Sistem Informasi Tugas Akhir Menggunakan Model Ruang Vektor (Studi Kasus: Jurusan Sistem Informasi)

Wahyudi,MT
Laboratorium Sistem Informasi Fakultas Sains dan Teknologi UINSUSKA RIAU
Jl.HR.Subrantas KM.15 Panam Pekanbaru Telp. 0761-8359937
wahyudi@uin-suska.ac.id

Abstrak

Tugas Akhir merupakan salah satu syarat penting bagi seorang mahasiswa untuk meraih gelar sarjana, seorang mahasiswa perlu mencari referensi dalam pembuatan tugas akhir, selama ini banyak mahasiswa mencari referensi dari judul dan isi tugas akhir yang sudah ada, permasalahannya adalah jika selama ini mahasiswa kesulitan mencari referensi yang sesuai dan menghabiskan banyak watu untuk membaca referensi yang ada karena banyaknya jumlah tugas akhir (TA) yang tersedia

Pencarian terutama di dalam sistem informasi telah banyak mempergunakan konsep basis data dengan objek data berupa tabel. Permasalahannya bahwa konsep ini adalah konsep yang kaku karena pembuatan query dan pembentukan relasi antar query bersifat artificial (buatan) sehingga pengguna sistem merasa seolah-olah dibatasi dengan aturan-aturan yang telah dibuat, tanpa memiliki kewenangan untuk melakukan modifikasi query sendiri

.

Pada Sistem informasi Tugas Akhir ini kata / kalimat yang dimasukkan user akan dibandingkan Similarity Coefficient (kesamaan) nya dengan abstrak TA yang sudah ada . Cara untuk menentukan nilai Similarity Coefficient adalah dengan menggunakan model ruang vektor yaitu kalimat dari user dan abstrak TA diumpamakan sebagai vektor sehingga diketahui nilai term frequency (tf) dan inverse term frequency (idf) , dari nilai tf dan idf-nya barulah bisa diketahui nilai Similarity Coefficient (SC).

***Kata Kunci** : Query, Similarity Coefficient, Tugas Akhir*

1. PENDAHULUAN

Manusia tidak pernah luput dan lepas dari namanya permasalahan pencarian, dimanapun dan kapanpun selalu ada model pencarian yang secara mutlak diperlukan. Dalam komputer, pencarian juga menjadi hal yang sangat penting di dalamnya, dimana seluruh aktifitas IT selalu berkaitan dengan hal tersebut. Bidang jaringan komputer, seperti pencarian alamat DNS yang telah tersimpan dalam *cache*, *traffic snaffing*, dan lain-lain. Bidang basis data, yang tentunya hampir 90 % melibatkan pencarian didalam prosesnya, bidang grafik, seperti *rendering*, *edge detection* dan lain-lain yang secara tidak langsung juga melibatkan proses pencarian didalamnya. Konsep pencarian terutama di dalam SI (Sistem Informasi) telah banyak memprgunakan konsep basis data didalamnya. Basis data merupakan konsep penyimpanan informasi yang terpilah menjadi data-data kecil (*field*) yang tentunya berguna untuk mempercepat model pencarian. Hal ini sangat baik sekali terutama untuk kebutuhan pencarian yang sangat cepat. Hanya saja permasalahannya bahwa konsep ini adalah konsep yang kaku. Penulis mengungkapkan hal ini karena menurut penulis kelemahan dalam basis data adalah pembuatan *query* dan pembentukan relasi antar *query* yang bersifat *Artificial* (Buatan) sehingga pengguna sistem merasa seolah-olah terbatasi dengan aturan-aturan yang telah dibuat, tanpa memiliki kewenangan untuk melakukan modifikasi *query* sendiri.

Disamping itu, pemilihan struktur data dan algoritma merupakan permasalahan yang kritis dalam disain sistem yang memungkinkan temu kembali dengan basis data berukuran besar secara efektif dan efisien. Untuk itu, agar memperoleh algoritma yang tepat dalam membangun suatu sistem informasi, perlu dilakukan studi mengenai model algoritma pembangun sistem temu kembali informasi (*information retrieval*).

