PENINGKATAN RELEVANSI HASIL PENCARIAN KATA KUNCI DENGAN PENERAPAN MODEL RUANG VEKTOR PADA SISTEM INFORMASI RUANG BACA DI JURUSAN ILMU KOMPUTER UNIVERSITAS UDAYANA

Ngurah Agus Sanjaya ER[a], Agus Muliantara[b], I Made Widiartha[c]
Program Studi Teknik informatika, Jurusan Ilmu Komputer,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
agus.sanjaya@cs.unud.ac.id[a], muliantara@cs.unud.ac.id[b], imadewidiartha@cs.unud.ac.id[c]

ABSTRAK

Sistem temu kembali informasi dapat memecahkan permasalahan pencarian informasi dengan cara tradisional yang ruang pencariannya terbatas pada judul, pengarang ataupun penerbit dari suatu dokumen. Dengan menggunakan *query* yang sesuai ruang pencarian pada sistem temu kembali informasi menjadi tidak terbatas.

Pada penelitian ini dikembangkan suatu sistem temu kembali informasi pada ruang baca Jurusan Ilmu Komputer, Universitas Udayana. Dokumen yang digunakan adalah berupa kumpulan abstrak dari tugas akhir mahasiswa. Proses pencarian *term* dimulai dengan melakukan tokenization, stop words removal dan stemming pada kumpulan dokumen. Kesamaan antara *query* masukan dengan dokumen dihitung menggunakan cosine similarity pada vektor *query* dan dokumen. Hasil pencarian berupa dokumen yang memiliki relevansi terhadap *query* dan diurut berdasarkan nilai cosine similarity yang menurun. Evaluasi terhadap sistem diukur dengan menghitung mean average precision dari uji coba.

Dari hasil pengujian yang dilakukan didapatkan kesimpulan bahwa sistem temu kembali informasi yang diterapkan pada ruang baca Jurusan Ilmu Komputer Universitas Udayana memberikan tingkat relevansi yang tinggi yang ditunjukkan oleh nilai mean average precision sebesar 70,84%.
Kata kunci*: sistem temu kembali informasi, cosine similarity, mean average precision*

ABSTRACT

Information retrieval can be applied to solve the problem of traditional searching where the search space is limited to title, author or publisher of a document. By using the appropriate query, the search space of an information retrieval system becomes unlimited.

In this research, an information retrieval system is developed for the reading room in Computer Science Department at Udayana University. Collection of documents used is the abstract of students thesis. Terms for dictionary are found by applying tokenization, stop words removal and stemming process on documents. The similarity between input query and a document is calculated using the cosine similarity its respective vectors. Search results for the user is given in the form of ranked documents in relevance to the query and sorted based on the value of cosine similarity. Evaluation of the system is measured by calculating the mean average precision of test results.

From the test results it can be concluded that the information retrieval system that is applied to the reading room of Computer Science Department, Udayana University provides a high level of relevance documents indicated by the mean average precision of 70.84%.

Keywords: information retrieval system, cosine similarity, mean average precision

1. PENDAHULUAN

Ruang baca suatu jurusan dalam universitas merupakan tempat dimana mahasiswa dapat mencari segala informasi untuk menunjang proses kegiatan belajar mengajarnya. Untuk dapat menemukan informasi yang dicari maka mahasiswa harus melakukan pencarian pada tumpukan dokumen tersebut dengan cara membacanya satu persatu. Tentu saja selain menghabiskan banyak tenaga, cara konvensional ini juga memerlukan banyak waktu. Informasi yang didapat pun belum tentu sesuai dengan yang diinginkan.

Permasalahan pencarian ini dapat diselesaikan dengan menerapkan sistem temu kembali informasi (*information retrieval system*) yang memungkinkan seorang pengguna, untuk mencari informasi tanpa dikenai batasan apa saja informasi yang dapat dicarinya. Informasi yang dicari disini harus relevan dengan kebutuhan pengguna dan proses pencariannya sendiri dilakukan secara otomatis. Dengan demikian sistem temu kembali informasi mendukung kebebasan pengguna dalam berekspresi melalui *query* untuk memenuhi kebutuhan informasinya. Disamping menyediakan pencarian informasi, sistem temu kembali informasi juga menangani permasalahan repre¬sentasi, penyimpanan dan organisasi dari informasi tersebut (Baeza-Yates, 2011).

