Algoritma Umum Pencarian Informasi Dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen

Hendra Bunyamin
Jurusan Teknik Informatika
Fakultas Teknologi Informasi Universitas Kristen Maranatha
Jl. Prof. drg. Suria Sumantri No. 65, Bandung 40164
E-mail: hendra.bunyamin@eng.maranatha.edu

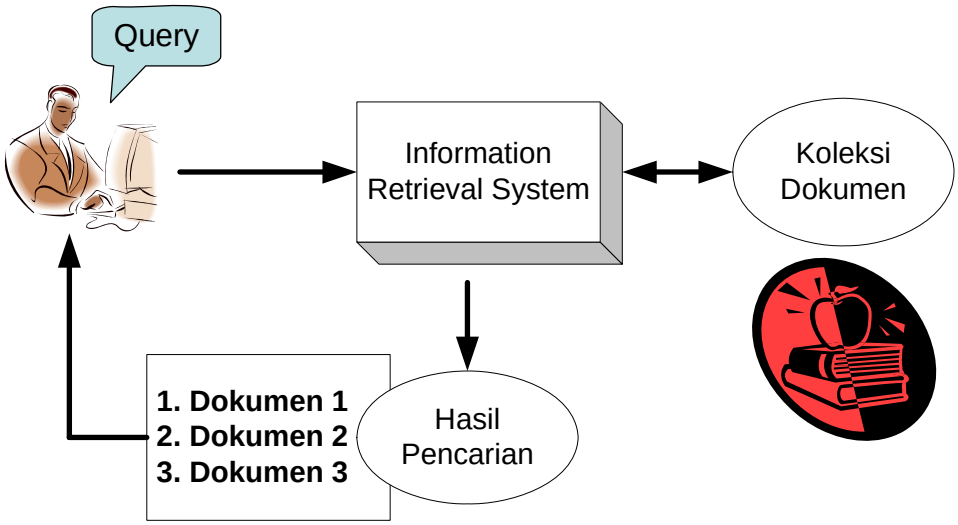
Abstract

Information retrieval (IR) system is a system, which is used to search and retrieve information relevant to the users' needs. IR system retrieves and displays documents that are relevant to the users' input (query). The information retrieval system has several steps and must execute the steps in order to obtain query results. The steps consist of two processes. The first one is processing query and the second one is processing the document collection. Processing query includes: conduct text operation, query formulation, and make terms index for query. Processing the document collection includes: conduct text operation, indexing, and make collection index for document collection. Obtaining terms index and collection index, we are able to process terms index and collection index to obtain ranking results. To obtain ranking results requires knowledge from basic linear algebra. This paper also explores how to make ranking from the most relevant documents to the most irrelevant documents

Keywords*: information retrieval system, non-interpolated average precision*

1. Pendahuluan

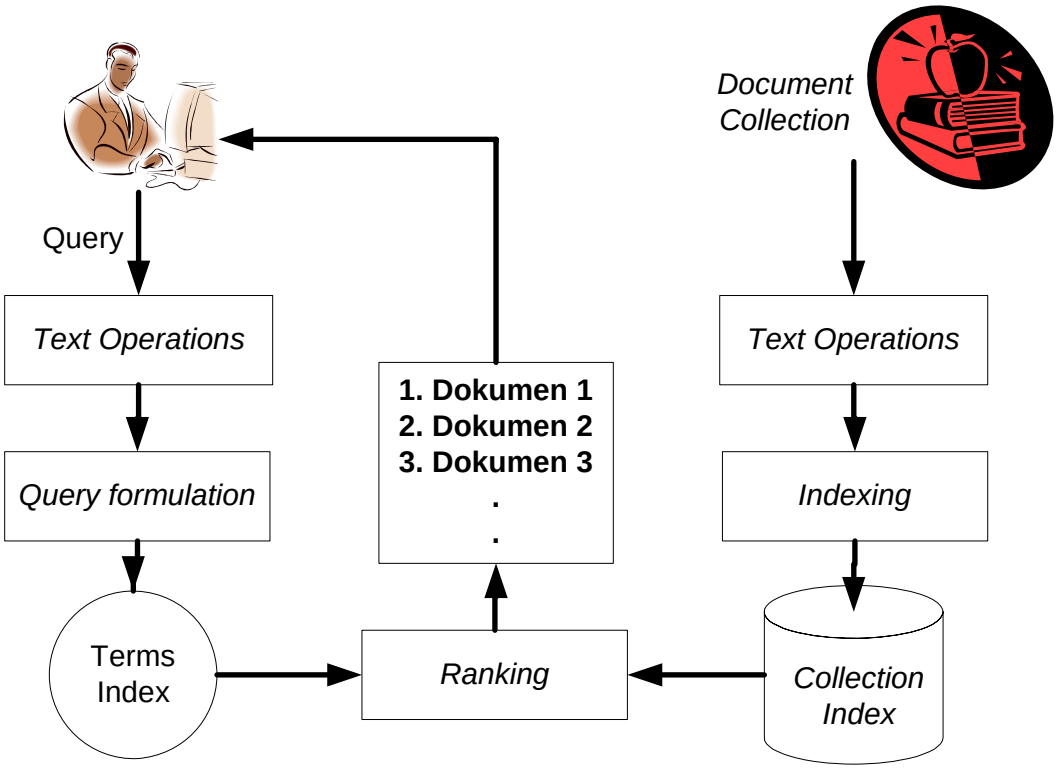
Information retrieval (IR) *system* digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.



