Sistem Temu Kembali Informasi dengan Pemeringkatan Metode Vector Space Model

Fatkhul Amin
Program Studi Teknik Informatika, Universitas Stikubank
email: fatkhulamin@gmail.com

Abstract

The objective of designing information retrieval system (IRS) with Vector Space Model (VSM) Method is to facilitate users to search Indonesian documents. IRS Software is designed to provide search results with the optimum number of documents (low recall) and accuracy (high precision) with VSM method that users may get fast and accurate results. VSM method provides a different credit for each document stored in a database which in turns to determine the document most similar to the query, where the documents with the highest credits are placed on the top of the search results. The evaluation of search results with IRS is conducted under recall and precision tests. This study fascinatingly creates a system which can preprocess (tokenizing, filtering, and stemming) within computation time of four minutes forty-one seconds.

Keywords: IRS, Vector Space Model, recall, precision

PENDAHULUAN

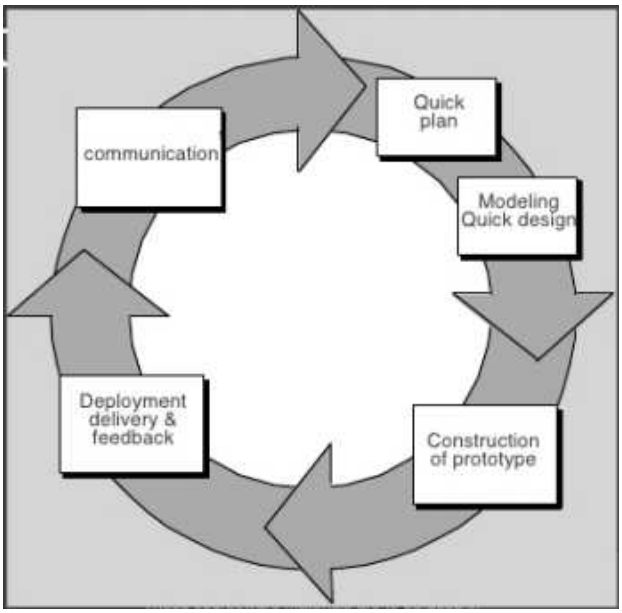
Pencarian informasi saat ini dilakukan dengan menggunakan mesin pencari atau Sistem Temu Kembali Informasi (STKI), *user* menuliskan *query* dan mesin pencari akan menampilkan hasil pencarian. Mesin pencari yang sudah ada dan banyak digunakan saat ini memberikan hasil perolehan pencarian yang banyak (banyak dokumen yang terambil), sehingga diperlukan waktu untuk menentukan hasil pencarian yang relevan. Menentukan hasil yang relevan sesuai dengan keinginan user dengan jumlah hasil pencarian yang banyak akan menyulitkan *user*. Hal ini terjadi karena dokumen yang terambil oleh sistem jumlahnya banyak, maka sistem berkemungkinan menampilkan hasil pencarian yang tidak relevan. Banyaknya dokumen hasil pencarian ini membuat waktu yang dibutuhkan dalam pencarian menjadi lebih banyak dari yang diharapkan.

Perkembangan penelusuran informasi saat ini menghasilkan *recall* yang tinggi dan *precision* yang rendah. *Recall* yang tinggi diartikan bahwa dokumen yang dihasilkan dalam penelusuran dokumen adalah banyak, sedangkan *precision* rendah dapat diartikan bahwa dokumen yang diharapkan dapat ditemukan sedikit.

METODE

1. Metode Pengembangan Sistem

STKI menggunakan Metode pengembangan Prototipe (*Evolutionary*). sistem yang sesungguhnya dipandang sebagai evolusi dari versi awal yang terbatas menuju produk finalnya. Sistem selalu di evaluasi untuk didapatkan sistem yang paling baik. Model prototipe yang digunakan pada pengembangan STKI bisa dilihat pada gambar 1.



