Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012 (Semantik 2012)
Semarang, 23 Juni 2012
ISBN 979 - 26 - 0255 - 0

TEMU BALIK INFORMASI PADA DOKUMEN TEKS BERBAHASA INDONESIA DENGAN METODE *VECTOR SPACE RETRIEVAL MODEL*

Giat Karyono[1], Fandy Setyo Utomo[2]
[1]*Program Studi Teknik Informatika, STMIK AMIKOM Purwokerto*
E-mail : giant_mercy@yahoo.co.id

[2]*Program Studi Sistem Informasi, STMIK AMIKOM Purwokerto*
E-mail : fandy_setyo_utomo@yahoo.com

ABSTRAK

Pencarian informasi berdasarkan query oleh pengguna, yang diharapkan dapat menemukan koleksi dokumen berdasarkan kebutuhan pengguna, dikenal dengan Information Retrieval atau temu balik informasi. Penelitian ini membahas tentang implementasi sistem temu balik informasi untuk mencari dan menemukan dokumen teks berbahasa indonesia menggunakan Vector Space Retrieval Model. Tujuan penelitian ini untuk menyediakan solusi pada mesin pencarian agar mampu menyediakan informasi dokumen teks pada database yang tepat menggunakan kata kunci tertentu. Hasil dari pencarian direpresentasikan dengan urutan/ranking kemiripan dokumen dengan query.

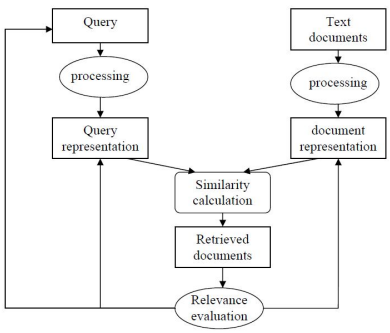
Kata kunci : Information Retrieval, Temu Balik Informasi, Vector Space Retrieval Model

1. PENDAHULUAN

ISO 2382/1 mendefinisikan *Information Retrieval* (IR) atau temu balik informasi sebagai tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan informasi mengenai subyek yang dibutuhkan. Tindakan tersebut mencakup *text indexing*, *inquiry analysis*, dan *relevance analysis* [2]. Data mencakup teks, tabel, gambar, ucapan, dan video. Informasi termasuk pengetahuan terkait yang dibutuhkan untuk mendukung penyelesaian masalah dan akuisisi pengetahuan. Tujuan dari sistem temu balik informasi adalah memenuhi kebutuhan informasi pengguna dengan me-*retrieve* semua dokumen yang mungkin relevan, pada waktu yang sama me-*retrieve* sesedikit mungkin dokumen yang tak relevan. Sistem ini menggunakan fungsi heuristik untuk mendapatkan dokumen-dokumen yang relevan dengan *query* pengguna. Sistem temu balik informasi yang baik memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhannya. Agar representasi dokumen lebih baik, dokumen-dokumen dengan topik atau isi yang mirip dikelompokkan bersama-sama [5]. Tujuan dari penelitian ini adalah merancang sebuah perangkat lunak temu balik informasi untuk mencari dan mencocokan dokumen teks berbahasa Indonesia menggunakan *Vector Space Retrieval Model* dengan tujuan memberikan sebuah solusi pada mesin pencarian untuk memberikan informasi kecocokan teks dalam *database* dengan menggunakan kata kunci tertentu, hasil dari pencocokan tersebut disajikan dalam bentuk peringkat.

