Aplikasi Pencarian Karya Tulis Ilmiah Berbasis *Web* Menggunakan Sistem Rekomendasi

Husni
Program Studi Teknik Informatika, Universitas Trunojoyo
Jl. Telang Raya PO BOX 2 Kamal, Bangkalan 69162
E-mail: *husni@if.trunojoyo.ac.id*

Abstrak

Penelitian ini mencoba untuk membangun suatu sistem pencarian karya tulis ilmiah berbasis web memanfaatkan teknologi sistem rekomendasi. Setiap pengguna dapat mencari karya tulis tertentu sesuai dengan kata kunci. Pengguna kemudian akan ditawarkan beberapa penelitian lain yang terkait dengan karya tulis yang sedang diakses. Pada penelitian ini, dari beberapa percobaan yang telah dilakukan, namun dengan jumlah paper kurang dari lima puluh, aplikasi web yang dibangun sudah memperlihatkan hasil yang baik. Beberapa kesalahan masih ditemukan. Kadang pencarian kemiripan antara query dengan daftar paper atau kemiripan antara satu paper dengan paper lainnya memberikan nilai kemiripan yang besar, di atas 70%, padahal tingkat kemiripan sebenarnya tidak sebesar itu. Terdapat paper dengan derajat kemiripan rendah padahal sebenarnya merupakan paper yang sangat dekat dengan paper yang sedang dibaca. Kesalahan-kesalahan ini, kemungkinan besar disebabkan tidak dilibatkannya abstrak dalam proses komputasi kemiripan. Teks yang terdapat pada judul pada beberapa kasus tidak merupakan representasi dari uraian paper. Penggunaan abstrak diyakini dapat mempertajam hasil kemiripan yang diberikan oleh sistem yang secara garis besar sudah berjalan baik ini.

Kata kunci: kemiripan karya ilmiah, aplikasi web, sistem rekomendasi

Abstract

This research attempt to develop a web□based search system of scientific writing using recommendation system technology. Each user can search for certain papers in accordance with the keyword. Users will then be offered several other papers related to the paper that is being accessed. In this study, from several experiments have been conducted, but with the amount of paper is less than fifty, the web applications have shown good results. Some errors are still found. Sometimes search for similarity between the query with a paper list or a paper with other paper gives a great similarity value, above 70%, although the similarity is actually not that big. There is a paper with a low degree of similarity when it actually is a paper that is very close to the paper being read. These errors, most likely due not involved paper abstract in the process of similarity computing. The text contained in the title in some cases do not constitute a representation of paper content. The use of abstract in similarity computing is believed to sharpen the results given by a system that largely has been running this well.

Keywords: *paper similarity, web application, recommendation system*

Pendahuluan

Situs web di Internet adalah alat yang sangat berdayaguna bagi pemiliknya. Suatu institusi dapat menjadikan situs web sebagai penjual paling depan dalam pemasaran produk-produknya. Seorang konsultan dapat memanfaatkan situs web untuk membangun komunikasi dengan lebih banyak client yang tersebar di seluruh dunia. Para peneliti dapat mempublikasikan karya ilmiah hasil temuannya kepada peneliti lain atau khalayak melalui situs web.

Informasi yang terdapat di dalam situs web telah digunakan sebagai parameter dalam menentukan ranking internasional suatu lembaga pendidikan tinggi. Salah satu lembaga populer yang menjadikan situs web sebagai tolok ukur kualitas suatu Universitas adalah Webometrics. Semakin banyak informasi yang diletakkan di situs web Universitas maka akan semakin tinggi ranking Webometrics yang akan diperoleh. Selain itu, semakin banyak orang yang mengakses dan menggunakan informasi yang disediakan akan turut menaikkan ranking situs web tersebut. Dan informasi paling berharga dalam penentuan ranking adalah laporan hasil penelitian atau karya tulis ilmiah. Semakin banyak karya tulis pada situs tersebut

dijadikan acuan atau rujukan penelitian atau penulisan karya ilmiah maka akan semakin cepat kenaikan ranking suatu Universitas.