Gambar 1. Evolutionary Prototyping

2. Metode Penyelesaian Masalah

STKI dengan menggunakan metode *Vector Space Model (VSM)* dirancang untuk mendapatkan hasil pencarian dengan *precision* tinggi dan *recall* rendah. Metode VSM dipilih karena cara kerja model ini efisien, mudah dalam representasi dan dapat diimplementasikan pada *document-matching*.

LANDASAN TEORI

1. Sistem Temu Kembali Informasi

Sistem temu kembali informasi *(information retrieval system)* merupakan suatu sistem yang menemukan *(retrieve)* informasi yang sesuai dengan kebutuhan *user* dari kumpulan informasi secara otomatis. Prinsip kerja sistem temu kembali informasi jika ada sebuah kumpulan dokumen dan seorang user yang memformulasikan sebuah pertanyaan (*request* atau *query*). Jawaban dari pertanyaan tersebut adalah sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan (Salton, 1989).

Sistem temu kembali informasi akan mengambil salah satu dari kemungkinan tersebut. Sistem temu kembali informasi dibagi dalam dua komponen utama yaitu sistem pengindeksan *(indexing)* menghasilkan basis data sistem dan temu kembali merupakan gabungan dari *user interface* dan *look-up-table*. Sistem temu kembali informasi didesain untuk menemukan dokumen atau informasi yang diperlukan oleh *user*.

Sistem Temu Kembali Informasi bertujuan untuk menjawab kebutuhan informasi *user* dengan sumber informasi yang tersedia dalam kondisi seperti sebagai berikut (Salton, 1989);

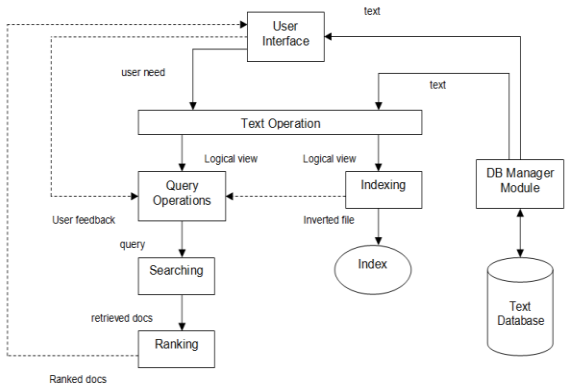
a. Mempresentasikan sekumpulan ide dalam sebuah dokumen menggunakan sekumpulan konsep.
b. Terdapat beberapa pengguna yang memerlukan ide, tapi tidak dapat mengidentifikasikan dan menemukannya dengan baik.
c. Sistem temu kembali informasi bertujuan untuk mempertemukan ide yang dikemukakan oleh penulis dalam dokumen dengan kebutuhan informasi pengguna yang dinyatakan dalam bentuk *key word query*/istilah penelusuran.

Fungsi utama sistem temu kembali informasi (Salton, 1989);

a. Mengidentifikasi sumber informasi yang relevan dengan minat masyarakat pengguna yang ditargetkan
b. Menganalisis isi sumber informasi (dokumen)
c. Merepresentasikan isi sumber informasi dengan cara tertentu yang memungkinkan untuk dipertemukan dengan pertanyaan pengguna
d. Merepresentasikan pertanyaan (*query*) *user* dengan cara tertentu yang memungkinkan untuk dipertemukan sumber informasi yang terdapat dalam basis data.
e. Mempertemukan pernyataan pencarian dengan data yang tersimpan dalam basis data
f. Menemu-kembalikan informasi yang relevan
g. Menyempurnakan unjuk kerja sistem berdasarkan umpan balik yang diberikan oleh user.

2. *Arsitektur Sistem Temu Kembali Informasi*

Proses Sistem temu kembali informasi seperti pada gambar 2 menggunakan arsitektur yang sederhana. Sebelum dilakukannya proses temu kembali diperlukan pendefinisian database. Selanjutnya mengikuti tahapan proses; Dokumen-dokumen yang akan digunakan, Operasi yang akan digunakan dalam pencarian, dan model pengolahan teks (Yates, 1999).