2. TEORI

2.1. Arsitektur Sistem Temu Balik Informasi



Gambar 1 : Proses Temu Balik Informasi Dokumen Teks [4]

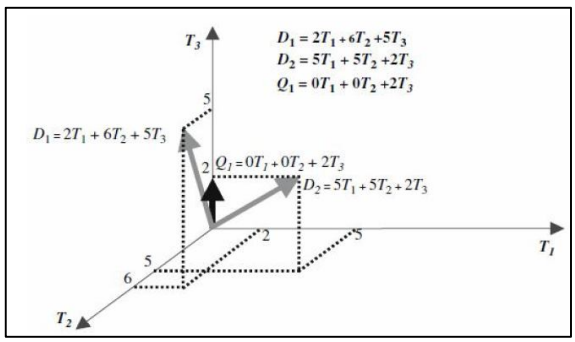
Ada dua pekerjaan yang ditangani oleh sistem ini, yaitu melakukan *pre-processing* terhadap *database* dan kemudian menerapkan metode tertentu untuk menghitung kedekatan (relevansi atau *similarity*) antara dokumen di dalam *database* yang telah di*preprocess* dengan *query* pengguna. *Query* yang dimasukkan pengguna dikonversi sesuai aturan tertentu untuk mengekstrak *term-term* penting yang sejalan dengan *term-term* yang sebelumnya telah diekstrak dari dokumen dan menghitung relevansi antara *query* dan dokumen berdasarkan pada *term-term* tersebut. Sebagai hasilnya, sistem mengembalikan suatu daftar dokumen terurut sesuai nilai kemiripannya dengan *query* pengguna [4]. Setiap dokumen (termasuk *query*) direpresentasikan menggunakan model *bag-of-words* yang mengabaikan urutan dari kata-kata di dalam dokumen, struktur sintaktis dari dokumen dan kalimat. Dokumen ditransformasi ke dalam suatu "tas" berisi kata-kata *independen. Term* disimpan dalam suatu *database* pencarian khusus yang ditata sebagai sebuah *inverted index*. Index ini merupakan konversi dari dokumen asli yang mengandung sekumpulan kata ke dalam daftar kata yang berasosiasi dengan dokumen terkait dimana kata-kata tersebut muncul. Proses dalam *Information Retrieval* dapat digambarkan sebagai sebuah proses untuk mendapatkan *retrieve document* dari *collection documents* yang ada melalui pencarian *query* yang diinputkan user.

2.2. Vector Space Retrieval Model

Pada sistem IR, kemiripan antar dokumen didefinisikan berdasarkan representasi *bag of words* dan dikonversi ke suatu model ruang vektor (vector space model, VSM). Model ini diperkenalkan oleh Salton [7] dan telah digunakan secara luas. Pada VSM, setiap dokumen di dalam database dan query pengguna direpresentasikan oleh suatu vektor multi-dimensi [2, 6]. Dimensi sesuai dengan jumlah term dalam dokumen yang terlibat. Pada model ini:

A. Vocabulary merupakan kumpulan semua term berbeda yang tersisa dari dokumen setelah preprocessing dan mengandung *t* term index. Term-term ini membentuk suatu ruang vektor.
B. Setiap *term* i di dalam dokumen atau *query* j, diberikan suatu bobot (*weight*) bernilai *real* wij.
C. Dokumen dan *query* diekspresikan sebagai vektor t dimensi dj = (w1, w2, ..., wtj) dan terdapat n dokumen di dalam koleksi, yaitu j = 1, 2, ..., n.

Contoh dari model ruang vektor tiga dimensi untuk dua dokumen D1 dan D2, satu query pengguna Q1, dan tiga term T1, T2 dan T3 diperlihatkan pada gambar 2 berikut ini,



Gambar 2 : Contoh Model Ruang

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks term document (atau matriks term-frequency). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu term dalam dokumen yang ditentukan. Nilai nol berarti bahwa term tersebut tidak hadir di dalam dokumen. Gambar 3 mempertegas penjelasan ini [2].

$$\begin{bmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ \dots & \dots & \dots & \dots & \dots \\ D_n & w_{1n} & w_{2n} & \dots & w_m \end{bmatrix}$$

Gambar 3 : Contoh matriks term document untuk database dengan n dokumen dan t term [2]