Pembangunan situs web yang memuat hasil atau laporan penelitian sebaiknya tidak sekedar bertujuan menyampaikan informasi kepada pengunjungnya. Hampir semua Universitas di Indonesia hanya memberikan informasi mengenai hibah penelitian yang diperoleh para penelitinya, kegiatan rutin yang dilakukan dan sedikit abstrak laporan penelitian. Seharusnya situs web Universitas menyediakan informasi yang lebih komprehensif, misalnya *soft-copy* dari ringkasan laporan penelitian, *draft paper* para peneliti yang telah dimuat di suatu jurnal ilmiah tetapi bukan nomor dan volume terbaru, *paper* yang tidak dimuat di jurnal ilmiah, karya tulis ilmiah mahasiswa, laporan pengabdian IPTEKS di masyarakat dan berbagai solusi yang ditempuh saat pelaksanaan Kuliah Kerja Nyata (KKN). Selain kelengkapan informasi, situs ini juga perlu memberikan kemudahan kepada pengunjung terutama dalam mendapatkan abstrak atau tulisan lengkap yang saling terkait. Jika pengunjung mendapatkan paper berjudul "Penanaman Semangka di Lahan Kering" maka sangat mungkin pengunjung tersebut memerlukan paper yang terkait dengan "Pengelolaan lahan kering", "Perbandingan panen lahan kering dan basah" atau "Potensi pertanian di kabupaten Bangkalan".

Penelitian ini mencoba untuk membangun suatu situs web yang dapat digunakan untuk menjawab dua masalah utama tersebut. Hasilnya adalah sebuah sistem berbasis isi (*content-based*) yang informatif, mudah digunakan oleh operator maupun pengunjung dan memuat berbagai karya tulis ilmiah yang terdapat di lingkungan Universitas.

Metodologi Penelitian

Tugas utama di dalam sistem berbasis *content* yang dibangun adalah menghitung atau mencari tingkat kemiripan antara *content* (dokumen) dengan *query* pengguna memanfaatkan teknik temu balik informasi. Efisiensi temu balik dari *content* yang didasarkan pada *query* diperoleh dengan pembuatan *index* [2]. Secara umum, pembuatan index dilakukan dalam 5 langkah, yaitu *markup and format removal*, *tokenization*, *filtration*, *stemming* dan *weighting*. Jika *content* diletakkan di dalam database maka tahapan *markup removal* dan *weighting* kadang dapat dihilangkan, namun pada koleksi dokumen web kelima langkah tersebut harus aplikasikan [3].

Pada tahap *markup and format removal* semua tag *markup* dan format khusus dihapus dari dokumen, terutama pada dokumen yang mempunyai banyak tag dan format seperti dokumen HTML. Pada tahap *tokenization*, semua teks yang telah sederhanakan tersebut diubah ke bentuk huruf kecil (*lowe case*) dan semua tanda baca dihilangkan. Berikutnya adalah *filtration*, yaitu proses memutuskan *term* mana yang akan digunakan untuk merepresentasikan dokumen sehingga dapat digunakan untuk mendeskripsikan isi dokumen dan membedakan dokumen tersebut dari dokumen lain di dalam koleksi. *Term* yang sering dipakai tidak dapat digunakan untuk tujuan ini, setidaknya karena dua hal. Pertama, jumlah dokumen yang relevan terhadap suatu query kemungkinan besar merupakan bagian kecil dari koleksi. *Term* yang efektif dalam pemisahan dokumen yang relevan dari tidak relevan kemungkinan besar adalah term yang muncul pada sedikit dokumen. Ini berarti bahwa *term* dengan frekuensi tinggi merupakan *poor descriminator*. Kedua, term yang muncul dalam banyak dokumen tidak mencerminkan definisi dari topik atau sub-topik dokumen. Karena itu, *term* yang sering digunakan dan *stop-word* dihapus. Namun, menghapus *stop-word* dalam suatu koleksi dokumen pada satu waktu akan sangat lama. Solusinya adalah dengan menyusun suatu pustaka *stop-word* atau *stop-list* dari *term* yang akan dihapus.

Tahap *stemming* mengacu kepada proses mereduksi suatu term ke term asli atau varian aslinya. Misalnya, "*computer*" dan "*computer*" diubah menjadi "*comput*" sedangkan "*walks*", "*walking*" dan "*walker*" diubah menjadi "*walk*". Pada bahasa inggris, *stemmer* yang paling popular adalah algoritma *stemming* Martin Porter [4, 5].