Informasi yang dikembalikan oleh sistem temu kembali informasi berupa kumpulan dokumen yang relevan terhadap *query* yang dimasukkan oleh pengguna. Dokumen merupakan unit dimana sistem temu kembali informasi tersebut dibangun (Manning, 2008). Dokumen bisa berupa kalimat dalam satu paragraf, paragraf dalam satu halaman, halaman yang membentuk satu bab atau kumpulan bab yang membentuk suatu buku. Jadi pengertian dokumen disini tidak dapat diartikan secara harfiah menjadi buku, namun dapat berupa unit-unit yang lebih kecil dari buku.

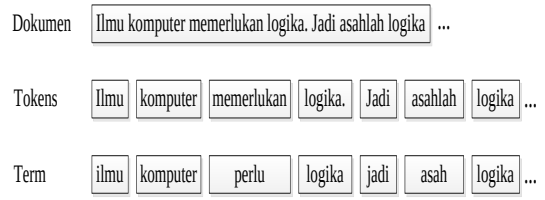
Sebelum dapat memberikan kembalian berupa kumpulan dokumen yang relevan maka sistem temu kembali informasi harus membangun suatu inverted index. Inverted index terdiri atas dua bagian yaitu kamus (*dictionary*) yang menyimpan kumpulan kata dan posting yang menyimpan informasi dokumen mana saja yang mengandung kata tersebut. Kembalian yang diberikan ke pengguna berupa kumpulan dokumen yang diurut berdasarkan peringkat relevansinya terhadap *query* masukan. Proses pemberian peringkat ini didasarkan pada konsep kesamaan (*similarity*) antara dokumen dengan *query*.

Pada penelitian ini untuk dapat mencari kesamaan antara dokumen dan *query* maka telah digunakan suatu model yaitu ruang vektor. Dokumen dan *query* masing-masing diwakili oleh suatu vektor dalam model ruang vektor ini. Besaran dari vektor-vektor tersebut merupakan nilai tf-idf (*term frequency-inverse document frequency*) dari dokumen dan *query*. Kesamaan antara vektor dokumen dengan *query* dihitung dengan fungsi kesamaan cosine. Dengan menggunakan fungsi kesamaan cosine, vektor dokumen dan *query* yang membentuk sudut 0^0 atau memiliki nilai *cosine* 1 akan memiliki kesamaan maksimum. Semakin besar sudut yang dibentuk antara vektor dokumen dan *query* maka semakin tidak relevan dokumen dan *query* tersebut. Dokumen yang mendapat peringkat tertinggi merupakan dokumen yang vektornya membentuk sudut terkecil dengan vektor *query* atau memiliki nilai cosine terbesar.

2. SISTEM TEMU KEMBALI INFORMASI

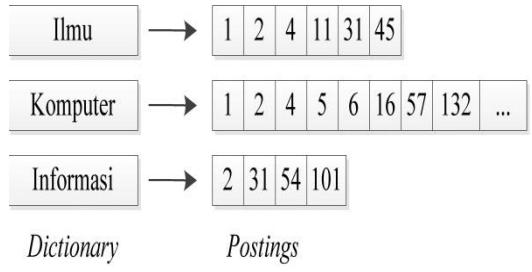
Temu kembali informasi dapat diartikan sebagai pencarian material koleksi yang besar yang tersimpan dalam komputer (Manning, 2008). Elemen penting dari sistem temu kembali informasi adalah *inverted index* yang menyimpan informasi seluruh kata yang ada dalam kumpulan dokumen yang digunakan untuk membangun sistem temu kembali informasi tersebut. Kumpulan seluruh kata tersebut sebagai kamus (*dictionary*). Kata yang dimasukkan ke dalam kamus didapatkan melalui proses tokenization yang dilakukan pada kumpulan dokumen. Tokenization menerima masukan berupa karakter¬karakter berurutan (*sequence*) kemudian memecahnya menjadi bagian-bagian yang lebih kecil (*token*) dengan menghilangkan karakter spesial seperti tanda baca. Token yang merupakan kumpulan karakter yang memiliki makna secara semantik umumnya dianggap sama dengan kata (*term*) walaupun sebenarnya tidak selalu memiliki

arti. Proses *tokenization* dapat dilihat pada Gambar 1.



