p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

SISTEM TEMU-KEMBALI INFORMASI DALAM DOKUMEN (PENCARIAN 10 KATA KUNCI DI EJOURNAL BSI)

Melisa Winda Pertiwi [1*] Taufiqurrochman [2]
[*12]Program Pascasarjana Magister Ilmu Komputer, Nusa Mandiri
Jl. Kramat Raya No. 18, Jakarta Pusat
[*]E-mail: melisa.wp@gmail.com

ABSTRAK

Dokumen tekstual sebagian besar telah tersedia secara digital atau elektronik dengan jumlah yang semakin banyak sejak penemuan komputer. Kebutuhan informasi membuat pengguna melakukan aktivitas pencarian pada koleksi dokumen yang dimilikinya. Metode pencarian dokumen yang tersedia di komputer biasanya hanya melihat ke dalam nama dokumen (file) yang dicari oleh pengguna tanpa memperhatikan isi isinya, oleh karena itu tidak dapat memenuhi kebutuhan informasinya. *Metode Latent Semantic Indexing* (LSI) bisa menjadi solusi untuk mengambil informasi yang relevan dari koleksi dokumen di dalam komputer. Metode LSI memiliki kemampuan untuk mengkorelasikan istilah terkait secara semantik yang laten dalam kumpulan teks menggunakan teknik *Singular Value Decomposition* (SVD). Sistem pencarian informasi menggunakan metode LSI telah dikembangkan dan diuji pada 10 kata kunci di website http://ejornal.bsi.ac.id/ejurnal. Pada proses pengujian yang telah dilakukan, dengan ambang batas 0,6 sistem dapat memberikan hasil pencarian dengan nilai presisi rata-rata 1,53% dan skor recall rata-rata sebesar 10,29% dalam proses pengujian.

Kata Kunci*: sistem temu balik, laten semantic indexing, singular value decompotion, precision, recall.*

ABSTRACT

Textual documents have largely been available digitally or electronically with increasing numbers since the invention of computers. Information needs to make users perform search activities on the collection of documents it has. Document search methods available on the computer usually only look into the name of the document (file) searched by the user regardless of the contents of the contents, therefore can not meet the information needs. The Latent Semantic Indexing Method (LSI) can be a solution to retrieve relevant information from a collection of documents inside a computer. The LSI method has the ability to correlate semantically related terms latently in a collection of texts using Singular Value Decomposition (SVD) techniques. Information retrieval system using LSI method has been developed and tested on 10 keywords in website http://ejornal.bsi.ac.id/ejurnal. In the testing process that has been done, with a threshold of 0.6 system can provide search results with an average precision value of 1.53% and an average recall score of 10.29% in the testing process.

Keywords: retrieval system, latent semantic indexing, singular value decompotion, precision, recall.

PENDAHULUAN

Pembuatan suatu penelitian makalah ilmiah sangat didukung dengan berbagai referensi makalah lainnya yang sudah terpublikasi dan memiliki nomor seri. Banyaknya makalah yang sudah dipublikasikan akan membuat pengguna mengalami kesulitan untuk memperoleh informasi yang dibutuhkan. Pengguna tidak dapat melihat isi makalah satu persatu untuk mendapatkan informasi yang tepat. Pengguna biasa menggunakan fasilitas pencarian kata kunci (*keyword*) yang disediakan oleh sistem operasi dengan memasukkan kata yang dicari. Kata kunci yang dimasukkan pengguna kemudian menjadi kueri untuk program pencarian dokumen/makalah pada komputer tersebut. Pencarian dokumen di dalam komputer biasanya hanya melihat pada nama dokumen yang dicari tanpa memperhatikan isi yang terkandung di dalamnya sehingga seringkali tidak dapat memenuhi kebutuhan informasi pengguna.

Fasilitas pencarian menggunakan pencocokan kata kunci dengan kata di dalam dokumen yang juga disediakan oleh sistem operasi masih belum dapat memberikan hasil pencarian yang

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

relevan. Metode pencocokan kata akan menghitung jumlah kata kunci yang muncul di dalam dokumen kemudian mengembalikan urutan dokumen dengan jumlah kemunculan terbanyak kepada pengguna. Karena pengguna bukan mencari dokumen yang mengandung banyak kata yang sama dengan kata kunci, metode pencocokan kata bukan merupakan solusi yang ideal untuk pencarian informasi. Selain itu, banyaknya kata yang memiliki kesamaan arti (sinonim) dan kata yang memiliki arti lebih dari satu (polisemi) dalam penggunaan kata kunci untuk mengekspresikan informasi yang dibutuhkan, akan membuat hasil pencarian dengan metode pencocokan kata semakin jauh dari relevan.