Tahap terakhir adalah *weighting* atau pembobotan. *Term* dibobot sesuai dengan model pembobotan yang dipilih, dapat *local weighting*, *global weighting* atau keduanya. *Local weighting* diekspresikan sebagai *term frequency*, *tf*. *Global weighting* ditentukan oleh nilai *idf* (*inverse document frequency*). Banyak skema *weighting* menggunakan kombinasi keduanya, *tf* * *idf*.

Pada model ruang vektor, dokumen ditampilkan sebagai suatu vektor. Pada sistem berbasis *content*, posisi suatu dokumen ditentukan oleh *keyword* yang terdapat di dalamnya [1]. Dokumen dapat berupa *query* yang mengandung *keyword list*, *record* database atau dokumen web. Jika terdapat dua dokumen A dan B pada suatu ruang vektor, maka jarak atau kemiripan kedua dokumen tersebut dapat dihitung menggunakan *cosine similarity* dengan rumus:

$$sim(A,B) = \cos(\theta) = \frac{A \bullet B}{|A||B|} \quad \text{..... (1)}$$

Di mana A • B adalah *dot product* dari vektor dokumen A dan B, |A| dan |B| adalah jarak *euclidean* dari vektor tersebut. Semakin besar nilai cosinus sudut θ berarti semakin dekat kedua vektor, menunjukkan bahwa dua dokumen tersebut juga sangat mirip.

Akurasi dari *cosine similarity* akan berkurang saat dokumen yang diproses mempunyai banyak *term* berulang dan hadir pada banyak dokumen lain. Ini diperbaiki dengan menyertakan bobot (*weight*) dalam perhitungan kemiripan antardokumen. Pendekatan umum adalah menggunakan metode *tf x idf*. Bobot *term j* dalam dokumen *i* dirumuskan sebagai:[2]

$$w_{i,j} = tf_{i,j} \, x \, idf_j = tf_{i,j} \, x \, log \, N/df_j \quad \text{..... (2)}$$

Dapat pula ditulis [3]:

$$w_{i,j} = tf_{i,j} \, x \, idf_j = tf_{i,j} \, x \, log \, ((N - df_j)/df_j) \quad \text{..... (3)}$$

Di mana *N* adalah jumlah dokumen dalam koleksi dan df_j adalah jumlah dokumen yang mengandung term *j*. Metode ini mengakibatkan bobot term menjadi tinggi jika term tersebut sering muncul pada sebagian kecil dokumen di dalam koleksi.

Pada aplikasi berbasis *query* , pengguna memasukkan suatu *query* dan mengharapkan beberapa dokumen yang relevan dengan *query* tersebut. Rumus *cosine similarity* berikut dapat digunakan untuk menghitung kemiripan antara suatu *query Q* dengan beberapa dokumen D_i :

$$sim(Q,D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} w_{i,j}}{\sqrt{\sum_{j=1}^{V} w^2_{Q,j}} \sqrt{\sum_{j=1}^{V} w^2_{i,j}}} \quad \text{..... (4)}$$

Di mana $w_{Q,j}$ adalah bobot term j dalam query yang didefinisikan sebagai $tf_{Q,j} \, x \, idf_j$. Penyebut dalam persamaan ini dinamakan faktor normalisasi yang mengabaikan pengaruh panjang dokumen pada *score* dokumen. Dengan begitu, suatu dokumen yang mengandung {*x, y, z*} akan mempunyai *score* sama dengan dokumen lain yang berisi {*x, x, y, y, z, z*}.

Karya tulis ilmiah atau *paper* secara umum terdiri dari Judul, Informasi Penulis termasuk nama institusi yang diwakilinya, alamat kantor dan alamat email yang dapat dihubungi. Kemudian terdapat abstrak yang bisanya ditulis dalam bahasa Inggris, bahkan beberapa paper mengharuskan ditulis dalam dua bahasa: Inggris dan Indonesia. Abstrak merupakan intisari dari seluruh uraian di dalam paper, dimulai dari rangkuman latar belakang masalah, penyelesaian yang telah ada, solusi yang ditawarkan di dalam paper tersebut dan kesimpulannya. Setelah Abstrak terdapat kata kunci atau keyword yaitu kata paling penting yang mencerminkan di mana paper tersebut berada dan semua kata kunci umumnya sudah digunakan di dalam Abstrak. Isi paper atau uraian dari hasil pemikiran atau penelitian di tulis di bawahnya. Uraian ini dimulai dari pendahuluan atau latar belakang masalah, tinjauan referensi berisi berbagai solusi yang sedang banyak digunakan dan di mana posisi dari kasus yang diangkat (*state-of-the-art*), analisis dan desain, implementasi yang dilakukan, diikuti kesimpulan serta saran yang dapat dilakukan lebih lanjut. Sebuah daftar referensi atau daftar pustaka menutup *paper* ini secara lengkap.