2. TEMU KEMBALI INFORMASI

Information Retrieval adalah suatu sistem yang menemukan (*retrieve*) informasi yang sesuai dengan kebutuhan pengguna dari kumpulan informasi secara otomatis. Salah satu aplikasi dari *Information Retrieval* adalah mesin pencari yang dapat diterapkan diberbagai bidang. Pada mesin pencari dengan *information retrieval* pengguna dapat memasukkan *query* yang bebas dalam arti kata *query* yang sesuai dengan bahasa manusia dan Sistem dapat menemukan dokumen yang sesuai dengan *query* yang ditulis oleh user.

Adapun fungsi utama Sistem Temu Kembali Informasi seperti dikemukakan oleh Lancaster (1979) dan Kent (1971) adalah sebagai berikut:

1. Mengidentifikasi sumber informasi yang relevan dengan minat masyarakat pengguna yang ditargetkan.
2. Menganalisis isi sumber informasi (dokumen).
3. Merepresentasikan isi sumber informasi dengan cara tertentu yang memungkinkan untuk dipertemukan dengan pertanyaan (*query*) pengguna.
4. Merepresentasikan pertanyaan (*query*) pengguna dengan cara tertentu yang memungkinkan untuk dipertemukan sumber informasi yang terdapat dalam basis data (kesesuaian).
5. Mempertemukan pernyataan pencarian dengan data yang tersimpan dalam basis data.
6. Menemu-kembalikan informasi yang relevan.
7. Menyempurnakan unjuk kerja sistem berdasarkan umpan balik yang diberikan oleh pengguna.

Sistem temu kembali informasi terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian pula ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan sistem temu kembali informasi dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut. Sebagai suatu sistem, sistem temu kembali informasi memiliki beberapa bagian yang membangun sistem secara keseluruhan

3. METODOLOGI PENELITIAN

Penelitian dicoba di angkat dalam penelitian ini menerapkan konsep pengolahan bahasa alami dalam pencarian dokumen yang sesuai dengan yang diinginkan pengguna Pada sistem informasi tugas khir ini terdapat beberapa proses yaitu :

1. Proses Koleksi Dokumen (*Collection Document*)
   Pada penelitian ini, dokumen yang digunakan adalah kumpulan dari beberapa abstrak laporan Tugas Akhir. Teks dokumen tersebut disimpan dalam sebuah *file* eksternal dan masing-masing diberi indeks secara berurutan. Pada penelitian ini, pengaturan teks dokumen dan pemberian indeks dilakukan secara otomatis.
2. Proses Kosa Kata Sebagai Istilah Indeks (*Text Operations* dan *Indexing*)
   Teks dokumen yang telah tersimpan kemudian dibentuk menjadi kosa kata sebagai istilah indeks.

Pada *information retrieval*, tahap untuk menghasilkan kumpulan kosa kata dibutuhkan proses *parsing* (penguraian) dan *stoplist*.

1. *Parsing* (penguraian). Proses ini berfungsi mengurai suatu rangkaian kalimat dalam dokumen menjadi kumpulan kosa kata.
2. *Stoplist*. Kata-kata buang merupakan kata-kata yang tidak memiliki kemampuan dalam membedakan dokumen yang satu dengan yang lainnya. Daftar kata *stoplist* telah ditentukan dan disimpan sebelumnya oleh pembangun sistem.
3. Proses Pembobotan Kata
   Tahap selanjutnya adalah memberikan nilai atau bobot pada masing-masing kata. Pada model ruang vektor, untuk menghitung bobot istilah indeks adalah berdasarkan istilah yang sering muncul dalam sebuah dokumen atau dikenal dengan istilah *term frequency* (tf) dan jumlah kemunculannya dalam koleksi dokumen yang disebut *inverse document frequency* (idf). Kemudian dihitung bobot masing-masing kata atau *term* tersebut dengan rumus:

$$idfj = \log\left|\frac{d}{dfj}\right| \quad \text{.....(1)}$$

   dengan ketentuan:
   Idfj = *inverse document frequency* tj
   d = total dokumen
   dfj = jumlah dukumen yang mengandung istilah tj

4. Proses Operasi Teks *Query* (*Text Operations Query*)
   Alur operasi selanjutnya adalah alur dari pihak *user* yakni melakukan pencarian dengan memasukkan *query* pada antarmuka (*interface*) sistem yang tersedia.