Keberhasilan dari model VSM ini ditentukan oleh skema pembobotan terhadap suatu term baik untuk cakupan lokal maupun global, dan faktor normalisasi [6]. Pembobotan lokal hanya berpedoman pada frekuensi munculnya term dalam suatu dokumen dan tidak melihat frekuensi kemunculan term tersebut di dalam dokumen lainnya. Pendekatan dalam

pembobotan lokal yang paling banyak diterapkan adalah term frequency (tf) meskipun terdapat skema lain seperti pembobotan biner, *augmented normalized* tf, logaritmik tf dan logaritmik alternatif.
Pembobotan global digunakan untuk memberikan tekanan terhadap *term* yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari term tertentu di seluruh dokumen. Banyak skema didasarkan pada pertimbangan bahwa semakin jarang suatu *term* muncul di dalam total koleksi maka *term* tersebut menjadi semakin berbeda. Pemanfaatan pembobotan ini dapat menghilangkan kebutuhan *stop word removal* karena *stop word* mempunyai bobot global yang sangat kecil. Namun pada prakteknya lebih baik menghilangkan *stop word* di dalam fase *pre-processing* sehingga semakin sedikit term yang harus ditangani. Pendekatan terhadap pembobotan global mencakup *inverse document frequency* (idf), *squared* idf, *probabilistic* idf, GF-idf, entropy. Pendekatan idf merupakan pembobotan yang paling banyak digunakan saat ini. Beberapa aplikasi tidak melibatkan bobot global, hanya memperhatikan tf, yaitu ketika tf sangat kecil atau saat diperlukan penekanan terhadap frekuensi term di dalam suatu dokumen [6].
Faktor normalisasi digunakan untuk menormalkan vektor dokumen sehingga proses *retrieval* tidak terpengaruh oleh panjang dari dokumen. Normalisasi ini diperlukan karena dokumen panjang biasanya mengandung perulangan *term* yang sama sehingga menaikkan frekuensi *term* (tf). Dokumen panjang juga mengandung banyak *term* yang berbeda sehingga menaikkan ukuran kemiripan antara query dengan dokumen tersebut, meningkatkan peluang di-*retrieve*nya dokumen yang lebih panjang. Beberapa pendekatan normalisasi adalah normalisasi cosinus, penjumlahan bobot, normalisasi ke- 4, normalisasi bobot maksimal dan normalisasi pivoted unique. Bobot lokal suatu *term* i di dalam dokumen j (*tfij*) dapat didefinisikan sebagai,

$$tfi_j = \frac{f_{ij}}{\max_i(f_{ij})} \qquad (1)$$

Dimana f_{ij} adalah jumlah berapa kali term i muncul di dalam dokumen j. Frekuensi tersebut dinormalisasi dengan frekuensi dari *most common term* di dalam dokumen tersebut.
Bobot global dari suatu term i pada pendekatan inverse document frequency (idfi) dapat didefinisikan sebagai,

$$idf_i = \log_2(\frac{n}{df_i}) \qquad (2)$$

Dimana df_i adalah frekuensi dokumen dari term i dan sama dengan jumlah dokumen yang mengandung term i. Log_2 digunakan untuk memperkecil pengaruhnya relative terhadap tf_{ij}. Bobot dari term i di dalam sistem IR (w_{ij}) dihitung menggunakan ukuran tf-idf yangdidefinisikan sebagai berikut [8, 20] :

$$w_{ij} = tf_{ij} \times idf_i \qquad (3)$$

Bobot tertinggi diberikan kepada term yang muncul sering kali dalam dokumen j tetapi jarang dalam dokumen lain.
Salah satu ukuran kemiripan teks yang populer adalah *cosine similarity*. Ukuran ini menghitung nilai cosinus sudut antara dua vektor. Jika terdapat dua vektor dokumen d_j dan query q, serta t term diekstrak dari koleksi dokumen maka nilai cosinus antara d_j dan q didefinisikan sebagai [2] :

$$similarity\ (\vec{d}_j, \vec{q}) = \frac{\vec{d}_j . \vec{q}}{|\vec{d}_j| . |\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}} \qquad (4)$$