Dalam perhitungan kemiripan, tidak semua bagian dari *paper* dilibatkan. Secara umum, dapat dikatakan bahwa Judul, Abstrak dan Kata kunci merupakan bagian yang paling penting dan sangat dibutuhkan. Tanpa judul, apa jadinya sebuah *paper*. Abstrak merupakan cerminan atau rangkuman dari semua yang dibahas di dalam *paper*. Dari abstrak dapat diketahui mengapa paper tersebut ada, apa yang dilakukan di dalamnya dan apa hasil akhirnya. Penting atau tidaknya suatu paper bagi seseorang dapat diketahui dari melihat abstraknya. Kata kunci merupakan bagian dari abstrak, yaitu kata-kata yang paling penting yang dapat mewakili paper tersebut. Kata kunci ini sangat penting di dalam proses pencarian. Uraian *paper* memang penting, namun itu setelah pengguna menemukan paper yang tepat berdasarkan rangkuman pada abstrak.

Hasil dan Pembahasan

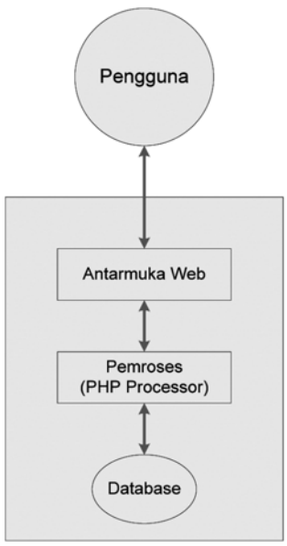
Penelitian ini hanya melibatkan Judul dari paper selama proses perhitungan kemiripan, baik kemiripan antara *query* dengan daftar paper maupun kemiripan antara satu *paper* dengan *paper* lainnya.

Aplikasi web yang dihasilkan akan memudahkan pencarian karya tulis ilmiah. Pengguna cukup memasukkan suatu kata kunci dan semua paper yang mirip dengan kata kunci tersebut ditampilkan. Sistem juga menampilkan sebanyak *n* paper terbaru yang dimasukkan ke dalam sistem, dan sebanyak n *paper* yang paling banyak dibaca oleh pengguna. Dari daftar paper yang ditampilkan, pengguna dapat memilih salah satu dan kemudian membacanya. Sistem akan menampilkan pula daftar *paper* yang terkait erat dengan paper yang sedang dibaca.

Pengguna akan memperoleh suatu antarmuka berbasis web yang dapat digunakan untuk menemukan *paper* terbaik sesuai kebutuhannya. Pengguna memilih salah satu paper yang disediakan oleh sistem atau memasukkan kata kunci tertentu di kotak pencarian yang disediakan. Selanjutnya, oleh *web browser*, permintaan dari pengguna akan dikirimkan ke *web server*. *Web server* bekerja sama dengan pemroses berupa suatu interpreter (bahasa pemrograman PHP) memroses permintaan pengguna. Jika diperlukan, PHP Pemroses melakukan komunikasi dengan server Database. Dalam penelitian ini digunakan server database MySQL. Secara garis besar, arsitektur sistem ini diperlihatkan pada Gambar 1.

Fitur penting yang disediakan oleh sistem ini adalah:

1. Menampilkan informasi *call for paper* dari penelitian atau seminar nasional/Internasional yang akan hadir.
2. Menampilkan *n* judul paper terbaru (terakhir kali dimasukkan oleh operator). *Default*nya nilai *n* adalah 5.
3. Menampilkan *n* judul *paper* yang paling banyak dibaca oleh pengguna. Defaultnya nilai *n* adalah 5. Fitur ini memunugkinkan pengguna mendapatkan paper yang banyak dijadikan acuan oleh pengguna lain.
4. Menampilkan judul *paper* yang sesuai dengan kata kunci pencarian. Ini memungkinkan pengguna menemukan paper yang mengandung kata tertentu. Tidak semua paper ditampilkan, hanya paper dengan derajat kemiripan di atas 20% yang ditampilkan.
5. Menampilkan judul *paper* yang terkait erat dengan *paper* yang sedang ditampilkan (di baca). Hanya paper dengan tingkat kemiripan di atas 30% yang ditampilkan oleh sistem.
6. Menampilkan detail dari suatu *paper* yang dipilih oleh pengguna. Jika pengguna memilih salah satu judul paper yang disediakan sebelumnya, sistem akan menampilkan isi lengkap dari paper tersebut mencakup Judul, Informasi Penulis, Abstrak, Kata kunci dan URL (*Uniform Resource Locator*) di mana file berisi uraian lengkap (isi) paper dapat didownload. Tidak semua paper mempunyai atribut file, ini terkait dengan perijinan. Jika penulis paper membolehkan karya tulisnya dipublikasikan secara bebas maka file tersebut tersedia, jika tidak hanya berhenti pada Judul, Penulis, Abstrak dan Kata kunci.



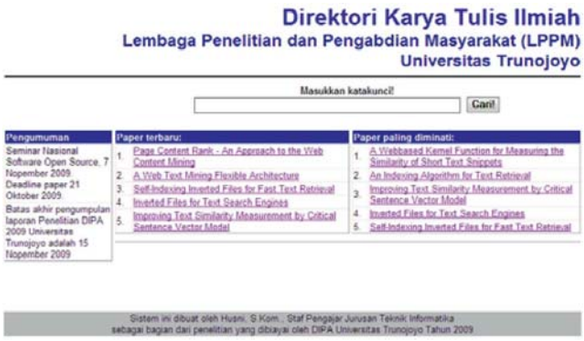
Gambar 1. Arsitektur Sistem Pencarian Karya Tulis Ilmiah

Secara garis besar, proses pencarian kemiripan antara *query* dengan daftar *paper*, sementara hanya melibatkan judul *paper*, adalah sebagai berikut:

1. Ambil query dari pengguna
2. Ambil judul dari semua paper
3. Lakukan perbandingan antara judul terpilih dengan *query*, simpan hasil perhitungan kemiripannya di dalam tabel sementara. Lakukan berulang sampai ke *paper* terakhir
4. Ambil semua *record* di dalam tabel sementara yang nilai kemiripannya dengan *query* lebih dari 20%. Urutkan hasilnya secara *Descending* (menurun, dari besar ke kecil)
5. Tampilkan kepada pengguna (*web browser*).

Sedangkan proses pencarian kemiripan antara judul *paper* yang sedang dibaca dengan daftar *paper*, sementara hanya melibatkan judul *paper*, adalah sebagai berikut:

1. Ambil judul paper yang sedang ditampilkan
2. Ambil judul dari semua *paper*
3. Lakukan perbandingan antara judul terpilih dengan judul *paper* yang sedang dibaca, simpan hasil perhitungan kemiripannya di dalam tabel sementara. Lakukan berulang sampai ke paper terakhir
4. Ambil semua *record* di dalam tabel sementara yang nilai kemiripannya dengan judul paper yang sedang dibaca di atas 30% tetapi tidak termasuk *paper* yang sedang dibaca. Urutkan hasilnya secara *Descending* (menurun, dari besar ke kecil)
5. Tampilkan kepada pengguna (web browser).



Gambar 2. Tampilan awal aplikasi Pencarian Karya Tulis Ilmiah

Saat pertama dibuka, aplikasi pencarian Karya Tulis Ilmiah berbasis web ini menampilkan pengumuman berupa informasi *call for paper* dari seminar nasional/ Internasional yang akan diselenggarakan atau Jurnal Ilmiah yang akan terbit, pada sisi kiri. Pada bagian kanan ditampilkan n judul paper yang paling akhir dimasukkan, kemudian diikuti n judul paper yang paling sering diakses atau ditampilkan oleh pengguna. Defaultnya, nilai n adalah 5.