Metode *Latent Semantic Indexing* (LSI, kadang disebut dengan *Latent Semantic Analysis*) telah mencoba mengatasi masalah pencocokan teks dalam bidang temu-kembali informasi (*information retrieval*). Metode ini dikembangkan pada tahun 1988 dan telah dipatenkan (U.S. Patent no. 4,839,853) oleh Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum, dan Lynn Streeter. Metode ini dinamakan LSI karena memiliki kemampuan untuk menemukan hubungan tersembunyi (*latent*) antara semua *terms* (kata) yang memiliki kedekatan makna secara kontekstual Metode LSI memberikan solusi untuk masalah kata-kata sinonim dan polisemi yang sering terjadi dalam temu-kembali informasi.

Metode LSI ini akan digunakan untuk mencari 10 kata kunci pada makalah yang sudah dipublikasi pada alamat ***http://ejornal.bsi.ac.id/ejurnal***.

LANDASAN TEORI

Latent Semantic Indexing

Latent Semantic Indexing (LSI) adalah sebuah teknik berbasis bidang vektor (*vector space*) yang dapat menciptakan asosiasi antara dokumen-dokumen yang berhubungan secara konseptual. Pada model bidang vector. sekumpulan dokumen diindeks berdasarkan *terms* dan direpresentasikan dalam bentuk *m x n term-document matrix A* yang dinotasikan sebagai,

$A = [aij]$ ……(1)

Elemen-elemen di dalam matriks A merupakan bobot frekuensi term i yang muncul pada dokumen j. Kolom-kolom pada matriks A merepresentasikan n vector dokumen dan baris-barisnya merepresentasikan m vektor terms. Fungsi temu-kembali membandingkan vektor kueri q dengan tiap kolom yang merepresentasikan dokumen tertentu dalam bidang vektor. Sebuah derajat kesamaan antara vektor kueri dan semua dokumen yang direpresentasikan dalam bidang vektor kemudian diukur untuk perangkingan. Contoh matriks *term-document* diperlihatkan pada gambar 1 :

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
eps	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Gambar 1 : contoh matriks *term-document*

LSI merupakan sebuah metode *automatic indexing* dan *retrieval* dengan memanfaatkan *semantic structure* (struktur asosiasi *terms* dengan dokumen) yang secara implisit terdapat dalam suatu dokumen untuk digunakan dalam pencarian dokumen yang relevan dengan *terms* dalam kueri. Metode LSI mengasumsikan bahwa terdapat sebuah latent *semantic structure*, yaitu sebuah struktur semantik yang tersembuyi (latent) dalam setiap dokumen, yang kabur atau tidak jelas karena keberagaman pemakaian kata dalam penulisan dokumen tersebut (dikenal dengan istilah *noise*). LSI menggunakan teknik-teknik statistik untuk mendapatkan *latent structure* dan menghilangkan noise yang ada. Sebuah penggambaran atau deskripsi dari terms dan dokumen-dokumen berdasarkan *latent semantic structure* tersebut digunakan untuk proses pengindeksan (indexing) dan pencarian kembali (retrieval).

Dekomposisi nilai singular (*singular value decom-position*) atau biasa dikenal dengan SVD digunakan dalam LSI untuk melakukan dekomposisi terhadap matriks *term-document* menjadi tiga buah matriks. SVD dari suatu matriks A didefinisikan sebagai,

$A = USV^T$ ..... (2)

Matriks U merupakan matriks yang berisi vektor eigen dari matriks AAT (matriks vector vector singular kiri), matriks V merupakan matriks yang berisi vektor eigen dari ATA

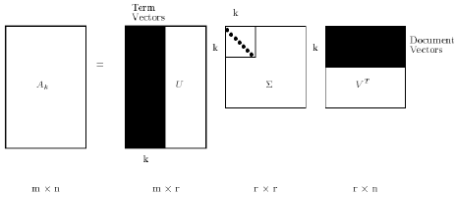
p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

(matriks vektor-vektor singular kanan), dan akar kuadrat dari nilai eigen AAT atau ATA mengisi nilai singular matriks S. Matriks U dan V dianggap sebagai matriks dari vector vector term dan dokumen secara berturut-turut.