3. METODOLOGI

3.1. Metodologi Perancangan Sistem

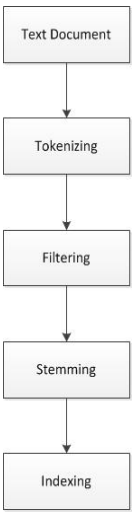
Dalam penelitian ini, 5 dokumen dalam format Microsoft Word (*.docx*) digunakan sebagai dokumen sumber yang informasinya akan di-Retrieve. Berikut adalah nama dokumen dan isi dokumennya masing-masing,

Tabel 1 : Dokumen Sumber

No	Dokumen	Isi Dokumen
D1	BIN.docx	Komandan Komando Pendidikan dan Latihan TNI Angkatan Darat Letnan Jenderal TNI Marciano Norman ditunjuk oleh Presiden Susilo Bambang Yudhoyono
D2	BUMN.docx	Dahlan Iskan didaulat sebagai Menteri Badan Usaha Milik Negara menggantikan Mustafa Abubakar
D3	gedung dpr.docx	Rencana pembangunan gedung baru DPR yang beberapa waktu lalu

		menuai kontroversi
D4	Humanoid.docx	Negeri sakura memang pengusung konsep-konsep robot humanoid tercanggih di Asia
D5	Industri.docx	Industri komunikasi dan kolaborasi enterprise di seluruh Asia Pasifik diprediksi berkembang sangat positif pada tahun 2012

3.2. Metodologi Indexing Teks



Gambar 4 : Metodologi Indexing Text

A. Tokenizing

Tokenizing adalah proses penghilangan tanda baca pada kalimat yang ada dalam dokumen sehingga menghasilkan kata-kata yang berdiri masing-masing.

B. Filtering

Tahap filtering adalah tahap pengambilan kata-kata yang penting dari hasil tokenizing. Tahap filtering ini menggunakan daftar stoplist atau wordlist. Stoplist yaitu penyaringan (filtering) terhadap kata-kata yang tidak layak untuk dijadikan sebagai pembeda atau sebagai kata kunci dalam pencarian dokumen sehingga kata-kata tersebut dapat dihilangkan dari dokumen. Sedangkan wordlist adalah daftar kata yang mungkin digunakan sebagai kata kunci dalam pencarian dokumen, dengan demikian maka tentu jumlah kata yang termasuk dalam wordlist akan lebih banyak daripada stoplist.

C. Stemming

Stemming adalah proses mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen. Dalam penelitian ini, proses stemming menggunakan Algoritma Porter.

Berikut ini adalah algoritma Porter untuk proses Stemming [1],

1) Hapus *Particle*.
2) Hapus Possesive Pronoun
3) Hapus awalan pertama. Jika tidak ada lanjutkan ke langkah 4a, jika ada maka lanjutkan ke langkah 4b.
4) a. Hapus awalan kedua, lanjutkan ke langkah 5a.
   b. Hapus akhiran. Jika tidak ditemukan maka kata tersebut diasumsikan sebagai root word. Jika ditemukan maka lanjutkan ke langkah 5b.
5) a. Hapus akhiran. Kemudian kata akhir diasumsikan sebagai root word
   b. Hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai root word.

Tabel 2 : Aturan untuk Inflectional Particle

Akhiran	Replacement	Additional Condition
-kah	Null	Null
-lah	Null	Null
-pun	Null	Null

Tabel 3 : Aturan untuk Inflectional Possesive Pronoun

Akhiran	Replacement	Additional Condition
-ku	Null	Null

-mu	Null	Null
-nya	Null	Null

Tabel 4 : Aturan untuk First Order Derivational Prefix

Awalan	Replacement	Additional Condition
Meng-	Null	Null
Meny-	S	V…*
Men-	Null	Null
Mem-	P	V…
Mem-	Null	Null
Me-	Null	Null
Peng-	Null	Null
Peny-	S	V…
Pen-	Null	Null
Pem-	P	V…
Pem-	Null	Null
di-	Null	Null
Ter-	Null	Null
Ke-	Null	Null