5. Proses Pembobotan *Query*

   Setelah *query* dimasukkan oleh *user*, maka dilakukan proses pembobotan *query* seperti pada proses pembobotan kata pada dokumen (vektor *query*) dengan menggunakan rumus 1

   Sebagai contoh, *query* yang di*input*kan *user* adalah “Jaringan Komputer” , dan 3 buah koleksi dokumen (D1, D2, D3) yaitu:
   D1 : Teknologi Komputer banyak digunakan di segala bidang
   D2 : Perkembangan komputer sangat pesat terutama di bidang jaringan
   D3 : Bahasa pemrograman Java menjadi bahasa yang cepat menarik perhatian programmer
   Istilah “Komputer” muncul pada 2 buah dokumen dengan total dokumen 3 sehingga nilai idf adalah:

$$idf = \log\left|\frac{3}{2}\right| = 0.176$$

Istilah "Jaringan" muncul pada 1 buah dokumen dengan total dokumen 3 sehingga nilai idf adalah:

$$idf = \log\left|\frac{3}{1}\right| = 0.477$$

6. *similarity coeffcient* (SC)

Proses penyesuaian adalah menghitung kedekatan antara vektor dokumen dengan vektor *query* dengan menggunakan persamaan *similarity coeffcient* (SC) (rumus 2).

$$SC(Q, Di) = \frac{\sum_{j=1}^{t}(wqj \quad dij)}{\sqrt{\sum_{j=1}^{t}(wqj)^2 \sum_{j=1}^{t}(dij)^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

dimana:
wqj = bobot istilah j pada *query* q = $frek_{qj}$ * idf_j
d_{ij} = bobot istilah j pada dokumen i = tf_{ij} * idf_j
tf_{ij} = *term frequency* = kemunculan istilah t_j pada dokumen D_i

Berdasarkan pada contoh proses pembobotan *query* maka:
Pada D1 : tf "jaringan"= 0 dan tf "komputer"= 1
Pada D2 : tf "jaringan"= 1 dan tf "komputer"= 1
Pada D3 : tf "jaringan"= 0 dan tf "komputer"= 0

Maka hasil penyesuaian berdasarkan rumus 2 adalah:

a. Sigma (Q, D1) = $(wq_1 * d_{11}) + (wq_2 * d_{12})$
= **[** $(frek_{q1} * idf_1) * (tf_{11} * idf_1)$ **]** + **[** $(frek_{q2} * idf_2) * (tf_{12} * idf_2)$ **]**
= **[**(1 * 0.477)(0 * 0.477) **]** + **[** (1 * 0.176)(1 * 0.176) **]**
= 0.031

$$SC(Q, D1) = \frac{0.031}{\sqrt{[(0.477)^2 + (0.176)^2] * [(0)^2 + (0.176)^2]}}$$

= 0.319
Sehingga hasil SC (Q, D1) = **0.319**

b. Sigma (Q, D2) = $(wq_1 * d_{11}) + (wq_2 * d_{12})$
= **[** $(frek_{q1} * idf_1) * (tf_{11} * idf_1)$ **]** + **[** $(frek_{q2} * idf_2) * (tf_{12} * idf_2)$ **]**
= **[**(1 * 0.477)(1 * 0.477) **]** +**[** (1 * 0.176)(1 * 0.176) **]**
= 0.258

$$SC(Q, D2) = \frac{0.258}{\sqrt{[(0.477)^2 + (0.176)^2] * [(0.477)^2 + (0.176)^2]}}$$

= 1.000
Sehingga hasil SC (Q, D2) = **1.000**

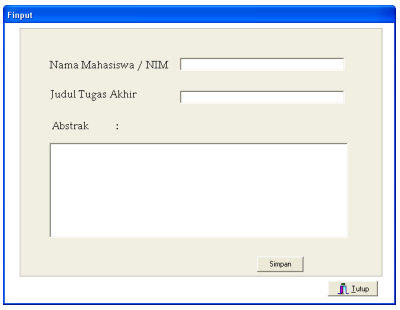
c. Sigma (Q, D3) = $(wq_1 * d_{11}) + (wq_2 * d_{12})$
= **[** $(frek_{q1} * idf_1) * (tf_{11} * idf_1)$ **]** + **[** $(frek_{q2} * idf_2) * (tf_{12} * idf_2)$ **]**
= **[**(1 * 0.477)(0 * 0.176) **]** + **[** (1 * 0.176)(0 * 0.176) **]**
= 0.000