Gambar 2. *The Process of Retrieving Information*

3. Arsitektur Sistem Temu Kembali Informasi

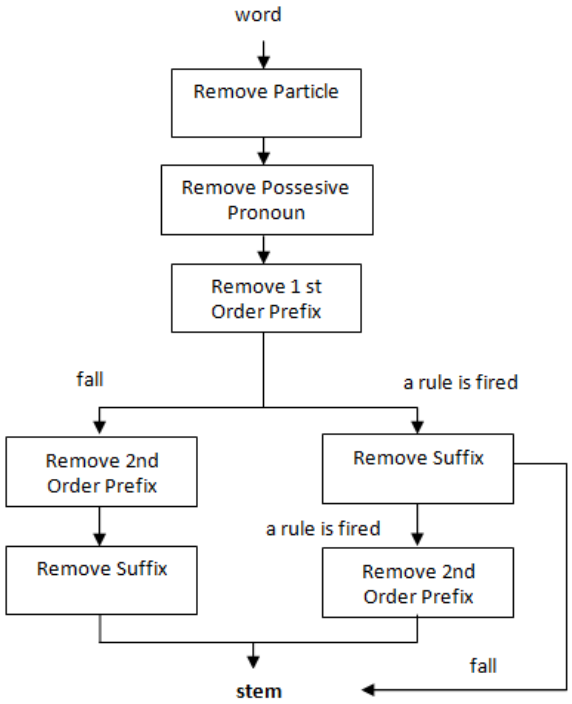
Tokenisasi merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca. Sebagai contoh, kata-kata "*computer*", "*computing*", dan "*compute*" semua berasal dari term yang sama yaitu "*comput*" , tanpa pengetahuan sebelumnya dari morfologi bahasa Inggris. Token seringkali disebut sebagai istilah (*term*) atau kata, sebagai contoh sebuah token merupakan suatu urutan karakter dari dokumen tertentu yang dikelompokkan sebagai unit semantik yang berguna untuk diproses (Salton, 1989).

4. *Filtering*

Eliminasi *stopwords* memiliki banyak keuntungan, yaitu akan mengurangi *space* pada tabel *term index* hingga 40% atau lebih (Yates, 1999). Proses *stopword removal* merupakan proses penghapusan *term* yang tidak memiliki arti atau tidak relevan. Proses ini dilakukan pada saat proses tokenisasi. Proses *Filtering* menggunakan daftar *stopword* yang digunakan oleh Tala (2003), yang merupakan *stopword* bahasa Indonesia yang berisi kata-kata seperti; ada, yang, ke, kepada, dan lain sebagainya

5. *Stemming*

Proses *Stemming* digunakan untuk mengubah *term* yang masih melekat dalam term tersebut awalan, sisipan, dan akhiran. Proses stemming dilakukan dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan *(prefixes*), sisipan *(infixes)*, akhiran *(suffixes)* dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang benar (Tala, 2003). Arsitektur proses *stemming* untuk bahasa Indonesia dapat dilihat pada gambar 3.



Gambar 3. *The basic design of a Porter stemmer for* Bahasa Indonesia

6. *Inverted Index*

Pada prinsipnya proses menemukan *records* adalah menjawab dari permintaan *(request)* informasi didasarkan pada kemiripan diantara *query* dan kumpulan *term* pada sistem (Salton, 1989). *Inverted file* atau *inverted index*

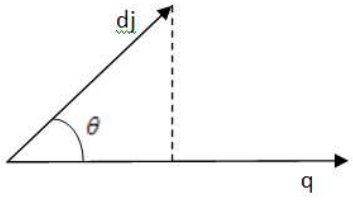
merupakan mekanisme untuk pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Elemen penting dalam struktur *inverted file* ada dua, yaitu: kata *(vocabulary)* dan kemunculan *(occurrences).*

7. *Vector Space Model*

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan *(similarity)* term dengan cara pembobotan *term*. Dokumen dipandang sebagi sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model,* sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* (Yates, 1999).