Tabel 5 : Aturan untuk Second Order Derivational Prefix

Awalan	Replacement	Additional Condition
Ber-	Null	Null
Bel-	Null	*Ajar*
Be-	Null	*k*er*
Per-	Null	Null
Pel-	Null	*Ajar*
Pe-	Null	Null

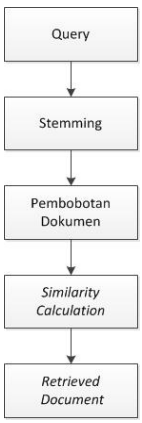
Tabel 6 : Aturan untuk Derrivational Suffix

Akhiran	Replacement	Additional Condition
-kan	Null	Prefix bukan anggota {ke, peng}
-an	Null	Prefix bukan anggota {di, meng, ter}
-i	Null	Prefix bukan anggota {ber, ke, peng}

D. Indexing

Teks dokumen yang telah melalui proses tokenizing, filtering, dan stemming, kemudian di-indeks ke dalam *database*.

3.3. Metodologi Pencarian Teks



Gambar 5 : Metodologi Pencarian Teks

A. Query
Pengguna melakukan pencarian dokumen dengan membuat suatu query pencarian.

B. Stemming
Metode untuk Stemming sama dengan proses Stemming yang ada pada metodologi *indexing text*, yaitu dengan menggunakan Algoritma Porter Stemming.

C. Pembobotan Dokumen
Perhitungan untuk pembobotan dokumen menggunakan **Persamaan 3**, yaitu menggunakan Algoritma TF-IDF.

D. Similarity Calculation
Perhitungan untuk mengukur tingkat kemiripan dokumen (*Similarity Calculation*) menggunakan **Persamaan 4**, yaitu *Cosine Similarity*.

E. Retrieved Document
Dokumen yang telah dihitung tingkat kemiripannya, kemudian disajikan kepada pengguna dalam bentuk perankingan dokumen.

4. HASIL DAN PEMBAHASAN

4.1. Tahap Indexing Teks

Setelah melalui tahapan tokenizing, filtering, dan stemming diperoleh hasil indexing kata tiap dokumen sebagai berikut :

Tabel 7 : Hasil Stemming

No	Dokumen	Hasil Indexing Text
D1	BIN.docx	Komandan komando di latih TN angkat darat letnan jenderal marciano norm tunjuk presiden susilo bambang yudhoyono
D2	BUMN.docx	Dahl Is daulat menteri usaha negara ganti mustafa abubakar
D3	gedung dpr.docx	Rencana bangun DPR ua kontroversi
D4	Humanoid.docx	Neger sakura usung konsep-konsep robot humanoid canggih asia
D5	Industri.docx	Industri komunikasi kolaborasi enterprise asia pasifik prediks kembang positif 2012

4.2. Tahap Pencarian

Jika terdapat Query : “Industri Komunikasi”, maka dengan menggunakan algoritma TF-IDF pada Persamaan 3, dapat dianalisa perhitungan untuk mencari bobot tiap term i pada dokumen j, yakni sebagai berikut :

Tabel 8 : Pembobotan Dokumen

	tf							idf	Wdt=tf.idf					
Term	Q	D1	D2	D3	D4	D5	df	log(n/df)	Q	D1	D2	D3	D4	D5
komandan	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
komando	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
di	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
latih	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
TN	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
angkat	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
darat	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
letnan	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
jenderal	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
marciano	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
norm	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
tunjuk	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
presiden	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
susilo	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
bambang	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
yudhoyono	0	1	0	0	0	0	1	0.69897	0	0.69897	0	0	0	0
Dahl	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
Is	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
daulat	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
menteri	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
usaha	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
negara	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
ganti	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
mustafa	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0