$$SC(Q, D2) = \frac{0}{\sqrt{[(0.477)^2 + (0)^2] * [(0.176)^2 + (0)^2]}}$$

= 0.000
Sehingga hasil SC (Q, D3) = **0.000**

4. IMPLEMENTASI

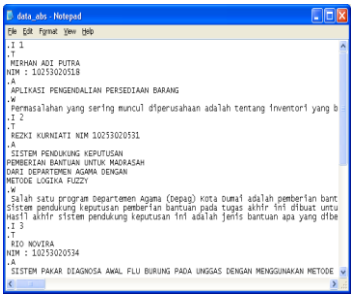
Metodologi penelitian telah dirancang untuk membentuk perangkat lunak Sistem Informasi Tugas Akhir, pada bab berikut akan diberikan uraian singkat implementasi perangkat lunak yang telah dibuat. Tidak semua *interface* perangkat lunak ditampilkan dalam makalah ini. Penelitian yang dilakukan adalah membuat suatu mesin pencari untuk mengetahui informasi Tugas Akhir. Pertama kali *admin* mengisi form seperti gambar 1 di bawah ini.



Gambar 1. Masukan data TA

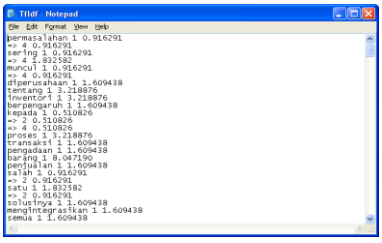
Setelah itu maka data yang disikan akan disimpan dalam sistem basis data

Dari basis data diatas akan dibuat suatu file berbentuk teks yang terdiri dari banyak dokumen. Tiap tiap dokumen dalam teks tersebut terdiri dari nomor dokumen nama dan nim, judul TA, Abstrak, seperti gambar 2. Proses pembentukan teks tersebut dilakukan pada kolom-kolom tertentu (gambar 2) yang berhubungan dengan kasus *retrieval* terhadap informasi yang akan dicari dan di *cluster*.



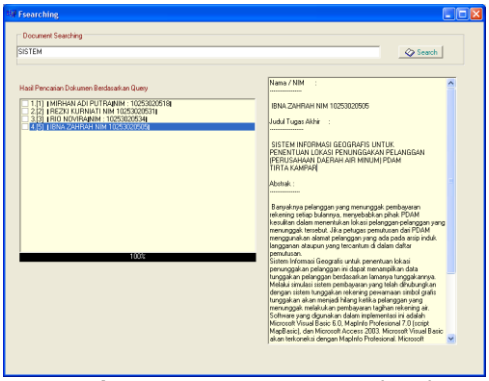
Gambar 2. Pembentukan teks eksternal untuk *pra processing* temu kembali informasi

Dari dokumen di atas akan diolah menjadi *inverted file* yang isinya dapat dilihat pada gambar 3. Proses pembentukan tersebut dilakukan sebagai fungsi atomic penunjuk (pointer) antar kata dan dokumen dan kemunculan kata tersebut di dokumen teks yang telah dibentuk. file dibawah merupakan file yang sudah dilakukan pembobotan untuk mendapatkan tf.idf nya



Gambar 3. *Inverted file* yang telah dibentuk menggunakan model Tf.Idf

Setelah nilai idf nya diperoleh maka dilakukan perangkingan dari yang terbesar sampai yang terkecil (*descending*). Sistem juga menyediakan form untuk mesin pencari dalam menemukan dokumen yang relevan dengan *query* yang diinginkan oleh pengguna (gambar 4).



Gambar 4. Form utama pencarian data TA

Form pencarian natural pada gambar 4 terdiri dari pemasukan query, hasil pencarian terhadap seluruh TA yang ditemukan yang sesuai dengan query yang diinginkan oleh pengguna, dan juga informasi lengkap tentang data TA termasuk Judul dan Abstrak lengkap. Dari *query* yang dimasukkan maka akan dilakukan proses pembobotan terhadap dokumen yang ada. Pembobotan otomatis biasanya berdasarkan istilah yang sering muncul dalam sebuah dokumen atau dikenal dengan istilah *term frequency(tf)* dan jumlah kemunculannya dalam koleksi dokumen disebut *inverse document frequency(idf)*. Bobot suatu istilah makin kecil jika istilah tersebut muncul dalam banyak dokumen dan makin besar jika sering muncul pada suatu dokumen.