Gambar 1. Proses *Tokenization*

Kata-kata yang didapat kemudian melalui proses stemming yaitu penghapusan imbuhan (awalan, akhiran serta awalan+akhiran) sehingga didapatkan kata dasarnya. Disamping penghapusan imbuhan, kata-kata tersebut juga dibandingkan dengan daftar kata yang dianggap tidak penting dan sering muncul (*stop words*) seperti ke, yang, dan, dengan dan lain sebagainya. Kata-kata pada dokumen yang diproses yang *term*asuk ke dalam daftar *stop words* tidak akan dimasukkan ke dalam kamus. Kata dasar yang berhasil melalui proses stemming dan stop words inilah yang dimasukkan ke dalam kamus. Kamus juga menyimpan total jumlah dokumen dimana masing-masing kata muncul. Sedangkan informasi dokumen mana saja yang mengandung suatu kata pada kamus disimpan pada *posting list* (Gambar 2). *Posting list* merupakan suatu struktur data berupa senarai (*linked list*) yang menyimpan dokumen ID dimana suatu kata pada kamus tersebut muncul. Dokumen ID disini merupakan suatu penanda unik yang biasanya berupa angka.



Gambar 2. *Dictionary* dan *Posting* Pembentuk *Inverted Index*

Jika kumpulan dokumen yang digunakan untuk membangun sistem temu kembali informasi ini berjumlah 1000 dokumen maka dokumen ID yang digunakan bisa dimulai dari 1 sampai dengan 1000. Satu posting dalam posting list berisikan informasi dokumen ID serta satu pointer yang menuju ke posting berikutnya. Dengan adanya kamus dan posting list ini maka data awal yang dibutuhkan untuk membangun sistem temu kembali informasi telah tersedia. Permasalahan berikutnya yang muncul adalah bagaimana cara memberikan nilai kesamaan untuk suatu dokumen terhadap *query* yang diberikan pengguna. Konsep yang digunakan dalam pembuatan peringkat ini adalah model ruang vektor.

3. MODEL RUANG VEKTOR

a. *Term Frequency* (tf)

Term frequency (tft,d) dari suatu *term* (t) pada dokumen (d) didefinisikan sebagai jumlah kemunculan dari t pada d. Untuk dapat menggunakan nilai tft,d dalam proses pembuatan peringkat maka harus diperhatikan beberapa hal yaitu: suatu dokumen A dimana t muncul sebanyak 10 kali adalah lebih relevan terhadap *query* pengguna dibandingkan dokumen B dimana t muncul hanya sekali. Walaupun demikian tidak berarti bahwa dokumen A 10 kali lebih relevan dibandingkan dokumen B terhadap *query* tersebut. Dengan pertimbangan tersebut maka pada proses pemberian peringkat biasanya digunakan *log frequency weight* dari t pada d yang dihitung sebagai berikut:

$$w_{t,d} = \begin{cases} 1 + log_{10} tf_{t,d}, & jika\ tf_{t,d} > 0 \\ 0, & sebaliknya \end{cases}$$

Dengan menggunakan *log frequency weight* maka jika suatu *term* muncul 1 kali maka nilai wt,d adalah 1. Jika muncul 2 kali maka wt,d menjadi 1,3; 10 kali nilainya menjadi 2 dan 1000 kali menjadi 4. Dari nilai tersebut dapat dilihat bahwa relevansi tidak bertambah secara proporsional sesuai dengan jumlah kemunculan suatu *term* t. Dengan menggunakan nilai *log frequency weight* ini maka nilai dari suatu pasangan dokumen dan *query* dihitung dengan rumus di bawah ini:

$$Nilai\ Skor = \sum_{t \in q \cap D} (1 + log_{10} tf_{t,d})$$

b. *Document Frequency* dan *Inverse Document Frequency* (idf)

Penggunaan *log frequency weight* untuk menentukan nilai dari suatu pasangan dokumen dan *query* memunculkan suatu permasalahan baru. Suatu *term* yang sering muncul dalam dokumen d memiliki nilai yang lebih besar dibandingkan dengan *term* lain. Namun untuk *query* yang mengandung kata yang jarang ditemui maka walaupun kata tersebut hanya muncul sekali dalam satu dokumen seharusnya memiliki nilai relevansi yang besar dibandingkan kata yang sering muncul tadi. Pemberian nilai yang lebih tinggi untuk suatu kata yang jarang ditemui dibandingkan kata yang sering muncul maka digunakan konsep *document frequency*. dft adalah *document frequency* dari t yaitu banyaknya dokumen yang mengandung t. Nilai dft tidak lebih besar dari N yang merupakan jumlah dokumen yang digunakan untuk membangun sistem temu kembali informasi. *Inverse document frequency* (idf) dari t dihitung menggunakan rumus:

$$idf_t = log_{10}\left(\frac{N}{df_t}\right)$$(Manning, 2009)