Proses penghitungan SVD ini akan menghasilkan perkiraan *low-rank* k terbaik yang dapat mengurangi *noise* dan menyingkapkan struktur nyata dari data yang digunakan. Proses menentukan *low-rank* k terbaik (*rank lowering*) dilakukan dengan cara menyimpan sebanyak k nilai singular pertama (terbesar) dan sisanya (nilai-nilai singular yang lebih kecil) diubah menjadi nol. Dengan melakukan *rank-lowering*, dimensi dari matriks singular S akan tereduksi, begitu juga dengan matriks singular kiri U dan matriks singular kanan V, sehingga persamaan SVD dituliskan sebagai,

$A_k = U_k S_k V_k$ ..... (3)

U_k merupakan matriks *m x k* yang berisi k kolom pertama dari U, V_k merupakan matriks *n x k* yang berisi k kolom pertama dari V, dan S_k merupakan matriks diagonal *k x k* yang berisi k nilai singular terbesar [1, 7, 8]. Hasil dekomposisi matriks diperlihatkan pada gambar 2. Bagian dari matriks yang diarsir tebal adalah rank-k *approximation* model dari matriks A, dan k adalah jumlah nilai singular yang tidak diabaikan.



Gambar 2. Hasil Dekomposisi Matriks
Sumber : [5]

Low-rank (reduction) model ini kemudian digunakan sebagai *vector space model* dalam LSI untuk menentukan letak terms atau dokumen. Pada matriks V, untuk setiap n dokumen, matriks ini berisikan n baris yang yang merupakan vektor eigen. Maka setiap baris yang ada menyimpan koordinat dari tiap dokumen dalam model vektor.

Kueri (kata kunci) yang dimasukkan pengguna harus direpresentasikan dalam bentuk vektor untuk dibandingkan denganvektor dokumen dalam proses pencarian. Dalam model LSI, kueri yang dimasukkan dibentuk menjadi pseudo-documents yang akan menentukan lokasi kueri tersebut dalam bidang matriks term-document yang telah tereduksi [4, 9]. Vektor kueri q^ dari *pseudo document* q dapat dituliskan sebagai,

$q = q^T U_k S_k^{-1}$ ..... (4)

Vektor kueri yang dihasilkan kemudian dibandingkan dengan seluruh vektor dokumen dalam matriks VT hasil dari dekomposisi matriks term-document dengan sebuah ukuran kesamaan (similarity). Sebuah ukuran kesamaan yang umum digunakan adalah *cosine similarity* yang mengukur sudut antara vektor kueri dengan vektor dokumen. *Cosine similarity* antara duah buah vektor A dan B dapat dituliskan,

$$similarity = \cos\theta = \frac{A \bullet B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}}.$$

Dokumen-dokumen yang diperbandingkan dengan vektor kueri kemudian diurutkan dalam bentuk rangking berdasarkan nilai kesamaannya. Nilai kesamaan akan berada antara -1 hingga 1. Semakin nilai kesamaan mendekati 1 menunjukkan bahwa dokumen semakin cocok atau mirip dengan kueri.

3. Implementasi

Proses yang berjalan di dalam sistem ini dibagi menjadi proses pengindeksan dan proses pencarian seperti terlihat pada gambar 3 (gambaran umum sistem). Proses pengindeksan merupakan proses untuk membuat indeks dan representasi dokumen-dokumen di dalam komputer menjadi bentuk vektor-vektor dokumen. Indeks merupakan kumpulan kata – kata atau konsep yang telah dipilih dan digunakan sebagai petunjuk menuju informasi atau dokumen terkait.

Proses pengindeksan mencakup operasi teks, pembobotan, pembentukan matriks *term-document*, dan penghitungan SVD (*Singular Value Decompo-sition*). Operasi teks akan mengubah teks di dalam dokumen menjadi potongan-potongan kata dasar atau akar katanya (*stemming*) dalam huruf kecil dan menghapus kata sambung dan kata-kata lainnya yang tidak digunakan dalam temukembali informasi (dikenal dengan *stop words*).