VSM memberikan sebuah kerangka pencocokan parsial adalah mungkin. Hal ini dicapai dengan menetapkan bobot non-biner untuk istilah indeks dalam *query* dan dokumen. Bobot istilah yang akhirnya digunakan untuk menghitung tingkat kesamaan antara setiap dokumen yang tersimpan dalam sistem dan permintaan user. Dokumen yang terambil disortir dalam urutan yang memiliki kemiripan, model vektor memperhitungkan pertimbangan dokumen yang relevan dengan permintaan user. Hasilnya adalah himpunan dokumen yang terambil jauh lebih akurat (dalam arti sesuai dengan informasi yang dibutuhkan oleh *user*).

Sebuah dokumen dj dan sebuah *query* q direpresentasikan sebagai vektor t-dimensi seperti pada gambar 4.



Gambar 4. The Cosines of θ is adopted as sim d_j, q

Dalam VSM koleksi dokumen direpresentasikan sebagai sebuah matrik *term* document (atau matrik *term frequency*). Setiap sel dalam matrik bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. Nilai nol berarti bahwa term tersebut tidak ada dalam dokumen. Gambar 5 menunjukkan matrik *term document* dengan n dokumen dan t *term*.

	T_1	T_2	T_3	$T_{...}$	T_t
D_1	W_{11}	W_{21}	W_{31}	$W_{...}$	T_{t1}
D_2	W_{12}	W_{22}	W_{32}	$W_{...}$	T_{t2}
D_3	W_{13}	W_{23}	W_{33}	$W_{...}$	T_{t3}
$D_{...}$	$W_{...}$	$W_{...}$	$W_{...}$	$W_{...}$	$W_{...}$
D_n	W_{1n}	W_{2n}	W_{3n}	$W_{...}$	T_{tn}

Gambar 5. Matrik *term-document*

Proses perhitungan VSM melalui tahapan perhitungan *term frequency (tf)* menggunakan persamaan (2)

$$tf = tf_{ij} \quad \text{.....(2)}$$

dengan *tf* adalah *term frequency,* dan $tf_{i,j}$ adalah banyaknya kemunculan term t_i dalam dokumen d_j, *Term frequency (tf)* dihitung dengan menghitung banyaknya kemunculan term t_i dalam dokumen d_j .

Perhitungan *Inverse Document Frequency (idf),* menggunakan persamaan (3)

$$idf_i = \log \frac{N}{df_i} \quad \text{.....(3)}$$

dengan idf_i adalah *inverse document frequency,* N adalah jumlah dokumen yang terambil oleh sistem, dan df_i adalah banyaknya dokumen dalam koleksi dimana term t_i muncul di dalamnya, maka Perhitungan idf_i digunakan untuk mengetahui banyaknya term yang dicari (df_i) yang muncul dalam dokumen lain yang ada pada database (korpus).

Perhitungan *term frequency Inverse Document Frequency (tfidf),* menggunakan persamaan (4)

$$W_{ij} = tf_{ij} . \log(\frac{N}{df_i}) \quad \text{.....(4)}$$

dengan W_{ij} *adalah bobot dokumen, N adalah* Jumlah dokumen yang terambil oleh system, $tf_{i,j}$ adalah banyaknya kemunculan *term* t_i pada dokumen dj, dan df_i adalah banyaknya dokumen dalam koleksi dimana term t_i muncul di dalamnya. Bobot dokumen (W_{ij}) dihitung untuk didapatkannya suatu bobot hasil perkalian atau kombinasi antara term frequency ($tf_{i,j}$) dan Inverse Document Frequency (df_i).

Perhitungan Jarak *query* menggunakan persamaan (5) dan dokumen, menggunakan persamaan (6)

$$| q |= \sqrt{\sum_{j=1}^{t} (W_{iq})^2} \quad \text{.....(5)}$$

dengan |q| adalah Jarak query, dan W_{iq} adalah bobot query dokumen ke-i, maka Jarak *query* ($|q|$) dihitung untuk didapatkan jarak *query* dari bobot query dokumen (W_{iq}) yang terambil oleh sistem. Jarak *query* bisa dihitung dengan persamaan akar jumlah kuadrat dari *query*.

$$| d_j |= \sqrt{\sum_{i=1}^{t} (W_{ij})^2} \quad \text{.....(6)}$$

dengan $|d_j|$ adalah jarak dokumen, dan W_{ij} adalah bobot dokumen ke-i, maka Jarak dokumen ($|d_j|$) dihitung untuk didapatkan jarak *dokumen* dari bobot dokumen dokumen (W_{ij}) yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen.