Penggunaan rumus tersebut dimaksudkan untuk dapat menekan efek dari dft. Sebagai contoh, jika jumlah dokumen (N) yang digunakan adalah 10.000 dokumen, dan satu kata yaitu "Calpurnia" muncul sebanyak 1 kali dalam 1 dokumen maka nilai idf-nya menjadi:

$$idf_{calpurnia} = log_{10}\left(\frac{10000}{1}\right) = 4$$

Jika kata "yang" muncul sebanyak 1.000 kali maka nilai idf-nya menjadi 1. Dengan menggunakan idf ini maka kata yang jarang muncul akan memiliki nilai relevansi yang lebih besar terhada *query* pengguna dibandingkan kata yang sering muncul. Kombinasi antara tf dan idf inilah yang akan digunakan untuk memberikan nilai terhadap suatu pasangan dokumen dan *query*.

c. Pembobotan Menggunakan tf-idf

Nilai tf-idf dari suatu kata (*term*) pada dokumen d, merupakan hasil perkalian dari bobot tf dan bobot idf dari kata tersebut.

$$\text{tf-idft,d} = \log\left(1 + tf_{t,d}\right) * \log_{10}\left(\frac{N}{df_t}\right)$$

Jika kata yang terkandung dalam *query* adalah lebih dari satu kata maka nilai dari pasangan dokumen d dan *query* q adalah:

$$\text{Nilai (q,d)} = \sum_{t \in q \cap d} tf.idf_{t,d}$$

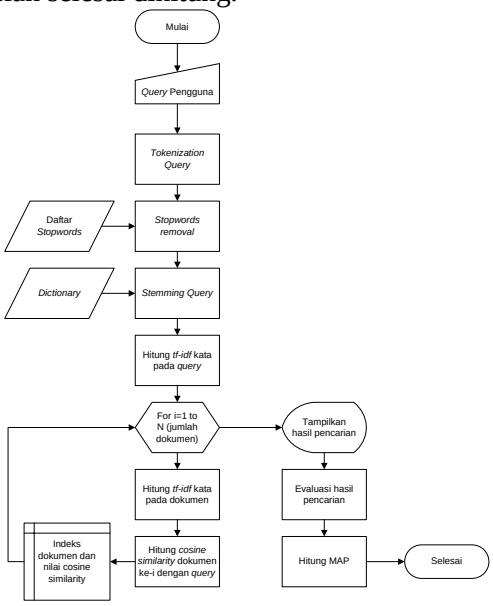
Dari rumus di atas dapat disimpulkan bahwa nilai suatu pasangan dokumen dan *query* merupakan penjumlahan dari tf-idf dari kata-kata yang muncul pada dokumen dan *query*. Misalnya *query* yang dimasukkan pengguna adalah "ilmu komputer" maka nilai suatu dokumen A terhadap *query* tersebut adalah penjumlahan dari tf-idfilmu dan tf-idfkomputer, jika kata "ilmu" dan "komputer" muncul pada dokumen A. Jika pada dokumen B kata "komputer" tidak muncul maka nilai dokumen B terhadap *query* adalah tf-idf ilmu saja. Dengan demikian, untuk *query* "ilmu komputer" kemungkinan besar dokumen A memiliki peringkat yang lebih baik dibandingkan dokumen B.

4. RANCANGAN SISTEM

a. *Flowchart*

Flowchart (diagram alir) dari sistem yang dikembangkan dapat dilihat pada Gambar 3. Proses dimulai ketika pengguna memasukkan *query* yang ingin dicari ke sistem. *Query* pengguna ini kemudian melalui proses tokenization untuk memecah masukan menjadi token. *Token-token* tersebut kemudian dihilangkan kata¬kata yang tidak penting yang terkandung di dalamnya (*stop words removal*). Sebelumnya kata-kata yang tidak penting tersebut harus disimpan terlebih dahulu pada basis data. Proses selanjutnya adalah pencarian kata dasar (*stemming*) dari masing-masing token. Proses stemming ini dibantu dengan dictionary yang berisikan kata-kata dasar dalam bahasa Indonesia.

