelompokan kategori berita lomba. Metode tersebut mampu mengelompokkan berita secara multilabel sehingga sangat sesuai digunakan karena berita lomba bisa memiliki lebih dari satu label. FSKNN dapat diandalkan karena optimasinya lebih baik dari Pohon Keputusan C4.5 dan MLKNN [2]. Hasil pengelompokan dimanfaatkan sebagai variabel penyaring pada fitur pencarian dan pemberi rekomendasi dengan memanfaatkan pustaka Lucene. Kerangka kerja Spring MVC akan digunakan dalam



Gambar 1. Alur Proses Sistem Pengelompokan Dokumen [4]



Gambar 2. Alur Proses Content Based Filtering [5]

pembangunan sistem, sebab merupakan salah satu kerangka kerja berbasis pemrograman Java yang cukup populer dan handal [3].

II. DASAR TEORI

A. Sistem Pengelompokan Berita

Sistem pengelompokan berita merupakan sistem yang berfungsi untuk mengelompokkan berita dalam kelas-kelas yang sesuai. Ada dua proses utama dari sistem ini, pelatihan dan uji coba. Pelatihan digunakan untuk membentuk sebuah kerangka pengelompokan berdasarkan data dokumen latih. Sedangkan pengujian digunakan untuk memberikan kelas yang sesuai terhadap dokumen yang belum terkelompokkan. Alur proses ideal dari sistem pengelompokan dokumen ditunjukkan pada Gambar 1.

B. Sistem Rekomendasi Berita Lomba

Sistem rekomendasi berita lomba merupakan sistem yang bertujuan memperkirakan informasi yang menarik bagi penggunanya dan juga membantu pengguna dalam memutuskan jenis berita lomba yang sesuai atau diinginkan olehnya. Sistem rekomendasi secara khusus merupakan suatu sistem pemberian saran kepada penggunanya dan bersifat personal, berbeda untuk semua pengguna sistem. Rekomendasi yang diberikan pada sistem rekomendasi dibangun berdasarkan atribut-atribut yang dimiliki oleh pengguna, semisal kegemaran, keahlian, pekerjaan dan sebagainya.

Beberapa algoritma telah diusulkan untuk sistem rekomendasi, salah satunya adalah *content-based filtering*[5] dimana alur prosesnya ditunjukkan pada Gambar 2.

Penelitian ini mengadopsi alur proses pada *content-based filtering*. Algoritma yang digunakan memanfaatkan ilmu temu kembali informasi, yakni dengan mencari nilai kesamaan dari dokumen-dokumen berita lomba dengan data profil dari pengguna. Implementasi algoritma memanfaatkan pustaka Lucene.

C. Spring Web MVC

Spring Web MVC merupakan bagian dari modul Spring Web *Servlet* yang ada pada kerangka kerja Spring. Kerangka kerja Spring Web MVC dirancang di sekitar *DispatcherServlet* yang mengirimkan permintaan ke *handler*.

Ada tiga cara kerja utama Spring Web MVC [3].

1. DispatcherServlet

- Kerangka Spring Web MVC dirancang di sebuah *DispatcherServlet*.
- Menangani permintaan (*request*) yang didasarkan pada pengaturan di *file Extensible Markup Language (XML)*.

2. Mapping Request

- Konfigurasi pemetaan (*mapping*) untuk pembangunan aplikasi web tersimpan dalam *web.xml*.

3. Spring IoC Container

- Merupakan inti dari kerangka kerja Spring.
- Berupa *file* konfigurasi untuk manajemen objek-objek yang ada di dalam aplikasi.

Spring Web MVC memisahkan lapisan elemen pembangunan menjadi elemen model, elemen *view*, dan elemen *controller*. Pemisahan tersebut dapat dijadikan standar dalam pembuatan kode program sehingga memudahkan dalam perawatan dan pengembangan ulang perangkat lunak. Aliran kerja proses Spring Web MVC ditunjukkan pada Gambar 3.