Pembobotan merupakan metode yang umum dan efektif dalam meningkatkan performa pencarian kembali dalam model vektor yang dilakukan dengan mentransformasi frekuensi kasar kemunculan suatu term di dalam dokumen menggunakan suatu fungsi.

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Transformasi semacam itu memiliki dua komponen, global weighting dan *local weighting*. Setiap term diberikan sebuah *global weight* (bobot global) yang mengindikasikan tingkat pentingnya term tersebut di dalam keseluruhan koleksi dokumen, dan *local weight* (bobot lokal) untuk menunjukkan tingkat pentingnya term i yang muncul dalam dokumen j. Bobot dari matriks A dapat dituliskan sebagai,

$a_{ij} = L(i, j) \times G(i)$ ..... (6)

L(i, j) adalah fungsi bobot lokal untuk term I di dalam dokumen *j*, dan *G(i)* adalah fungsi obot global untuk term i. Bobot local dihitung dengan menggunakan *tf* (*term frequency*) dengan norma-lisasi yang didefinisikan sebagai,

$$L(i,j) = tf_{ij} = \frac{n_{ij}}{m_j} \quad \text{..... (7)}$$

n_{ij} merupakan jumlah kemunculan dari term *i* di dalam dokumen *j*, dan mj merupakan jumlah seluruh terms di dalam dokumen *j*. Sedangkan bobot global dihitung dengan menggunakan *idf* yang didefinisikan sebagai

$$G(i) = idf_i = \log\frac{N}{df_i} \quad \text{..... (8)}$$

N melambangkan jumlah total dari dokumen yang terdapat di dalam koleksi, dan dfi merupakan jumlah dokumen-dokumen di dalam koleksi yang mengandung term *i*.

Setelah bobot seluruh kata dihitung, matriks *term-document* dibentuk. Pembentukan matriks term-document digunakan untuk menyatakan hubungan antara terms dengan dokumen-dokumen dalam sistem temu kembali informasi. Sebagaimana terlihat pada gambar 1, kolom pada matriks *term document* mewakili dokumen, dan barisnya mewakili terms. Matriks yang dihasilkan kemudian digunakan dalam proses SVD. Akhir dari proses pengindeksan adalah sebuah vector dokumen yang digunakan dalam proses pencarian.

Proses pencarian dokumen yang relevan terhadap kueri yang dimasukkan pengguna berupa suatu kata kunci. Dalam proses pencarian, operasi teks yang sama seperti dalam proses pengindeksan juga dilakukan terhadap kata kunci yang dimasukkan. Sesudah operasi teks dilakukan, operasi kueri kemudian memproses kata kunci menjadi *pseudo-document* dan menghitung vektor kueri. Vektor kueri yang dihasilkan dari operasi kueri kemudian digunakan untuk menghitung nilai kesamaan (*cosine similarity*) dengan vektor dokumen yang dihasilkan dari proses pengindeksan. Proses penghitungan nilai kesamaan ini menghasilkan nilai kesamaan tiap dokumen terhadap kata kunci.

Setelah nilai kesamaan dokumen terhadap kata kunci dihasilkan, dilakukan pengurutan nilai tersebut dengan memilih dokumen yang nilainya telah melebihi batas ambang (*threshold*) tertentu kemudian menampilkannya kepada pengguna dalam bentuk rangking yang diurutkan dari nilai yang terbesar hingga terkecil (descending).

4. Pengujian

Pengujian untuk sebuah sistem temu kembali informasi dilakukan menggunakan *precision* dan *recall*. *Precision* mengukur kemampuan sistem untuk mengembalikan hanya dokumen-dokumen yang relevan, sedangkan *recall* mengukur kemampuan sistem untuk mengembalikan semua dokumen yang relevan. Nilai *precision* (*P*) didefinisikan sebagai,

$$P = \frac{D_r}{D_t} \quad \text{..... (9)}$$

Sedangkan *recall* (*R*) didefinisikan dengan,

$$R = \frac{D_r}{N_r} \quad \text{..... (10)}$$

Dr merupakan jumlah dari dokumen relevan yang diperoleh, Dt adalah jumlah total dari dokumen yang diperoleh, dan Nr adalah jumlah total dari dokumen yang relevan di dalam koleksi dokumen.