Perhitungan pengukuran Similaritas *query document (inner product),* menggunakan persamaan (7)

$$sim(q, d_j) = \sum_{i=1}^{t} W_{iq}.W_{ij} \quad \text{.....(7)}$$

dengan W_{ij} adalah bobot term dalam dokumen, W_{iq} adalah bobot query, dan Sim (q, dj) adalah Similaritas antara *query* dan dokumen. Similaritas antara *query* dan dokumen atau *inner product/Sim (q, dj)* digunakan untuk mendapatkan bobot dengan didasarkan pada bobot *term* dalam dokumen (W_{ij}) dan bobot query (W_{iq}) atau dengan cara menjumlah bobot *q* dikalikan dengan bobot dokumen.

Pengukuran *Cosine Similarity* (menghitung nilai kosinus sudut antara dua vector) menggunakan persamaan (1)

$$Sim(q, d_j) = \frac{q.d_j}{|q| * |d_j|} = \frac{\sum_{i=1}^{t} W_{iq} . W_{ij}}{\sqrt{\sum_{j=1}^{t} (W_{iq})^2} * \sqrt{\sum_{i=1}^{t} (W_{ij})^2}}$$

(1)

Similaritas antara *query* dan dokumen atau Sim(q,dj) berbanding lurus terhadap jumlah bobot query (q) dikali bobot dokumen (dj) dan berbanding terbalik terhadap akar jumlah kuadrat q (|q|) dikali akar jumlah kuadrat dokumen (|dj|). Perhitungan similaritas menghasilkan bobot dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan *inner product*.

8. *Uji Recall dan Precision*

Tujuan uji *Recall* dan *Precision* adalah untuk mendapatkan informasi hasil pencarian yang didapatkan oleh STKI. Hasil pencarian STKI bisa dinilai tingkat *recall* dan *precision* nya. *Precision* dapat dianggap sebagai ukuran ketepatan atau ketelitian, sedangkan *recall* adalah kesempurnaan. Nilai *precision* adalah proporsi dokumen yang terambil oleh sistem adalah relevan. Nilai *recall* adalah proporsi dokumen relevan yang terambil oleh sistem (Salton, 1989).

Nilai *recall* dan *precision* bernilai antara 0 sd 1. Sistem temu kembali informasi diharapkan untuk dapat memberikan nilai *recall* dan *precision* mendekati 1. Pengguna rata-rata ingin mencapai nilai *recall* tinggi dan *precision* tinggi, pada kenyataannya hal itu harus dikompromikan karena sulit dicapai (Salton, 1989).

$$R = \frac{Number\ of\ relevant\ items\ retrieved}{Total\ number\ of\ relevant\ items\ in\ collection} \quad \text{....(8)}$$

dengan R adalah *recall,*maka nilai *R* didapatkan dengan membandingkan *Number of relevant items retrieved* dengan *Total number of relevant items in collection*. *Recall* adalah dokumen yang terpanggil dari STKI sesuai dengan permintaan *user* yang mengikuti pola dari STKI. Nilai *recall* makin besar belum bisa dikatakan suatu STKI baik atau tidak.

$$R = \frac{Number\ of\ relevant\ items\ retrieved}{Total\ number\ of\ relevant\ items\ retrieved} \quad \ldots.(9)$$

dengan *P* adalah *Precision.* maka nilai *P* didapatkan dengan membandingkan *Number of relevant items retrieved* dengan *Total number of items retrieved.* *Precision* adalah jumlah dokumen yang terpanggil dari *database* relevan setelah dinilai *user* dengan informasi yang dibutuhkan. Semakin besar nilai *precision* suatu STKI, maka STKI bisa dikatakan baik.