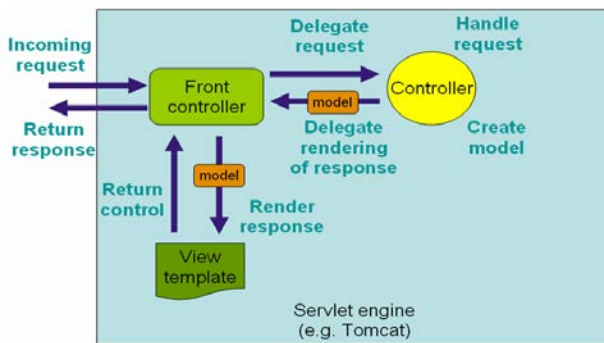
D. Fuzzy Similarity K Nearest Neighbors (FSKNN)

FSKNN merupakan algoritma klasifikasi dokumen secara multilabel. Algoritma ini menyempurnakan algoritma MLKNN dengan perbaikan di sisi komputasi [2] karena penambahan algoritma *clustering* untuk pencarian *k*-tetangga terdekat.

Secara umum, FSKNN terdiri dari dua bagian :

1. Pelatihan data

Proses pelatihan data digunakan untuk mengelompokkan dokumen ke dalam kategori-kategori yang mungkin masih terjadi ketidaksesuaian pada kategori aslinya. Proses ini membentuk desain yang digunakan dalam pencarian kategori suatu dokumen yang belum diketahui kategorinya. Hasil pelatihan digunakan sebagai acuan untuk menentukan klasifikasi pada dokumen yang sebenarnya. Pada bagian ini terdapat dua tahap proses, yaitu proses pengelompokkan data latih ke dalam kategori dan perhitungan nilai *prior* dan *likelihood* yang digunakan untuk membentuk sebuah desain dalam penentuan kategori dokumen baru.



Gambar 1. Aliran Kerja Proses Spring Web MVC [6]

Proses-proses pada pelatihan data sebagai berikut.

- a. Pengelompokan data latih ke dalam kategori-kategori

Tahap ini bertujuan untuk mendapatkan derajat keanggotaan suatu dokumen terhadap kategori. Tahap ini yang diyakini mampu memperbaiki sisi komputasi dari metode-metode sebelumnya.

 1. Menghitung distribusi term pada setiap kategori.
 2. Menghitung nilai keanggotaan setiap term pada setiap kategori.
 3. Menghitung nilai keanggotaan setiap term terhadap setiap dokumen.
 4. Menghitung *fuzzy similarity* setiap dokumen terhadap setiap kategori.
 5. Menghitung nilai keanggotaan dokumen pada setiap kategori.
 6. Melakukan *clustering* dokumen ke dalam kategori yang sesuai. Untuk setiap dokumen, apabila nilai derajat keanggotaan pada suatu kategori lebih dari *threshold*, maka dokumen tersebut merupakan anggota pada kategori yang dimaksud.
 7. Membentuk himpunan pencarian pada setiap dokumen.
- b. Menghitung nilai *prior* dan *likelihood*
 1. Menghitung nilai *prior*.
 2. Menemukan *k-nearest neighbours* (*k* tetangga terdekat) dokumen pada himpunan pencarian yang telah terbentuk. Perhitungan ini memanfaatkan pembobotan tf-idf dan persamaan kosinus
 3. Menghitung nilai *likelihood*.
2. Pengujian

Proses pengujian digunakan untuk mengelompokkan dokumen baru ke dalam kategori yang sesuai:

 - a. Menghitung nilai keanggotaan dokumen pada kategori memanfaatkan nilai yang didapat pada proses pelatihan data.
 - b. Pengecekan nilai keanggotaan dokumen pada kategori. Jika nilai lebih besar dari *threshold*, maka dokumen tersebut diklasifikasikan ke dalam kategori tersebut.
 - c. Membentuk himpunan pencarian dari dokumen tersebut.
 - d. Menemukan *k* tetangga terdekat dokumen pada himpunan pencarian yang telah terbentuk.
 - e. Menghitung nilai *label vector* dokumen untuk setiap kategori berdasarkan nilai *prior* dan *likelihood*, jika

nilai *label vector* adalah 1 maka dokumen merupakan anggota kategori, sebaliknya jika nilai 0 maka dokumen tidak termasuk ke dalam kategori.