Gambar 10 Ilustrasi *information retrieval system*

Salah satu aplikasi umum dari IR *system* adalah *search engine* atau mesin pencarian yang terdapat pada jaringan internet. Pengguna dapat mencari halaman-halaman web yang dibutuhkannya melalui *search engine*. Contoh lain dari IR *system* adalah sistem informasi perpustakaan.

IR *system* terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan IR *system* dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut.
Sebagai suatu sistem, IR *system* memiliki beberapa bagian yang membangun sistem secara keseluruhan. Bagian-bagian yang terdapat pada IR *system* digambarkan pada Gambar 1



Gambar 11 Bagian-bagian *information retrieval system*

Gambar 2 memperlihatkan bahwa terdapat dua buah alur operasi pada IR *system*. Alur pertama dimulai dari koleksi dokumen dan alur kedua dimulai dari *query* pengguna. Alur pertama yaitu pemrosesan terhadap koleksi dokumen menjadi basis data indeks tidak tergantung pada alur kedua. Sedangkan alur kedua tergantung dari keberadaan basis data indeks yang dihasilkan pada alur pertama.

Bagian-bagian dari IR system menurut gambar 2 meliputi:

1. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam query maupun dokumen (*term selection*) dalam pentransformasian dokumen atau query menjadi *term index* (indeks dari kata-kata).
2. *Query formulation* (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata *query*.
3. *Ranking* (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
4. *Indexing* (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

IR *system* menerima query dari pengguna, kemudian melakukan perangkingan terhadap dokumen pada koleksi berdasarkan kesesuaiannya dengan *query*. Hasil perangkingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem relevan dengan *query*. Namun relevansi dokumen terhadap suatu *query* merupakan penilaian pengguna yang subjektif dan dipengaruhi banyak faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna.

Model IR *system* menentukan detil IR *system* yaitu meliputi representasi dokumen maupun *query*, fungsi pencarian (*retrieval function*) dan notasi kesesuaian (*relevance notation*) dokumen terhadap *query*.

Terdapat beberapa model IR *system* seperti model *boolean* dan model ruang vektor. Dalam tulisan ini, model ruang vektor dipilih karena model ruang vektor mampu menghasilkan dokumen-dokumen terurut berdasarkan kesesuaian dengan *query*. Dan juga query di dalam model ruang vektor dapat berupa sekumpulan kata-kata dari pengguna dalam ekspresi bebas.