 ISBN 979 - 26 - 0255 - 0

abubakar	0	0	1	0	0	0	1	0.69897	0	0	0.69897	0	0	0
Rencana	0	0	0	1	0	0	1	0.69897	0	0	0	0.69897	0	0
bangun	0	0	0	1	0	0	1	0.69897	0	0	0	0.69897	0	0
DPR	0	0	0	1	0	0	1	0.69897	0	0	0	0.69897	0	0
ua	0	0	0	1	0	0	1	0.69897	0	0	0	0.69897	0	0
kontroversi	0	0	0	1	0	0	1	0.69897	0	0	0	0.69897	0	0
Neger	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
sakura	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
usung	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
konsep-konsep	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
robot	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
humanoid	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
canggih	0	0	0	0	1	0	1	0.69897	0	0	0	0	0.69897	0
asia	0	0	0	0	1	1	2	0.39794	0	0	0	0	0.39794	0.39794
industri	1	0	0	0	0	1	1	0.69897	0.69897	0	0	0	0	0.69897
komunikasi	1	0	0	0	0	1	1	0.69897	0.69897	0	0	0	0	0.69897
kolaborasi	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
enterprise	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
pasifik	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
prediks	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
kembang	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
positif	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
2012	0	0	0	0	0	1	1	0.69897	0	0	0	0	0	0.69897
								TOTAL	1.39794	11.18352	6.29073	3.49485	5.29073	6.68867

Dari data di atas dapat diketahui bobot masing-masing dokumen adalah sebagai berikut :
Bobot untuk D1 : 0 + 0 = 0
Bobot untuk D2 : 0 + 0 = 0
Bobot untuk D3 : 0 + 0 = 0
Bobot untuk D4 : 0 + 0 = 0
Bobot untuk D5 : 0.69897 + 0.69897 = 1.39794
Setelah bobot dari masing-masing dokumen ditemukan, lalu dicari nilai *Cosine Similarity* menggunakan **persamaan 4**, untuk menentukan tingkat kesamaan dokumen yang ada di basis data dengan Query yang ditentukan,

Tabel 9. Perhitungan Vector Space Model

Term	Query	D1	D2	D3	D4	D5	Q*D1	Q*D2	Q*D3	Q*D4	Q*D5
komandan	0	0.48856	0	0	0	0	0	0	0	0	0
komando	0	0.48856	0	0	0	0	0	0	0	0	0
di	0	0.48856	0	0	0	0	0	0	0	0	0
latih	0	0.48856	0	0	0	0	0	0	0	0	0
TN	0	0.48856	0	0	0	0	0	0	0	0	0
angkat	0	0.48856	0	0	0	0	0	0	0	0	0
darat	0	0.48856	0	0	0	0	0	0	0	0	0
letnan	0	0.48856	0	0	0	0	0	0	0	0	0
jenderal	0	0.48856	0	0	0	0	0	0	0	0	0
marciano	0	0.48856	0	0	0	0	0	0	0	0	0
norm	0	0.48856	0	0	0	0	0	0	0	0	0
tunjuk	0	0.48856	0	0	0	0	0	0	0	0	0
presiden	0	0.48856	0	0	0	0	0	0	0	0	0
susilo	0	0.48856	0	0	0	0	0	0	0	0	0
bambang	0	0.48856	0	0	0	0	0	0	0	0	0
yudhoyono	0	0.48856	0	0	0	0	0	0	0	0	0
Dahl	0	0	0.48856	0	0	0	0	0	0	0	0
Is	0	0	0.48856	0	0	0	0	0	0	0	0
daulat	0	0	0.48856	0	0	0	0	0	0	0	0
menteri	0	0	0.48856	0	0	0	0	0	0	0	0
usaha	0	0	0.48856	0	0	0	0	0	0	0	0
negara	0	0	0.48856	0	0	0	0	0	0	0	0
ganti	0	0	0.48856	0	0	0	0	0	0	0	0
mustafa	0	0	0.48856	0	0	0	0	0	0	0	0
abubakar	0	0	0.48856	0	0	0	0	0	0	0	0
Rencana	0	0	0	0.48856	0	0	0	0	0	0	0
bangun	0	0	0	0.48856	0	0	0	0	0	0	0
DPR	0	0	0	0.48856	0	0	0	0	0	0	0
ua	0	0	0	0.48856	0	0	0	0	0	0	0
kontroversi	0	0	0	0.48856	0	0	0	0	0	0	0
Neger	0	0	0	0	0.48856	0	0	0	0	0	0
usung	0	0	0	0	0.48856	0	0	0	0	0	0
konsep-konsep	0	0	0	0	0.48856	0	0	0	0	0	0
robot	0	0	0	0	0.48856	0	0	0	0	0	0
humanoid	0	0	0	0	0.48856	0	0	0	0	0	0
canggih	0	0	0	0	0.48856	0	0	0	0	0	0
asia	0	0	0	0	0.15836	0.15836	0	0	0	0	0
industri	0.48856	0	0	0	0	0.48856	0	0	0	0	0.23869
komunikasi	0.48856	0	0	0	0	0.48856	0	0	0	0	0.23869
kolaborasi	0	0	0	0	0	0.48856	0	0	0	0	0
enterprise	0	0	0	0	0	0.48856	0	0	0	0	0
pasifik	0	0	0	0	0	0.48856	0	0	0	0	0
prediks	0	0	0	0	0	0.48856	0	0	0	0	0
kembang	0	0	0	0	0	0.48856	0	0	0	0	0
positif	0	0	0	0	0	0.48856	0	0	0	0	0
2012	0	0	0	0	0	0.48856	0	0	0	0	0
	Sqrt(Q)			Sqrt(D_i)					Sqrt($Q*D_i$)		
	0.98849	2.79588	2.09691	1.56295	1.89163	2.13434	0	0	0	0	0.69093