E. Lucene

Lucene merupakan pustaka untuk mesin pencari yang bersifat *open source* [7]. Lucene dapat mengindeks informasi berbasis teks dan pencarian berdasarkan berbagai jenis penyaringan berdasarkan kriteria. Lucene memiliki kemampuan pencarian yang sangat fleksibel dan kuat, menggunakan logika Fuzzy untuk menemukan item dari indeks. Lucene memiliki kelas-kelas utama yang digunakan untuk membangun mesin pencari.

1. Kelas Document mewakili sebuah dokumen pada Lucene. Lucene mengindeks objek dokumen dan mendapatkan objek dokumen kembali ketika melakukan pencarian.
2. Kelas Field merupakan bagian dari kelas Document. Objek Field akan berisi nama dari suatu bagian beserta data aktualnya.
3. Kelas Analyzer adalah kelas abstrak yang digunakan untuk menyediakan sebuah *interface* yang mampu memproses kelas Document dan mengubahnya menjadi token sehingga dapat diindeks. Ada banyak implementasi dari kelas ini tapi yang paling umum digunakan adalah SimpleAnalyzer dan StandardAnalyzer.
4. Kelas IndexWriter digunakan untuk menciptakan dan memelihara indeks.
5. Kelas IndexSearcher digunakan untuk mencari dokumen melalui indeks.
6. Kelas QueryParser digunakan untuk membangun parser yang digunakan untuk pencarian melalui indeks.
7. Kelas Query adalah kelas abstrak yang berisi kriteria pencarian yang dibuat oleh QueryParser.
8. Kelas Hits berisi objek dokumen yang dikembalikan dari proses pencarian dengan kelas Query pada indeks.

III. ANALISIS DAN PERANCANGAN

A. Perancangan Kasus Penggunaan

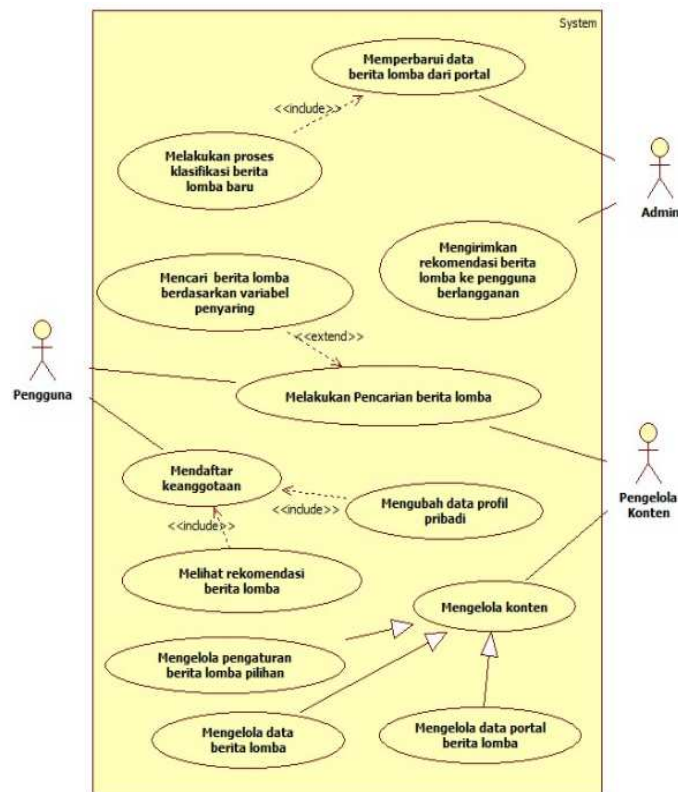
Kebutuhan fungsional berisi kumpulan proses bisnis dalam perangkat lunak yang harus dipenuhi. Kebutuhan fungsional mendeskripsikan fitur-fitur yang dimiliki sistem. Daftar kebutuhan fungsional sistem ditunjukkan pada Tabel 1. Sedangkan representasi kebutuhan fungsional pada diagram kasus penggunaan ditampilkan pada Gambar 4.