2. Model Ruang Vektor

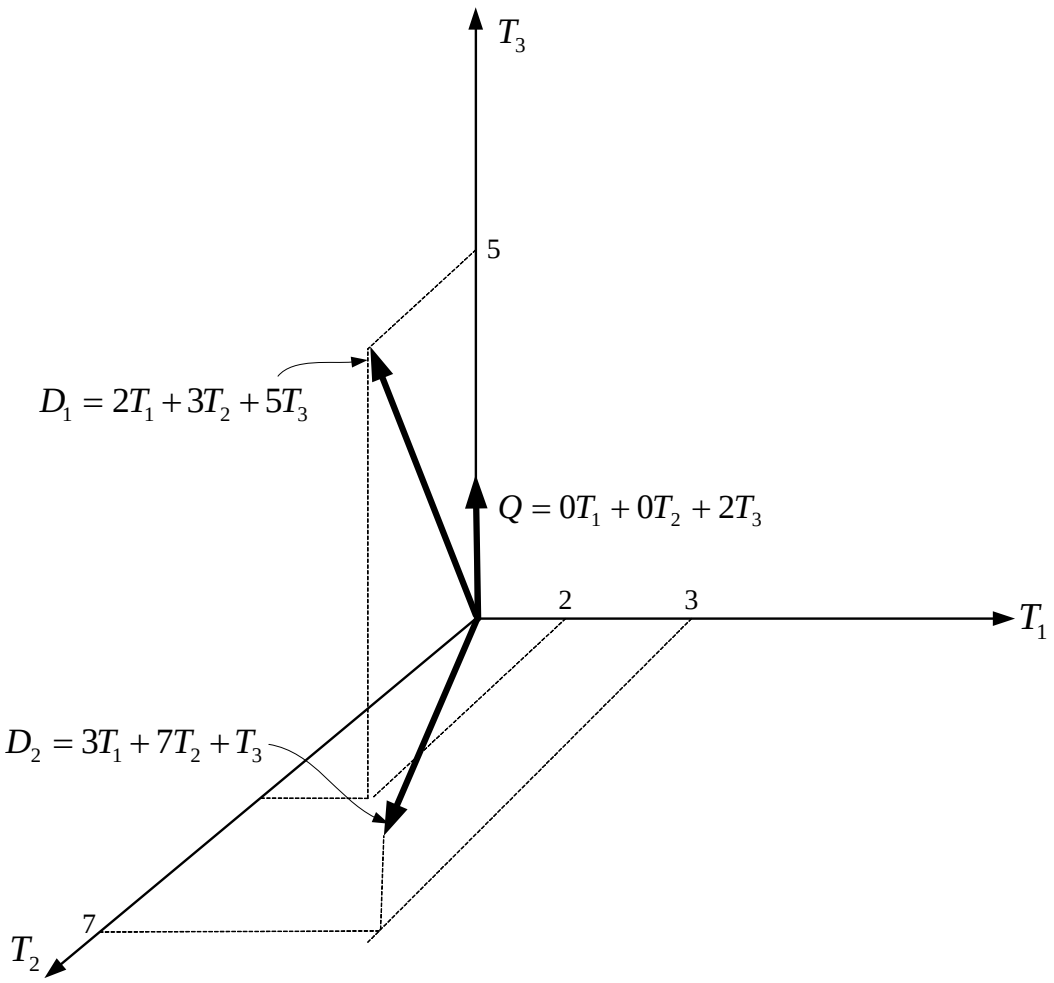
Misalkan terdapat sejumlah n kata yang berbeda sebagai kamus kata (*vocabulary*) atau indeks kata (*terms index*). Kata-kata ini akan membentuk ruang vektor yang memiliki dimensi sebesar n. Setiap kata i dalam dokumen atau *query* diberikan bobot sebesar w_i. Baik dokumen maupun *query* direpresentasikan sebagai vektor berdimensi n.

Sebagai contoh terdapat 3 buah kata (T_1, T_2 dan T_3), 2 buah dokumen (D_1 dan D_2) serta sebuah *query* Q. Masing-masing bernilai:

$$D_1 = 2T_1 + 3T_2 + 5T_3;\ D_2 = 3T_1 + 7T_2 + 0T_3;\ Q = 0T_1 + 0T_2 + 2T_3$$

Maka representasi grafis dari ketiga vektor ini adalah seperti pada gambar 3

Koleksi dokumen direpresentasi pula dalam ruang vektor sebagai matriks kata-dokumen (*terms-documents matrix*). Nilai dari elemen matriks w_{ij} adalah bobot kata i dalam dokumen j.