Pengujian dilakukan dengan makalah – makalah penelitian yang ada pada alamat ***http://ejornal.bsi.ac.id/ejurnal*** **(**diakses pada 25 November 2016**)** berbahasa Indonesia. Sebelum dilakukan uji pencarian, terlebih dahulu dilakukan proses pengindeksan dokumen-dokumen mulai dari operasi teks, pembobotan, SVD. Matriks tersebut kemudian digunakan untuk proses SVD dan penghitungan vector dokumen.

Himpunan batas ambang yang optimal untuk masing-masing kueri (10 kueri) diperoleh {0.9, 0.7, 0.9, 0.9, 0.8, 0.9, 0.9, 0.9, 0.6, 0.9}. Batas ambang yang optimal untuk kueri secara umum ditentukan dengan dua cara yaitu (1) dengan mengambil nilai batas ambang optimal yang paling rendah dari himpunan batas ambang optimal tiap kueri dan (2) dengan mengambil nilai rata-rata dari batas ambang optimal tiap kueri. Dengan cara pertama diperoleh batas ambang 0.6. Sedangkan dengan cara kedua, nilai batas ambang yang digunakan adalah 0.84. Tabel 1 memperlihatkan hasil uji pencarian

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

menggunakan kedua batas ambang optimal yang telah ditentukan (0.6 dan 0.84). Nilai rata-rata precision dan recall untuk kedua batas ambang dapat dilihat pada tabel 2.

Tabel 1. Hasil Uji Pencarian Menggunakan Batas Ambang 0.6 dan 0.84

No	Kueri	Batas Ambang	D_r	D_t	N_r	P	R
1	Manajemen informatika	0.6	5	53	5	0.094339623	1
		0.84	5	25	5	0.2	1
2	Basis Data	0.6	3	16	3	0.1875	1
		0.84	3	8	3	0.375	1
3	Kesehatan	0.6	1	47	1	0.021276596	1
		0.84	1	24	1	0.041666667	1
4	Algoritma	0.6	9	47	8	0.191489362	1.125
		0.84	9	24	8	0.375	1.125
5	Sistem Pakar	0.6	5	13	5	0.384615385	1
		0.84	5	6	5	0.833333333	1
6	Perancangan Web	0.6	1	27	1	0.037037037	1
		0.84	1	12	1	0.083333333	1
7	Metode Waterfall	0.6	1	23	1	0.043478261	1
Tabel 1. Hasil Uji Pencarian Menggunakan Batas Ambang 0.6 dan 0.84							
		0.84	1	12	1	0.083333333	1
8	Data Mining	0.6	1	19	1	0.052631579	1
		0.84	1	8	1	0.125	1
9	Jaringan Komputer	0.6	7	15	6	0.466666667	1.166666667
		0.84	7	6	6	1.166666667	1.166666667
10	Teknologi Informasi	0.6	4	76	4	0.052631579	1
		0.84	4	25	4	0.16	1

Keterangan :
Dr = Jumlah dokumen relevan yang diperoleh
Dt = Jumlah dokumen yang diperoleh
Nr = Jumlah dokumen relevan dalan koleksi dokumen
P = Precision
R = Recall

Tabel 2. Rata-rata Nilai *Precision* dan *Recall* untuk Batas Ambang 0.6 dan 0.84

No	Batas Ambang	Rata - rata Precision	Rata - rata Recall
1	0.6	0.153166609	1.029166667
2	0.84	0.344333333	0.929166667

ANALISIS HASIL PENGUJIAN

Hasil pengujian menunjukkan bahwa batas ambang yang optimal untuk suatu kueri belum tentu optimal untuk kueri yang lain. Oleh karena itu, dilakukan percobaan agar didapatkan nilai batas ambang yang optimal untuk kueri secara umum dengan dua cara yang berbeda. Setelah dilakukan percobaan, batas ambang 0.6 memiliki rata-rata nilai recall yang lebih tinggi untuk seluruh kueri (1,0291) dibandingkan dengan rata-rata nilai recall pada batas ambang 0.84 (0,9291). Hal tersebut menunjukkan bahwa dengan batas ambang 0.6 sistem temu-kembali informasi mampu mempero-leh 10.29% dokumen relevan dari seluruh dokumen relevan dalam kumpulan dokumen yang digunakan. Sedangkan dengan batas ambang 0.84 dokumen relevan yang dapat dikembalikan oleh sistem adalah sebesar 9.29%.

p- ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Sebaliknya, batas ambang 0.84 memiliki rata-rata nilai precision yang lebih tinggi (0.344) dibandingkan rata-rata nilai *precision* pada batas ambang 0.6 (0.153). Hal tersebut menunjukkan bahwa dengan batas ambang 0.84 kemampuan sistem dalam mengembalikan hanya dokumen yang relevan adalah sebesar 3.43% yang berarti perbandingan dokumen relevan yang diperoleh dengan seluruh dokumen yang diperoleh lebih tinggi atau ketat dibandingkan dengan batas ambang 0.6 sebesar 1.53%.

KESIMPULAN

Setelah dilakukan penelitian pada sistem temu-kembali informasi dalam dokumen menggunakan metode LSI, maka dapat disimpulkan beberapa hal sebagai berikut:

Berdasarkan pengujian yang dilakukan, sistem temu-kembali informasi menggunakan metode LSI dapat memberikan hasil pencarian yang relevan dengan nilai recall 10.29% dan precision 1.53% pada batas ambang 0.6.

Pada batas ambang 0.84 didapat nilai *recall* 9.29% dan *precision* 3.44%. ini membuktikan bahwa nilai yang lebih relevan berada pada batas ambang 0.6 untuk *recall* dan 0.84 untuk *precision*.

Saran

Beberapa hal yang dapat dijadikan saran untuk penelitian dan pengembangan sistem temu-kembali informasi lebih lanjut adalah ketika melakukan perhitungan pastikan data sudah sesuai, namun seiring perkembangan dan *update* setiap data maka metode LSI ini harus terus dilakukan percobaan supaya lebih relevan (bisa terjadi bila terdapat data/dokumen yang dihapus atau ditambah data).

Setiap kali terdapat perubahan (penambahan / pengurangan dokumen ataupun teksnya) dalam kumpulan dokumen pada lokasi (direktori) yang sama, perlu dilakukan pengindeksan dan penghitungan ulang dari awal untuk mendapat-kan nilai SVD yang baru. Algoritma untuk memperbaharui (update) SVD diperlukan untuk menghemat waktu dan memori daripada meng-hitung keseluruhan nilai SVD dari awal.

DAFTAR PUSTAKA

Alhensiri, A. A., *Web Information Retrieval and Search Engine Techniques*, Al-Satil Journal, Libya, 2003, 55-92.

Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*, England: Pearson Education Limited, 1999.

Basuki, T. A., Penggunaan Semi Discrete Decomposition pada Latent Semantic Indexing untuk Temu-Kembali Informasi, INTEGRAL, vol. 6 no. 1, April 2001, 5-13.

Berry, M. W., Dumais, S. T., O'Brien, G. W., *Using Linear Algebra for Intelligent Information Retrieval, SIAM Review*, 37(4), 1995, 573-595.

Berry, M. W., Browne, M.,*Understanding Search Engines: Mathematical Modelling and Text Retrieval Second Edition, Philadelphia*: Society for Industrial and Applied Mathematics, 2005.

Blom, K., *Information Retrieval using the Singular Value Decomposition and Krylov subspaces*, Sweden: Department of Mathematics, Chalmers University of Technology, 1999.

Chen, C., Stoffel, N., Post, M., Basu, C., Bassu, D., Behrens, C., Telcordia LSI Engine: Implementation and Scalability Issues, *Proc. of the 11th Int. Workshop on Research Issues in Data Engineering (RIDE 2001): Document Management for Data Intensive Business and Scientific Applications, Heidelberg, Germany, Apr. 1-2, 2001*.

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., Harshman, R. A., *Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science*, 41(6), 1990, 391-407.

Dumais, S. T., LSI meets TREC: A Status Report, In: D. Harman (Ed.), *The First Text REtrieval Conference (TREC1), National Institute of Standards and Technology Special Publication* 500-207, 1993, pp. 137-152.

Letsche, T. A., *Toward Large-Scale Information Retrieval Using Latent Semantic Indexing*, Master of Science Thesis, Knoxville: University of Tennessee, Knoxville, 1996.