B. Perancangan Arsitektur Sistem

Sistem dibangun menggunakan pengembangan terpisah antara aplikasi klien dan *server*. Sisi klien adalah aplikasi yang dapat diakses melalui perambah situs web, sedangkan sisi *server* berupa aplikasi yang terinstal pada komputer *server*. Diagram arsitektur sistem ditunjukkan pada Gambar 5. Dalam diagram tersebut, pengguna yang dimaksud adalah pengguna akhir dari sistem yang tidak mempunyai hak akses istimewa. Pengelola adalah pengguna yang mempunyai hak akses istimewa untuk dapat melakukan perawatan konten seperti manajemen portal dan manajemen berita. Sedangkan

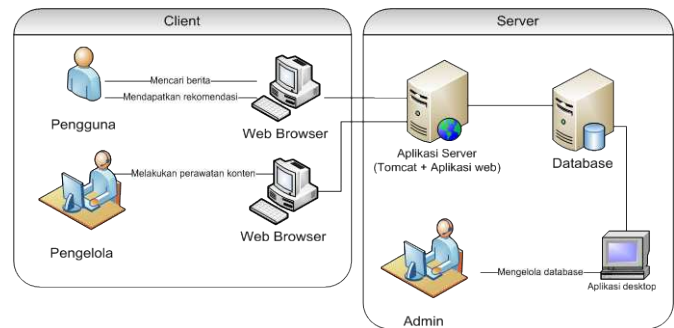
Tabel 1.
Kebutuhan Fungsional Sistem

Kebutuhan Fungsional	Deskripsi
Memperbarui data berita lomba	Sistem mampu memperbarui data berita lomba dengan membaca data pada portal lomba
Melakukan proses klasifikasi berita lomba	Sistem mampu mengklasifikasi berita lomba ke dalam kategori tertentu
Mengirimkan rekomen-dasi berita lomba	Sistem mampu mengirimkan rekomendasi lomba ke pengguna melalui surat elektronik
Mencari berita lomba	Pengguna dapat melakukan pencarian berita lomba
Mendaftar keanggotaan	Pengguna dapat mendaftarkan keanggotaanya ke dalam sistem
Mengubah data profil	Pengguna terdaftar dapat mengubah data profil pribadinya
Melihat rekomendasi berita lomba	Pengguna dapat melihat rekomendasi berita lomba yang sesuai dengan proil pribadi dan sejarah pencarian
Mengelola konten sistem	Pengelola dapat mengelola konten sistem yang meliputi data portal lomba, data berita lomba, dan data pengaturan berita pilihan

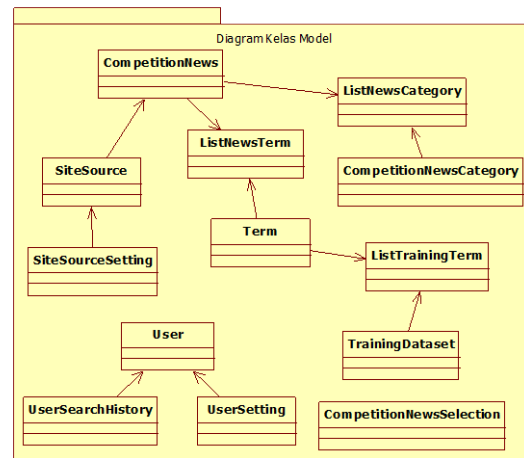


Gambar 4. Diagram Kasus Penggunaan

admin adalah pengguna yang mengoperasikan aplikasi *desktop* pada *server*. Aplikasi *desktop* mempunyai fitur untuk melakukan klasifikasi yang meliputi beberapa proses seperti pra pemrosesan teks, penghitungan bobot dan penghitungan nilai untuk klasifikasi.



Gambar 5. Arsitektur Kerja Sistem



Gambar 6. Diagram Kelas Sistem

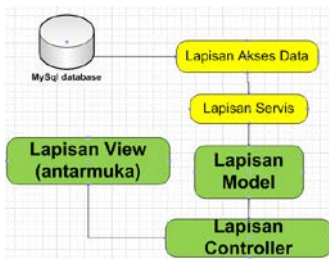
C. Perancangan Diagram Kelas

Perancangan dari hubungan antara Kelas-Kelas utama yang diperlukan dalam membangun sistem digambarkan dengan diagram kelas pada Gambar 6.