IMPLEMENTASI

1. Sistem temu kembali informasi

Proses Tokenisasi dilakukan dengan mekanisme jika dokumen pada korpus ditemukan spasi, maka *term* yang ada diantara spasi akan di *retrieved* (akan diambil oleh sistem) kemudian *term* ditempatkan dalam tabel tabelawal. Hasil proses berupa *term* asli (term yang masih memiliki imbuhan, tanda baca yang melekat, dan angka). Keunggulan proses ini waktu komputasi cepat.

Proses *Filtering* dilakukan dengan mekanisme jika *term* pada tabel tabelawal ditemukan tanda baca, huruf kapital, dan angka. Maka program akan menghilangkan (tanda baca dan angka) dan mengganti (huruf kapital menjadi huruf kecil), kemudian memeriksa *term* dengan *stopwords*. Hasil proses berupa *term* pilihan (tanpa tanda baca, tanpa huruf kapital, dan bukan termasuk *stopwords*). Keunggulan proses ini sistem mampu mereduksi tanda baca, angka, merubah term menjadi huruf kecil, dan memeriksa term stopwords dengan waktu komputasi yang cepat.

Proses *Stemming* dilakukan program dengan cara menghilangkan imbuhan yang terdapat pada term hasil *filtering*. Proses menghilangkan dilakukan dengan menghilangkan awalan, sisipan, dan akhiran. Hasil proses ini dimasukkan dalam tabel tabelfreq.

Proses Pembobotan dokumen dengan metode VSM dilakukan dalam proses pencarian dokumen. Program akan bekerja ketika *user* melakukan *query*, selanjutnya program akan memproses *query* tersebut dengan perhitungan-perhitungan *tf*, *idf, tfidf*, jarak *query* dan dokumen, similaritas dan *cosine similarity*

2. *Proses STKI*

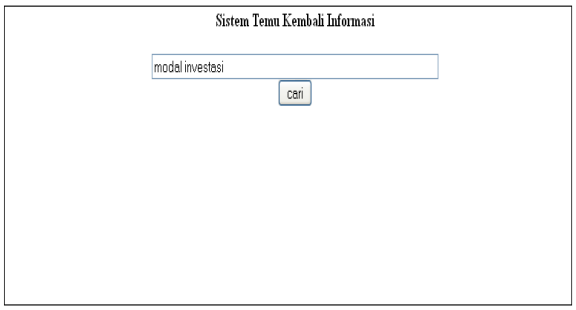
Kumpulan abstrak diletakkan pada Tabel korpus (tabel 1), Selanjutany data pada tabel korpus digunakan dalam proses tokenisasi, *filtering*, dan *stemming*.

Tabel 1. Tabel Korpus

id	judul	isi	dokumen
1	Pengujian teori static trade off dan pecking order...	Pendanaan perusahaan mengikuti beberapa pola tertе...	m1
3	Tinjauan rencana pengembangan investasi jangka men...	Dalam melakukan investasi umumnya perusahaan dihad...	m2
2	Analisis hubungan antara audience characteristics ...	Product placement merupakan suatu hal yang tidak a...	m3
4	Pengaruh penganggaran partisipatif terhadap kesenj...	Penelitian ini bertujuan untuk mencari tahu dan me...	a1
5	Pengaruh atribut-atribut dari program pembinaan lo...	Penelitian ini membahas tentang pengaruh atribut-a...	m4
6	Analisis capital structure dan hubungannya dengan ...	Skripsi ini secara keseluruhan ingin meneliti peng...	a2
7	Uji interdependensi Bursa Saham Indonesia dengan B...	Penelitian ini bertujuan untuk melihat adakah hubu...	m5
8	Pengaruh persepsi nilai konsumen terhadap perilaku...	Pertumbuhan bisnis ritel semakin meningkat dengan ...	m6
9	Pengaruh tingkat dan strategi diversifikasi terhad...	Diversifikasi adalah perusahaan yang mengembangkan...	m7
10	Peranan persepsi konsumen atas service brand dalam...	Skripsi ini membahas tentang peranan dari persepsi...	m8