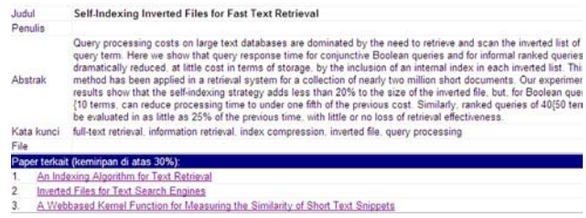
Pada bagian lebih atas, ditampilkan kotak isian dan sebuah tombol Cari. Pada kotak isian tersebut, pengguna dapat memasukkan suatu frase atau *keyword* untuk menemukan paper yang sesuai dengan kebutuhan. Tampilan awal ini diperlihatkan pada Gambar 2.

Potongan kode program yang berfungsi menampilkan n judul *paper* yang paling sering diakses adalah:

```
//tampilkan 5 penelitian paling sering diakses
$query = "SELECT Id, Title FROM paper ORDER BY Click DESC LIMIT 5";
$result = mysql_query($query, $db);
$count = 0;
while($row = mysql_fetch_row($result)) {
    $count++;
    print('<tr>
    <td width="7%" align="left"><font size="2" face="Arial">' . $count . '
    <td width="87%"><a href="index.php?q=show&id=' . $row[0] .'">
    <font face="Arial" size="2">' . $row[1] . '</font></a></td></tr>');
}
```

Tampilan detail menampilkan informasi lengkap dari suatu *paper* ditambah apa yang disebutkan pada tampilkan awal. Informasi lengkap mencakup Judul, Informasi Penulis, Abstrak, Kata kunci, dan lokasi File yang dapat didownload jika ada (biasanya berformat PDF). Contoh dari tampilan detail dari salah satu *paper* diperlihatkan pada Gambar 3.

Di bawah detail dari paper, ditampilkan daftar judul dari paper yang terkait dengan paper yang sedang ditampilkan detailnya (dianggap sedang dibaca oleh pengguna). Daftar judul paper yang ditampilkan telah



Gambar 3. Tampilan detail dari suatu paper

Gambar 4. Hasil pencarian untuk query "Web Mining"

	Id	Title	Sim
	1	A Webbased Kernel Function for Measuring the Similarity of Short Text Snippets	17.9775
	2	An Indexing Algorithm for Text Retrieval	16
	3	Improving Text Similarity Measurement by Critical Sentence Vector Model	7.40741
	4	Inverted Files for Text Search Engines	16.6667
	5	Self-Indexing Inverted Files for Fast Text Retrieval	16.129
	6	A Web Text Mining Flexible Architecture	40.8163
	7	Page Content Rank - An Approach to the Web Content Mining	29.9507

Gambar 5. Daftar semua paper dalam database dan nilai kemiripannya dengan query "Web Mining"

diurutkan secara *descending* sehingga judul yang paling atas merupakan *paper* yang paling mirip dengan *paper* yang sedang dibaca.

Fitur pencarian difasilitasi oleh kotak isian pencarian dan tombol Cari yang terdapat pada bagian atas web aplikasi pencarian Karya Tulis Ilmiah. Sistem akan mencari *paper* yang sesuai dengan kata kunci yang dimasukkan dalam kotak isian pencarian dan kemudian menampilkan hasilnya yang telah diurutkan secara *descending* kepada pengguna. Urutan *descending* ini mengakibatkan judul paper paling atas mempunyai kemiripan paling besar dengan *query* yang diberikan pengguna. Hanya *paper* dengan tingkat kemiripan di atas 20% yang ditampilkan.

Gambar 4 memperlihatkan hasil pencarian untuk *query* "Web Mining".

Sebagaimana telah disebutkan pada bagian analisis dan desain proses pencarian kemiripan mengikuti beberapa langkah. Salah satunya adalah membentuk tabel temporer (tabel sementara) yang menampung kemiripan setiap paper dengan query yang dimasukkan pengguna. Daftar kemiripan yang tersimpan di dalam database (tabel temp), pada pencarian menggunakan keyword "Web Mining" diperlihatkan pada gambar 5.

Daftar *paper* yang ditampilkan pada halaman web adalah yang mempunyai kemiripan di atas 20%, yaitu *record* nomor 6 dan 7 yang diurutkan

secara *Descending* (besar ke kecil), sebagaimana diperlihatkan Gambar 4.