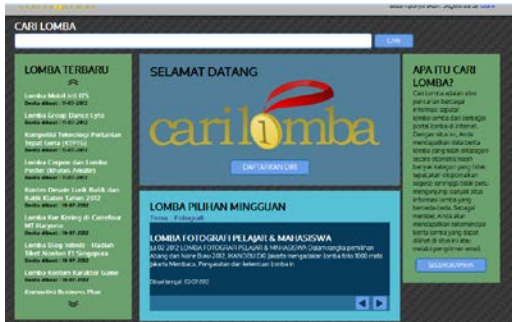
D. Perancangan Arsitektur Pembangunan Perangkat Lunak

Perangkat lunak dibangun dengan menggunakan arsitektur 3-tier yaitu arsitektur MVC yang terdapat pembagian layer menjadi lapisan model, *view*, dan *controller*. Dalam prakteknya, terdapat beberapa lapisan pendukung, diantaranya adalah lapisan data akses dan lapisan servis. Diagram arsitektur MVC yang telah disesuaikan dengan lapisan pendukung sistem dapat dilihat pada Gambar 7.

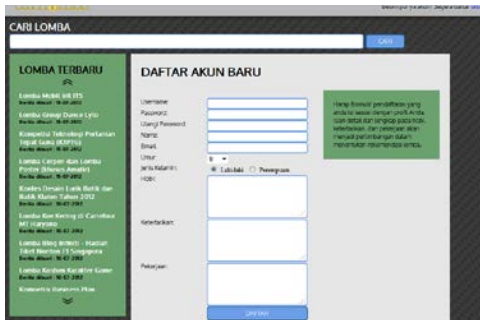
Lapisan Model merepresentasikan tabel-tabel pada basis data dalam bentuk kelas. Lapisan View berupa lapisan yang menghubungkan sistem dengan pengguna. Lapisan ini menghasilkan bentuk antarmuka. Lapisan Controller berisi logika-logika bisnis yang ada pada perangkat lunak. Lapisan ini berhubungan erat dengan lapisan *view* karena pemanggilan fungsi-fungsi dalam lapisan ini adalah melalui lapisan *view*. Lapisan Akses Data merupakan penghubung antara servis dengan lapisan basis data. Lapisan ini mengurus segala perubahan basis data, termasuk pembacaan dan penyimpanan. Semua perubahan basis data dilakukan dengan menggunakan objek *instance* dari kelas lapisan model. Lapisan Servis merupakan penghubung antara lapisan akses data dengan lapisan Model. Lapisan ini berisi fungsi-fungsi logika manipulasi Model yang akan disalurkan ke akses data.



Gambar 7. Diagram Arsitektur MVC Termodifikasi



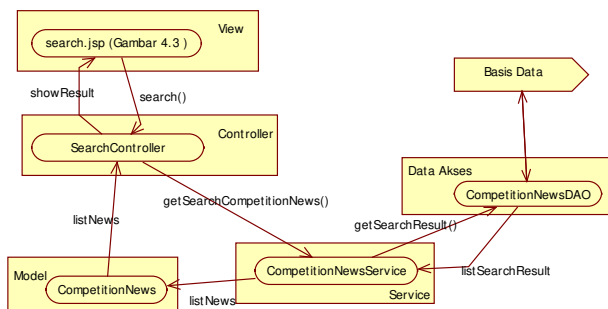
Gambar 8. Layar Utama Aplikasi



Gambar 9. Layar Pendaftaran Aplikasi



Gambar 10. Layar Pencarian Aplikasi



Gambar 11. Alur MVC Pencarian Berita Lomba

Tabel 2.
Hasil Pengujian Fungsionalitas

ID	Nama	Hasil
UC-01	Uji Coba Melakukan Proses Klasifikasi Berita Lomba	Berhasil
UC-02	Uji Coba Memperbarui Data Berita Lomba dari Portal	Berhasil
UC-03	Uji Coba Mengirimkan Rekomendasi Lomba ke Pengguna Berlangganan	Berhasil
UC-04	Uji Coba Melakukan Pencarian Berita Lomba	Berhasil
UC-05	Uji Coba Mendaftar Keanggotaan	Berhasil
UC-06	Uji Coba Melihat Rekomendasi	Berhasil
UC-07	Uji Coba Mengubah Data Profil	Berhasil