Setelah *query* melalui proses penghilangan kata-kata tidak penting dan pencarian kata dasar maka selanjutnya bobot tf-idf untuk *query* tersebut dapat dihitung. Kemudian untuk semua dokumen yang terdapat pada sistem dihitung pula bobot tf-idf-nya terhadap *query* masukan pengguna. Disamping itu dihitung pula *cosine similarity* masing-masing dokumen terhadap *query* masukan yang nantinya digunakan sebagai dasar dalam pengurutan hasil pencarian. Proses berakhir ketika pengguna telah memberikan evaluasi terhadap hasil pencarian dan nilai MAP telah selesai dihitung.

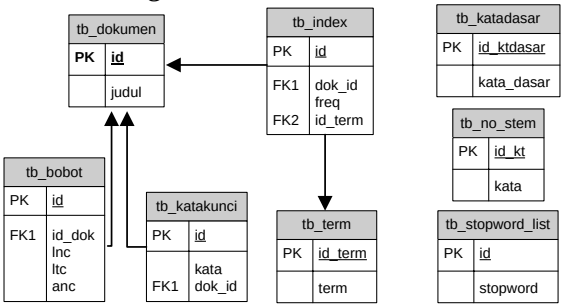


Gambar 3. *Flowchart* Sistem

b. Basis Data

Rancangan basis data yang digunakan dapat dilihat pada Gambar 4. Basis data yang dirancang terdiri atas 8 buah tabel sebagai berikut:

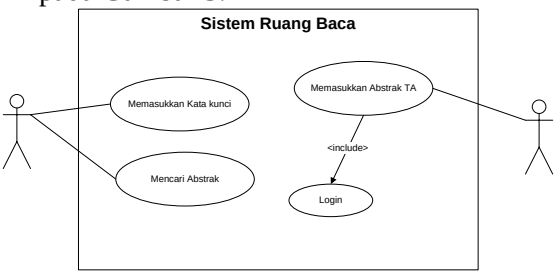
- Tb_dokumen berisi kumpulan abstrak yang merupakan objek penelitian.
- Tb_katadasar, berisi kumpulan kata dasar yang diunduh dari internet yang digunakan dalam proses *stemming*.
- Tb_no_stem, berisi kumpulan kata yang tidak perlu melalui proses *stemming*.
- Tb_stopword_list, berisi kumpulan kata yang umum yang sering digunakan tapi tidak mempengaruhi proses pencarian, seperti: di, oleh, ini, itu, dll. Dan data pada tb_stopword_list digunakan pada proses *stopword removal*.
- Tb_index, berisi hasil dari proses *preprocessing* yang meliputi *stopword removal* dan *stemming*.
- Tb_*term*, berisi kumpulan kata yang terkandung dari masing-masing artikel.
- Tb_bobot dan tb_katakunci merupakan suatu media penyimpanan untuk menyimpan data bobot total dan relevansi terhadap *query* pada masing-masing artikel.



Gambar 4. Basis Data

c. *Use Case*

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Adapun *use case diagram* dari sistem informasi ruang baca ini adalah seperti yang ditunjukkan pada Gambar 5.

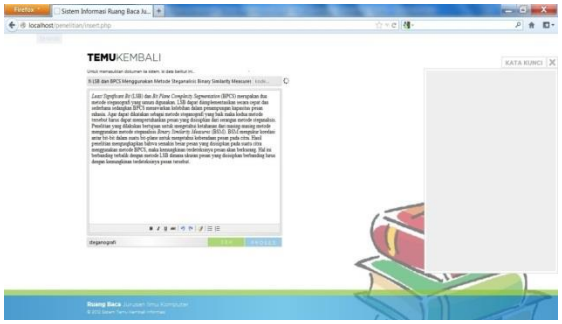


Gambar 5. *Use Case Diagram* Sistem Ruang Baca

5. HASIL

a. Antarmuka

Antarmuka (*user interface*) merupakan perantara antara sistem dengan pengguna sistem (user). Dalam sistem ruang baca ini antarmuka pengguna dirancang sedemikian rupa untuk kemudahan pengguna dalam penggunaan sistem ruang baca ini. Antarmuka dalam sistem ini seperti Gambar 6 dan Gambar 7.