Potongan kode program berikut memperlihatkan langkah-langkah yang disebutkan di atas:

```
//ambil semua record dalam tabel paper
delquery = "DELETE FROM temp";
mysql_query($delquery) or die ("Ada kesalahan dalam pemrosesan data.");

query = "SELECT Id, Title FROM paper";
result = mysql_query($query, $db);

while($row = mysql_fetch_row($result)) {
   $sim = similar_text($_POST['teks'], $row[1], $percent);
   $addquery = "INSERT INTO temp (Id, Title, Sim) VALUES ($row[0], '$row[1]', $percent)
   mysql_query($addquery) or die ("Ada kesalahan dalam pemrosesan data.");

query = "SELECT Id, Title FROM temp WHERE Sim > 20 ORDER BY Sim DESC";
result = mysql_query($query, $db);
count = 0;
while($row = mysql_fetch_row($result)) {
   $count++;
   print('<tr><td width="4%" align="left"><font size="2" face="Arial">' . $count . '.</
   <td width="96%"><a href="index.php?q=showid=' . $row[0] .'">
   <font face="Arial" size="2">' . $row[1] . '</font></a></td></tr>');
```

Kemiripan antarpaper pada dasarnya sama dengan kemiripan antara *paper* dengan *query*. Di sini, *query* diganti dengan Judul dari *paper* yang sedang dibaca. Berdasarkan judul tersebut, dilakukan proses sebagaimana diuraikan dalam sub-bab analisis dan desain. Gambar 3 memperlihatkan hasil yang pencarian kemiripan untuk *paper* berjudul "*Self-Indexing Inverted File for Fast text Retrieval*".

Ukuran kemiripan antara *record-record* di dalam database dengan paper berjudul "*Self-Indexing Inverted File for Fast text Retrieval*" diperlihatkan pada gambar 6 dan yang ditampilkan hanya yang mempunyai derajat kesamaan di atas 30% tetapi tidak termasuk *paper* itu sendiri, yaitu yang mencapai 100%.

Id	Title	Sim
1	A Webbased Kernel Function for Measuring the Simi...	33.5878
2	An Indexing Algorithm for Text Retrieval	65.2174
3	Improving Text Similarity Measurement by Critical ...	24.3902
4	Inverted Files for Text Search Engines	62.2222
5	Self-Indexing Inverted Files for Fast Text Retriev...	100
6	A Web Text Mining Flexible Architecture	24.1758
7	Page Content Rank - An Approach to the Web Content...	12.844

Gambar 6. Daftar semua paper dan nilai kemiripannya dengan paper berjudul "*Self-Indexing Inverted File for Fast text Retrieval*"

Simpulan

Pemanfaatan sistem rekomendasi dalam pencarian karya tulis ilmiah sangat membantu pengguna mendapatkan karya tulis yang sesuai dengan kebutuhan. Aplikasi yang dibangun masih melakukan perbandingan antarjudul paper untuk mengetahui tingkat kemiripan antarkarya ilmiah. Meskipun hasil yang diberikan sudah cukup baik, hasil yang lebih baik diperkirakan akan diperoleh jika melibatkan kata kunci dan abstrak dalam komputasinya. Selanjutnya, aplikasi ini akan dikembangkan untuk memanfaatkan perhitungan kemiripan *cosine similarity* dengan berbagai variasinya sekaligus melibatkan elemen paling penting dari *paper*, yaitu abstrak.

Daftar Pustaka

[1] Lee, D.L., 1997. "Document Ranking and the Vector-Space Model", *IEEE* March-April 1997, diunduh dari *citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.195, diakses 10 Januari 2010.*

[2] Chu, W., Liu, Z., Mao, W.. 2002, "*Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining*". Diunduh dari *citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.2314, diakes 20 Januari 2010.*

[3] **Garcia, E., 2005. "***Document Indexing Tutorial for Information Retrieval Students and Search Engine Marketers***", Diunduh dari** http://www.miislita.com/information-retrieval-tutorial/indexing.html, Diakses **10 Januari 2010.**

[4] Willett, P., 2006. " The Porter stemming algorithm: then and now" *Electronic Library and Information Systems*, 40 (3). halaman 219–223. Diunduh dari http://eprints.whiterose.ac.uk/1434/01/willettp9_PorterStemmingReview.pdf, diakses 22 Januari 2010.

[5] Joshua, S.E., 2005. "*English Stemming Algorithm*", *diunduh dari: www.pr-sol.com/whitepapers/English Stemming Algorithm.pdf, Diakses 22 Januari 2010.*