IV. IMPLEMENTASI

Pada aplikasi klien, sistem dibangun menggunakan kerangka kerja Spring sedangkan pada *server* menggunakan Java SE. Implementasi yang dilakukan berupa implementasi kode dan antarmuka. Dalam bab ini, dijelaskan beberapa hasil implementasi antarmuka dari sistem. Gambar 8 – Gambar 10 merupakan beberapa cuplikan antarmuka sistem pada sisi klien. Sedangkan Gambar 11 menggambarkan salah satu contoh alur implementasi MVC Spring dari proses utama.

V. PENGUJIAN DAN EVALUASI

A. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan melihat apakah fitur-fitur yang disediakan aplikasi berjalan sebagaimana mestinya Hasil pengujian fungsionalitas sistem terangkum dalam Tabel 2.

B. Pengujian Non Fungsionalitas

Pengujian non fungsionalitas dilakukan dengan menguji akurasi hasil klasifikasi berita lomba yang dilakukan oleh aplikasi. Pengujian dilakukan dengan melakukan klasifikasi terhadap data berita lomba baru yang didapat dari portal lomba. Jumlah berita lomba baru adalah 161 yang didapat dari 7 portal. Beberapa skenario inialisasi nilai *threshold* dan nilai *k* diuji coba untuk mendapatkan hasil yang terbaik. Rangkuman hasil pengujian non fungsionalitas perangkat lunak ditampilkan pada Tabel 3. Hasil tersebut didapat dengan nilai *k*=10 dan *threshold* =0.8.

Beberapa hipotesis faktor penyebab kurang akuratnya hasil klasifikasi berita lomba sebagai berikut:

1. Beberapa konten berita lomba yang didapat dari portal bukan merupakan konten tentang berita lomba secara utuh, kadang hanya berupa sedikit penjelasan sumber berita atau hanya menampilkan poster saja. Beberapa lomba bahkan ada yang menggunakan bahasa Inggris. Hal tersebut menyebabkan *corpus* term yang dimiliki oleh berita lomba kurang relevan.

Tabel 3.
Hasil Pengujian Akurasi

Kategori	Hasil	
	Jumlah data	Persentase
Prediksi kategori sama tepat dengan hasil klasifikasi	263	66.92 %
Prediksi kategori sama sebagian dengan hasil klasifikasi (Berupa himpunan bagian)	46	11.70 %
Prediksi kategori sama sebagian dengan hasil klasifikasi (Berupa himpunan irisan)	3	0.76 %
Prediksi kategori sama sekali tidak mirip dengan hasil klasifikasi	81	20.61 %
TOTAL	393	99.99 %

- Daftar *stopwords* yang telah dibuat masih kurang optimal, kemungkinan ada term yang sebenarnya diperlukan tetapi tidak bisa digunakan karena masuk ke dalam *stopwords* atau sebaliknya.
- Data latih yang digunakan untuk melatih data klasifikasi belum mewakili semua kategori secara merata.

C. Pengujian Penggunaan

Pengujian penggunaan dilakukan untuk memastikan bahwa perangkat lunak yang dibangun tercapai manfaatnya dalam memenuhi kebutuhan pengguna. Pengujian penggunaan dilakukan di lingkungan teknik informatika ITS, dimana perangkat lunak *server* ditempatkan pada komputer *server* laboratorium Rekayasa Perangkat Lunak. Tujuan dari pengujian ini adalah untuk menilai penggunaan antarmuka sistem, aliran informasi dan kebermanfaatan perangkat lunak secara objektif. Sasaran pengguna untuk pengujian adalah mahasiswa jurusan teknik informatika. Pengujian ini memanfaatkan kuisioner.