Gambar 6. *User Interface* Halaman Muka

Gambar 7. User Interface

b. Pengujian Relevansi Hasil Pencarian

Tabel 1. Hasil Penilaian Relevansi *Query*

Dokumen	Query									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	R	R	R	R	R	R	R	R	R	R
2	R	R	R	R	R	R	R	R	R	R
3	R	R	R	R	R	R	TR	R	TR	R
4	R	R	R	R	TR	TR	R	R	R	R
5	TR	R	R	R	TR	R	R	R	TR	R
6	R	R	TR	R	R	R	R	TR	R	TR
7	TR	TR	R	R	R	R	R	TR	R	R
8	TR	R	R	R	R	TR	TR	R	R	R
9	R	R	R	TR	TR	TR	R	R	R	TR
10	R	R	TR	R	R	TR	R	R	R	R

Penghitungan average precision (AP) untuk Q1:

$$AP = \frac{\left(\frac{1}{1}\right)+\left(\frac{2}{2}\right)+\left(\frac{3}{3}\right)+\left(\frac{4}{4}\right)+\left(\frac{5}{6}\right)+\left(\frac{6}{9}\right)+\left(\frac{7}{10}\right)}{10} = \frac{6{,}2}{10} = 0{,}62$$

Penghitungan AP ini dilakukan untuk semua *query* (Q1 sampai dengan Q10). Hasil perhitungan nilai AP untuk masing-masing *query* dapat dilihat pada tabel 2.

Tabel 2. Nilai Average Precision untuk Tiap *Query* Masukan

doc	Query									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
AP	0,62	0,866	0,762	0,89	0,583	0,549	0,682	0,733	0,646	0,753

Dari tabel 2 di atas, nilai *mean average precision* (MAP) dapat dihitung sebagai berikut:

$$MAP = \frac{0{,}62+0{,}866+0{,}762+0{,}89+0{,}583+0{,}549+0{,}682+0{,}733+0{,}646+0{,}753}{10}$$

Persentase MAP dapat dihitung berdasarkan nilai MAP yaitu : 0,7084 * 100% = 70,84%

Pengujian relevansi dilakukan menggunakan sepuluh (10) buah *query* masukan yang meliputi "basis data", "algoritma", "implementasi", "analisis", "teknologi", "sistem", "informasi", "metode", "proses" dan "perbandingan". Relevansi dinilai terhadap sepuluh (10) buah dokumen hasil pencarian untuk masing-masing *query* masukan. Tabel 1 memperlihatkan hasil penilaian relevansi untuk masing-masing *query* masukan.

6. SIMPULAN

Dari hasil pengujian yang dilakukan didapatkan kesimpulan bahwa sistem temu kembali informasi yang diterapkan pada ruang baca Jurusan Ilmu Komputer Universitas Udayana memberikan tingkat relevansi yang tinggi yang ditunjukkan oleh nilai mean average precision sebesar 70,84%.

7. SARAN

Sistem temu kembali informasi yang dikembangkan pada penelitian ini tidak memperhatikan urutan kata baik pada *query* masukan pengguna maupun kemunculannya pada dokumen. Pada penelitian selanjutnya dapat dikembangkan sistem temu kembali informasi yang memperhatikan urutan kata untuk dibandingkan tingkat relevansinya dengan sistem yang telah dikembangkan pada penelitian ini.

8. DAFTAR PUSTAKA

Agusta, L. 2009. Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming

Dokumen Teks Bahasa Indonesia, Bali: Konferensi Nasional Sistem dan Informatika 2009.

Baeza-Yates, R., Ribeiro-Neto. 2011. Modern Information Retrieval: The Concepts and Technology Behind Search. 2nd Edition. Addison Wesley.

Croft, B., Metzler, D., Strohman, T. 2008. Search Engines : Information Retrieval in Practice. France, Addison Wesley.

Elmasri, R., Navathe, S. 2010. Fundamentals of Basis data Systems, 6th Edition, Pearson Education, Inc. Kendall, K. E., Kendall, J.E. 2010. 8th Systems Analysis and Design. Edition, Prentice Hall.

Liu, Yan-Tie., Xu, J., Qin, T., Xiong, W., Li, H. 2007. LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. SIGIR '07 The 30th Annual International SIGIR Conference, July 23-27, Amsterdam.

Manning, C.D., Raghavan, P., and Schutze, H. 2009. An Introduction to Information Retrieval. Cambridge University Press.

Pressman, R. S. 2009. Software Engineering: A Practitioner's Approach. 7th Edition, McGraw-Hill.