Untuk mengukur efektifitas dua rasio umum yang biasa digunakan adalah precision dan recall. Precision adalah ukuran kemampuan suatu sistem untuk menampilkan hanya dokumen relevan. Recall adalah kemampuan suatu sistem untuk menampilkan seluruh dokumen yang relevan.

$$Precision = \frac{\text{jumlah dokumen relevan yg berhasil ditemukan}}{\text{Jumlah dokumen yang ditemukan}}$$

$$Recall = \frac{\text{jumlah dokumen relevan yg berhasil ditemukan}}{\text{Jumlah dokumen relevan dalam koleksi}}$$

Nilai *recall* dan *precision* selalu berbanding terbalik, semakin tinggi nilai *recall* semakin rendah nilai *precision*, begitu juga sebaliknya semakin rendah nilai *recall* semakin tinggi nilai *precision*. *Precision* dapat dihitung pada berbagai titik *recall.* Secara umum, semakin tinggi nilai recall semakin banyak jumlah dokumen yang harus dicari. Pada mesin pencarian yang sempurna, hasil pencarian semuanya merupakan dokumen yang relevan atau dengan kata lain pada setiap nilai *recall*, nilai precision selalu 1.00. pada kenyataannya, ada dokumen yang tidak relevan juga diambil oleh mesin pencari.

5. PENUTUP

Model yang diusulkan pada pencarian dokumen sangat berguna bagi dosen dan mahasiswa yang memerlukan data TA. Hasil yang didapat dengan model yang diusulkan ini memiliki tingkat akurasi pengelompokan cukup baik (sekitar 89 prosen untuk tingkat recall dan 60 prosen untuk tingkat precisionnya). Diperkirakan tidak digunakannya stemming berpengaruh terhadap hasil pembentukan kelompok data, walaupun menurut penulis hanya berkurang sekitar 2-5 prosen saja (cukup kecil).

Daftar Pustaka

[1] Ackerman, Rich, "*Vector Model of Information Rretrieval*", http://www.hray.com/ , 2008
[2] Andi, "*Pengembangan Perangkat Lunak Simulasi dengan C++ Builder*", halaman 7, Wahana Komputer, Semarang, 2004
[3] Harjono, Kristopher David, "*Perluasan Vektor pada Metode Search Vector Space*", Jurnal Ilmu Komputer fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Parahyangan , Vol. 10 No. 2, 2005
[4] Hasugian, Jonner, "*Penggunaan Bahasa Alamiah dan Kosa Kata Terkontrol dalam Sistem Temu Kembali Informasi berbasis teks*", Ilmu Perpustakaan Fakultas Sastra USU, 2003
[5] Himpunan Mahasiswa Ilmu Informasi dan Perpustakaan, "*Model Vektor dan Clustering*",http://himaforsta.org/index.php?option=com_content&task=view&id=3&Itemid=1
[6] Jogiyanto H.M. "*Konsep Dasar Pemrograman Bahasa C".* Yogyakarta : Andi Offset, 2001
[7] M. Erwin AH dan Wahyudi, "*Customer information gathering menggunakan metode temu kembali informasi dengan model ruang vektor*", *Proceeding* SNATI 2005 ISBN: 979-756-061-6, T. Informatika UII. 2005
[8] Mandala, Rila dan Hendra Setiawan, "*Peningkatan Performansi Sistem Temu-Kembali Informasi dengan Perluasan Query Secara Otomatis*" Laboratorium Keahlian Informatika Teori Departemen Sistem Informasi, Institut Teknologi Bandung, 2006

[9] Mandala Rila, Takenobu Takunaga, Hozumi Tanaka. *"Query expansion using heterogenous thesauri. "* Proceeding of Information Processing and Management. 1999.
[10] Mandala Rila, Takenobu Takunaga, Hozumi Tanaka. "*The exploration and Analysis of Using Multiple Thesaurus types for Query Expansion in Information Retrieval".* Journal of Information Processing. 2000.
[11] Mandala Rila, "*Sistem Temu-kembali informasi dengan menggunakan model probabilistik"* Jurnal Informatika, ITB, Bandung, 2002.
[12] Miyamoto, Sadaki; "*Fuzzy sets in Information Retrieval and cluster analysis"* Kluwer Academic Publisher; London, 1990.
[13] Siregar, Ridwan A, "*Perpustakaan Sebagai Informasi Penelitian*", Program Studi Perpustakaan dan Informasi, USU, 1858-1447, Vol 2 No. 1, 1998.