Secara umum, sebagian besar pengguna mengemukakan pendapat bahwasannya perangkat lunak yang dibangun merupakan perangkat lunak berbasis web yang terorganisir dengan baik. Pengguna merasa dimudahkan dengan adanya perangkat lunak yang dapat membantu mereka menemukan lomba yang sesuai. Pengguna juga menganggap bahwa mereka cocok dengan perangkat lunak sehingga pengguna tertarik untuk menggunakannya dikemudian hari dan merekomendasikan teman-teman lainnya untuk menggunakan perangkat lunak ini.

D. Pengujian Indeks Kappa

Pengujian indeks kappa digunakan untuk menguji konsistensi dari pelabelan data hasil pengelompokan (yang dijadikan sebagai *gold set*). Nilai indeks kappa didasarkan atas dua pakar yang menilai label data pengujian menurut pengetahuan masing-masing. Dari kedua penilaian tersebut maka dihitung nilai indeks kappa-nya untuk disimpulkan apakah pelabelan data pengujian dapat diterima. Indeks kappa dihitung menggunakan persamaan:

$$k = \frac{P(A) - P(E)}{1 - P(E)} \quad (1)$$

dimana $P(A)$ adalah proporsi berapa kali pakar saling setuju dan $P(E)$ merupakan proporsi pakar sama-sama setuju hanya karena kebetulan.

Jumlah data pengujian yang digunakan dibatasi sejumlah 115 sebagai sampel dengan jumlah kategori 5. Dari matriks dan perhitungan formula indeks kappa didapat $k=0.924$ yang menunjukkan bahwa penilaian label dari pakar memiliki tingkat konsistensi yang tinggi dan dapat diterima.

VI. KESIMPULAN

Berdasarkan hasil pengamatan selama proses perancangan, implementasi dan pengujian perangkat lunak dalam penelitian ini, dapat diambil kesimpulan sebagai berikut.

- Kerangka kerja Spring Web MVC dapat diimplementasikan dengan baik ke dalam aplikasi web untuk aplikasi klien perangkat lunak.
- Metode FSKNN dapat diimplementasikan dengan baik dan bisa berjalan sesuai dengan spesifikasinya.
- Proses pembacaan RSS *feed* dan *crawling* ke URL terkait sehingga data berita lomba bisa didapatkan secara otomatis dapat terimplementasikan dengan baik.
- Pustaka Lucene dapat terimplementasi dengan baik untuk membangun sistem mesin pencari dan pemberi rekomendasi.
- Hasil pengujian hasil klasifikasi bisa disebut cukup akurat dengan persentase kebenaran sempurna sebesar 66.92 %, kesalahan sempurna sebesar 20.61 % dan sisanya merupakan kebenaran parsial.
- Hasil pengujian yang dilakukan menunjukkan fungsionalitas dan penggunaan perangkat lunak berjalan dengan tepat guna sesuai tujuan dan manfaat.

UCAPAN TERIMA KASIH

Para Penulis mengucapkan terima kasih kepada Allah SWT atas limpahan rahmat dan hidayah-Nya dan semua pihak yang telah membantu dalam penelitian maupun penulisan makalah ini.

DAFTAR PUSTAKA

- RSS Advisory Board. *Really Simple Syndication Specifications, Tutorials and Discussion*, URL: <http://www.rssboard.org/rss-specification>, (2012, Mei).
- T Jiang, Jung-Yi, Tsai, Shian-Chi, Lee, Shie-Jue. *FSKNN: Multi-label Text Categorization based on Fuzzy Similarity and K Nearest Neighbors*, (2011).
- B. Gunawan, *Apa itu Spring Framework*, URL: <http://budigunawan.wordpress.com/2010/02/01/apa-itu-spring-framework>, (2012, Juni)
- D-A, Chiang, *Expert Systems with Applications*, (2008).
- R.V. Meteren, M.V. Someren, *Using Content-Based Filtering for Recommendation*.
- R. Johnson, *Spring Framework Reference Documentation*, (2004).
- The Apache Software Foundation. *Apache Lucene Core*, URL: <http://lucene.apache.org/core/>, (2012, Mei).