Untuk menghitung nilai *Cosinus* sudut antara *vector query* dengan tiap dokumen dapat menggunakan rumus :

$Cosine(D_i)=Sqrt(Q*D_i) / (Sqrt(Q)*Sqrt(D_i))$

Dokumen 1 (D1)

$Cosine(D_1) = 0 / (0.98849*2.79588) = 0$

Dokumen 2 (D2)

$Cosine(D_2) = 0 / (0.98849*2.09691) = 0$

Dokumen 3 (D3)

$Cosine(D_3) = 0 / (0.98849*1.56295) = 0$

Dokumen 4 (D4)

$Cosine(D_4) = 0 / (0.98849*1.89163) = 0$

Dokumen 5 (D5)

$Cosine(D_5) = 0.69093 / (0.98849*2.13434) =$ **0.32749**

Dari hasil perhitungan di atas, dapat diranking tingkat kemiripan dokumen dengan query "Industri Komunikasi", yaitu sebagai berikut :

D5, D1, D2, D3, D4.

Dapat diketahui bahwa, dokumen 5 (D5) memiliki tingkat kemiripan yang tinggi dengan query.

5. PENUTUP

Dari hasil penelitian, dapat disimpulkan bahwa Sistem Temu Balik Informasi menggunakan Vector Space Model dapat digunakan sebagai mesin pencarian untuk pencarian dokumen teks berbahasa indonesia.

DAFTAR PUSTAKA

[1] Agusta, Ledy. Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief dan Adriani untuk Stemming Dokumen Teks Bahasa Indonesia. Pada Konferensi Nasional Sistem dan Informatika 2009. No Jurnal : KNS&I09-036.
[2] Cios, Krzysztof J. Etc. (2007) Data Mining A Knowledge Discovery Approach, Springer.
[3] Lee D.L. (1997). Document Ranking and the Vector-Space Model. IEEE March-April 1997.
[4] Lu, Guojun. Multimedia Database Management Systems. ARTECH HOUSE Inc. Canton Street : 1999.
[5] Murad, Azmi MA., Martin, Trevor. (2007) Word Similarity for Document Gouping using Soft Computing. IJCSNS International Journal of Computer Science and Network Security, Vol.7 No.8, August 2007, pp. 20- 27
[6] Polettini, Nicola (2004) The Vector Space Model in Information Retrieval – Term Weighting Problem
[7] Salton, Gerard (1983) Introduction to Modern Information Retrieval, McGraw Hil