Hasil *scanner term* pada korpus diletakkan pada tabel awal. Hasil *scanner term* yang ada pada tabel awal adalah 53.223 term dengan space 7,3 MB. Hasil scanner *term* selanjutnya diletakkan pada tabel kedua. Hasil penscanneran term pada tabel kedua adalah 29.404 term dengan space 2,6 MB. Hasil dari proses *filtering* menjadikan jumlah *term* berkurang menjadi 17.930 dari jumlah *term* semula 29.404. Pengurangan ini membuat space yang dibutuhkan untuk tabel term menjadi 2,6 MB. Proses *stemming* menghasilkan kumpulan term berupa kata dasar hasil scanner *term* pada tabel kedua. Proses *stemming* didukung stopword tala yang digunakan untuk mengurangi term yang ada pada tabel kedua. Jumlah *stopword removal* tala adalah 758 term dengan space 15,8 KB

Selanjutnya dilakukan perhitungan VSM oleh sistem dan dibuat *inteface* (gambar 6) untuk user dan Contoh tampilan hasil pencarian keyword "modal investasi " (gambar 7).



Gambar 6. *Interface* Sistem Temu Kembali Informasi

Gambar 7. Hasil Pencarian keyword "modal investasi"

3. *Uji Recall dan Precision*

Studi kasus menggunakan STKI ini dilakukan dengan memasukkan *keyword* 2 *term* (modal investasi, ekonomi indonesia, bursa efek) dan 3 *term* (bursa efek jakarta, bursa efek indonesia). Hasil pencarian selanjutnya dilakukan uji *recall* dan *precision* seperti terlihat pada tabel 2.

Tabel 2. Hasil Pengujian *Recall* dan *Precision*

No	Term	Recall	Precision
1	Modal Investasi	0,22	0,83
2	Ekonomi Indonesia	0,43	0,84
3	Bursa Efek	0,15	0,87
4	Bursa Efek Jakarta	0,12	0,64
5	Bursa Efek Indonesia	0,52	0,99

4. Pengujian program berdasarkan waktu

Waktu yang dibutuhkan dalam proses komputasi proses tokenisasi dengan jumlah dokumen 300 adalah 24 detik. Sedangkan waktu komputasi untuk proses *filtering* adalah 4 menit 17 detik. Jadi total waktu yang dibutuhkan dalam proses preprosesing dengan jumlah dokumen 300 adalah 4 menit 41 detik (perhitungan waktu komputasi dilakukan menggunakan *stopwatch*).

Proses pencarian term : 2 *term* dan 3 *term* dengan menggunakan sistem temu kembali informasi (metode *vector space model*) membutuhkan waktu komputasi rata-rata 1,5 detik (perhitungan waktu komputasi dilakukan menggunakan *stopwatch*).

Waktu preposesing untuk *scanner file* dari tabelkedua menjadi tabelfrek dipilih dengan jumlah dokumen 300, hal ini dikarenakan ketika digunakan data sejumlah 400 dokumen, terjadi *error* karena waktu komputasi berlangsung lebih dari 10 menit,sementara waktu komputasi menggunakan pc ini adalah maksimal 10 menit.

KESIMPULAN

Sistem Temu Kembali Informasi (STKI) dengan studi kasus menggunakan 300 dokumen abstraksi skripsi mahasiswa menghasilkan; STKI mampu melakukan tokenisasi (pemisahan *term*), *filtering term* (reduksi tanda baca, angka, dan *stopwords*) dan *Stemming term* (membuat kata dasar) dengan waktu komputasi 4 menit 41 detik.

Hasil Uji *recall* dan *precision* STKI menunjukkan hasil pencarian dokumen teks bahasa Indonesia memiliki rata-rata *recall* = 0,19 dan rata-rata *precision* = 0,54.