Gambar 12 Contoh vektor-vektor D_1, D_2, D_3 dan Q

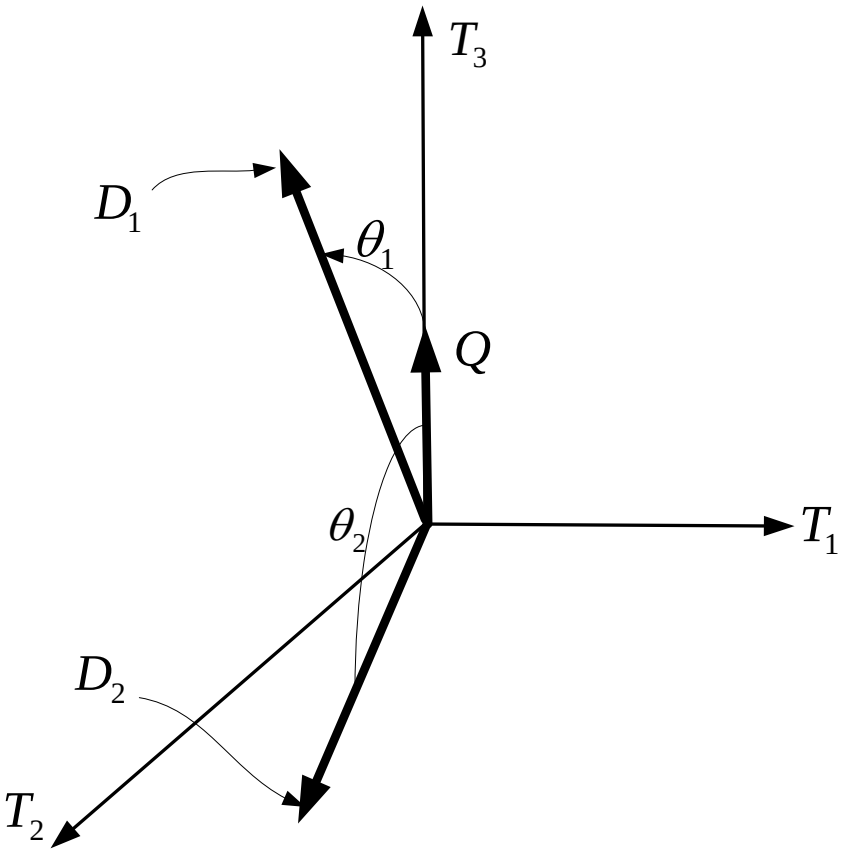
Misalkan terdapat sekumpulan kata T sejumlah m, yaitu $T = (T_1, T_2, \text{K}, T_m)$ dan sekumpulan dokumen D sejumlah n, yaitu $D = (D_1, D_2, \text{K}, D_n)$ serta w_{ij} adalah bobot kata i pada dokumen j. Maka gambar 4 adalah representasi matriks kata-dokumen

$$\begin{array}{c} \\ T_1 \\ T_2 \\ \text{M} \\ T_m \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \Lambda & D_n \end{array} \\ \begin{bmatrix} w_{11} & w_{12} & \Lambda & w_{1n} \\ w_{21} & w_{22} & \Lambda & w_{2n} \\ \text{M} & \text{M} & \text{M} & \text{M} \\ w_{m1} & w_{m2} & \Lambda & w_{mn} \end{bmatrix} \end{array}$$

Gambar 13 Representasi matriks kata-dokumen

Penentuan relevansi dokumen dengan *query* dipandang sebagai pengukuran kesamaan (*similarity measure*) antara vektor dokumen dengan vektor *query*. Semakin "sama" suatu

vektor dokumen dengan vektor *query* maka dokumen dapat dipandang semakin relevan dengan *query*. Salah satu pengukuran kesesuaian yang baik adalah dengan memperhatikan perbedaan arah (*direction difference*) dari kedua vektor tersebut. Perbedaan arah kedua vektor dalam geometri dapat dianggap sebagai sudut yang terbentuk oleh kedua vektor. Gambar 5 mengilustrasikan kesamaan antara dokumen D_1 dan D_2 dengan *query* Q. Sudut θ_1 menggambarkan kesamaan dokumen D_1 dengan *query* sedangkan sudut θ_2 menggambarkan kesamaan dokumen D_2 dengan *query*.



Gambar 5 Representasi grafis sudut vektor dokumen dan *query*

Jika Q adalah vektor *query* dan D adalah vektor dokumen, yang merupakan dua buah vektor dalam ruang berdimensi-n, dan θ adalah sudut yang dibentuk oleh kedua vektor tersebut. Maka

$$Q \bullet D = |Q|\,|D| \cos\theta \quad \text{.....(1)}$$

dengan $Q \bullet D$ adalah hasil perkalian titik (*dot product*) kedua vektor, sedangkan

$$|D| = \sqrt{\sum_{i=1}^{n} D_i^2} \text{ dan } |Q| = \sqrt{\sum_{i=1}^{n} Q_i^2} \quad \text{.....(2)}$$

merupakan *norm* atau panjang vektor di dalam ruang berdimensi-n.
Perhitungan kesamaan (*Similarity*) kedua vektor adalah sebagai berikut

$$Sim(Q, D) = \cos(Q, D) = \frac{Q \bullet D}{|Q||D|} = \frac{1}{|Q||D|} \sum_{i=1}^{n} Q_i \times D_i \quad \text{.....(3)}$$

dengan $Q_i \times D_i$ adalah perkalian antara Q_i dan D_i.

Metode pengukuran kesesuaian ini memiliki beberapa keuntungan, yaitu adanya normalisasi terhadap panjang dokumen. Hal ini memperkecil pengaruh panjang dokumen. Panjang kedua vektor digunakan sebagai faktor normalisasi. Hal ini diperlukan karena dokumen yang panjang cenderung mendapatkan nilai yang besar dibandingkan dengan dokumen yang lebih pendek.