STKI yang dibangun memiliki keunggulan mampu melakukan pencarian dokumen teks bahasa Indonesia dengan waktu komputasi rata-rata 1,5 detik dan hasil pencarian dengan nilai *precision* = 0,54 serta dilengkapi dengan bobot dan letak dokumen pada database.

DAFTAR PUSTAKA

Budi, G.S., Gunawan, I., (2006). *Algoritma Porter Stemmer* for Bahasa Indonesia untuk *Pre-Processing Text Mining* Berbasis Metode Market Basket Analisys. UK Petra.

Budi, I., Aji, R.F., (2006). Efektifitas Seleksi Fitur dalam Sistem Temu Kembali Informasi. Seminar Nasional Aplikasi Teknologi Informasi (SNATI), ISSN : 1907-5022.

Bum, K.Y., (2010). *An autonomous assessment system based on combined latent semantic kernels. Expert Systems with Applications:*

An International Journal , Volume 37 Issue 4.

Bunyamin., (2008). Aplikasi IR CATA dengan Metode *Generalized Vector Space Model*, Jurnal Informatika, Vol. 4. No. 1

Bo, Y., (2009). Combining neural networks and semantic feature space for email classification. Knowledge-Based *Systems* , Volume 22 Issue 5.

Erk, K., (2008). *A Structured Vector Space Model for Word Meaning in Context. Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*. 897-906.

Erk, K., (2010). *A flexible, corpus-driven model of regular and inverse selectional preferences . Computational Linguistics* , Volume 36 Issue 4.

Haryono, M.E.A., (2005). Pembentukan Intisari Topik secara Otomatis dalam suatu Paragraf dengan *Vector Space Model.* Seminar Nasional Aplikasi Teknologi Informasi, SNATI 2005.

Haryono, M.EA., Wahyudi., (2005). *Customer Information Gathering* menggunakan Metode Temu Kembali Informasi dengan Model Ruang Vektor. Seminar Nasional Aplikasi Teknologi Informasi, SNATI 2005.

Hasugian, J., (2006). Penggunaan Bahasa Alamiah dan Kosa Kata Terkendali dalam Sistem Temu Balik Informasi berbasis Teks. Jurnal Studi Perpustakaan dan Informasi, Vol 2, No 2 Desember.

Kadir, A., (2001). Dasar Pemrograman Web Dinamis menggunakan PHP. Penerbit Andi. Yogyakarta.

Lopez, C., Ribeiro, C., (2010). *Using Local Precision to Compare Search Engines in Consumer Health Information Retrieval.* **SIGIR '10:** Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval.

Manning, C., Raghavan, P., (2007). An Introduction to Information Retrieval, Stanford. USA.

Mao, W., (2007). *The phrase-based vector space model for automatic retrieval of free-text medical documents*. *Data & Knowledge Engineering* , Volume 61 Issue 1.

Meadow, C.T., (1997). *Text Information Retrieval Systems*. Academic Press.New York.

Mcenery,(1998).**W3Corporaproject**.*1998*http://www.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/introduction2.html diakses tanggal 14 juli 2012.

Mondal, D., Gangopadhyay, A., (2010). *Medical Decision Making Using Vector Space Model.* IHI '10**:** *Proceedings of the 1st ACM International Health Informatics Symposium*.

Peranginangin, K., (2006). Aplikasi Web dengan PHP dan MySQL. Penerbit Andi, Yogyakarta.

Smith, (1991). *Software Prototyping: Adoption, Practice and Management*. McGraw-Hill, London.

Tala, F.Z., (2003). *A Study of Stemming Effects on Information Retrieval in bahasa Indonesia*. Institut for logic, Language and Computation Universiteit van Amsterdam The Netherlands.

Rijsbergen, C.J.V., (1979). Information *Retrieval, Second Edition. Butterworths*. London.

Salton, G., (1989). Automatic Text Processing, The Transformation, Analysis, and Retrieval of information by computer. Addison – Wesly Publishing Company, Inc. USA.

Yates, R.B, (1999). *Modern Information Retrieval*, *Addison Wesley*-Pearson international edition, Boston. USA.