Proses perangkingan dari dokumen dapat dianggap sebagai proses pemilihan (vektor) dokumen yang dekat dengan (vektor) *query*, kedekatan ini diindikasikan dengan sudut yang dibentuk. Nilai cosinus yang cenderung besar mengindikasikan bahwa dokumen cenderung sesuai *query*. Nilai cosinus sama dengan 1 mengindikasikan bahwa dokumen sesuai dengan *query*.

3. Pembobotan Kata

Bagian sebelumnya membahas mengenai metode pengukuran kesesuaian antara dokumen dan *query* dalam model ruang vektor. Dokumen maupun *query* direpresentasikan sebagai vektor berdimensi-n. Bagian ini akan membahas mengenai nilai dari vektor atau bobot kata dalam dokumen.

Salah satu cara untuk memberi bobot terhadap suatu kata adalah memberikan nilai jumlah kemunculan suatu kata (*term frequency*) sebagai bobot. Semakin besar kemunculan suatu kata dalam dokumen akan memberikan nilai kesesuaian yang semakin besar.

Faktor lain yang diperhatikan dalam pemberian bobot adalah kejarangmunculan kata (*term scarcity*) dalam koleksi. Kata yang muncul pada sedikit dokumen harus dipandang sebagai kata yang lebih penting (*uncommon terms*) daripada kata yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*inverse document frequency*). Hal ini merupakan usulan dari George Zipf. Zipf mengamati bahwa frekuensi dari sesuatu cenderung kebalikan secara proporsional dengan urutannya.

Faktor terakhirnya adalah faktor normalisasi terhadap panjang dokumen. Dokumen dalam koleksi dokumen memiliki karakteristik panjang yang beragam. Ketimpangan terjadi karena dokumen yang panjang akan cenderung mempunyai frekuensi kemunculan kata yang besar. Sehingga untuk mengurangi ketimpangan tersebut diperlukan faktor normalisasi dalam pembobotan.

Perbedaan antara normalisasi pada pembobotan dan perangkingan adalah normalisasi pada pembobotan dilakukan terhadap suatu kata dalam suatu dokumen sedangkan pada perangkingan dilakukan terhadap suatu dokumen dalam koleksi dokumen.

Pembobotan yang dianggap paling baik adalah menggunakan persamaan

$$w_i = \frac{\log(tf_i) + 1.0}{\sqrt{\sum_{j=1}^{t} [\log(tf_j) + 1.0]^2}} \quad \text{.....(4)}$$

untuk pembobotan kata i (w_i) pada dokumen dan menggunakan persamaan

$$q_i = \frac{(\log(tf_i)+1.0)+\log(N/n_i)}{\sqrt{\sum_{j=1}^{t}[(\log(tf_j)+1.0)\times(\log(N/n_j))]^2}} \quad \text{.....(5)}$$

untuk pembobotan kata i (q_i) pada *query*. Dengan tf_i adalah frekuensi kemunculan kata i, n_i banyak dokumen yang mengandung kata i dan N jumlah dokumen dalam koleksi.

4. Kesimpulan

Pengguna menggunakan IR *system* sebagai alat bantu untuk dapat mencari dokumen yang sesuai dengan *query* pengguna. Di dalam IR *system*, terdapat beberapa proses yang harus dilakukan sehingga IR system dapat menampilkan daftar *ranking* dokumen dari dokumen yang paling relevan dengan *query* sampai dengan dokumen yang tidak relevan dengan *query*.

Model IR *system* yang digunakan dalam tulisan ini adalah model ruang vektor. Di dalam model ruang vektor, *query* dan dokumen direpresentasikan sebagai vektor-vektor. Kesesuaian vektor query dengan vektor-vektor dokumen dihitung dengan menggunakan aljabar linier sederhana.

Daftar Pustaka

[Jac90] *Jacob, Bill (1990), Linear Algebra, W.H. Freeman and Company.*

[Kar98] *Karlgren, Jussi (1998), The Basics of Information Retrieval.*
URL: http://citeseer.nj.nec.com/146825.html

[Lid01] *Liddy, Elizabeth (2001), How a search engine works*
URL: http://www.infotoday.com/searcher/may01/liddy.htm

[Rij79] *Rijsbergen, C.J. van (1979), Information Retrieval, Butterworths, London.*

[Set02] *Setiawan, Hendra (2002), Umpan Balik Relevansi pada Sistem Temu Kembali Informasi, Tugas Akhir Departemen Teknik Informatika ITB.*