dinamika teknik implementasi SE 2011.PDF

by Fatkhul Amin

Submission date: 29-May-2020 07:15AM (UTC+0700)
Submission ID: 1333741702
File name: dinamika teknik implementasi SE 2011.PDF (1.77M)
Word count: 3066
Character count: 17477

IMPLEMENTASI *SEARCH ENGINE* (MESIN PENCARI) MENGGUNAKAN METODE *VECTOR SPACE MODEL*

Fatkhul Amin
Dosen Fakultas Teknologi Informasi Universitas Stikubank Semarang

DINAMIKA TEKNIK
Vol. V, No. 1
Januari 2011
Hal 45 - 58

Abstract

Growth of Machine Searcher in this time yield high recall and storey;level of precision low. Recall which is high to be interpreted that document which yielded in penelusuran of document many, while low accuration storey level can be interpreted that expected document can be found a few/little or lower. Solution to overcome the problem of above by making meeting system return information use method of Vector Space Model (VSM). Method of VSM selected because way of model efficient, easy to in representasi and earn implementation [at] document-matching.

Key Word : *Vector Space Model, Recall, Search Machine*

Pendahuluan

Informasi saat ini sangat mudah didapatkan oleh setiap orang dimanapun berada. Informasi bisa mudah didapatkan karena perkembangan teknologi informasi yang semakin cepat dan pengetahuan setiap orang yang terus bertambah. Teknologi informasi khususnya internet sangat mendukung terjadinya pertukaran informasi dengan sangat cepat. Internet menjadi media informasi dan telekomunikasi yang telah dimanfaatkan banyak orang dengan banyak kepentingan. Internet menciptakan banyak informasi dan karena kemudahan dalam akses internet inilah banyak orang menggunakan media online dalam kegiatannya sehari-hari. Setiap orang membutuhkan informasi dan mereka bisa mencari informasi dengan menggunakan mesin pencari yang ada didunia maya.

Informasi yang berkualitas dipengaruhi oleh relevansi, keakuratan dan tepat waktu [Al Bahra:2005]. Pencarian informasi saat ini dilakukan dengan menggunakan mesin pencari yang ada pada situs layanan di dunia maya. Mesin pencari yang sudah ada dan banyak digunakan saat ini memberikan hasil informasi yang sangat banyak, sehingga diperlukan waktu untuk memilah-milah informasi yang dibutuhkan. Informasi yang didapatkan bila terlalu banyak akan menyulitkan user karena user juga mendapatkan informasi yang tidak berguna . Banyaknya informasi ini

menjadikan waktu yang ada akan terbuang atau tidak efisien karena terlalu banyaknya informasi menyebabkan pekerjaan-pekerjaan yang tidak efektif dilakukan yaitu : mencari dan memilih.

Penelitian terkait dengan metode vector space model juga dilakukan antara lain Dr. Khalaf Khatatneh, M. Wedyan, DR. Mohamed Alham, DR Basem Alrifai (2005) dalam publikasinya yang berjudul "*Using new Data Structure to Implement Documents Vectors in Vector Space Model in Information Retrieval System*" Bagaimana menggunakan tabel terstruktur dan vector space model untuk lebih menghemat ruang untuk file dokumen yang sebelumnya memerlukan space besar dalam sistem temu kembali informasi. Aplikasinya dilakukan dengan menggunakan tabel dimana pada baris pertama digunakan kata kunci dan baris kedua digunakan pembobotan dari setiap kata kunci.

Penelitian tentang "Perluasan Vektor pada Metode Search Vector Space" dilakukan oleh kristoper David Harjono (2005). Tentang penambahan kumpulan term yang relevan kedalam vektor dokumen dan/atau vektor *query*. Tujuan dari perluasan vektor adalah untuk meningkatkan recall dari hasil pencarian. Pada penelitian initerm yang dianggap relevan adalah term yang memiliki hubungan sinonim dengan term asli. Perluasan dilakukan dengan menggunakan bagian *noun* dari database WordNet sebagai sumber data sinonim.

Penelitian tentang algoritma porter stemmer for bahasa Indonesia untuk pre-processing text mining berbasis metode market basket analysis pernah dilakukan oleh Gregorius S. Budhi, Ibnu Gunawan dan Ferry Yuwono (2006), peneliti mengajukan penggunaan algoritma Porter Stemmer for Bahasa Indonesia[8], untuk proses Stemmer pada langkah pre-processing yang merubah sebuah teks dalam bahasa Indonesia menjadi bentuk Compact Transaction. Compact Transaction digunakan sebagai masukan untuk proses *Keyword-Based Association Analysis*, sebuah metode Text Mining yang dikembangkan dari metode *Market Basket Analysis*, digunakan untuk membentuk rule-rule asosiasi dari data teks. Pengujian dilakukan mengggunakan sample data teks dalam bahasa Indonesia berupa Abstrak Tugas Akhir mahasiswa Universitas Kristen Petra Surabaya. Dari hasil pengujian dapat

disimpulkan bahwa algoritma Porter Stemmer for Bahasa Indonesia dapat digunakan pada proses Stemmer saat merubah sebuah data teks dalam bahasa Indonesia menjadi bentuk *Compact Transaction*. Hasil dari proses ini tidak selalu benar sehingga masih diperlukan pemeriksaan manual.

Penelitian tentang Sistem temu kembali informasi juga dilakukan oleh Herni Februariyanti (2010), Peneliti membangun klaster dokumen dengan menggunakan Algoritma *Hierarchical Agglomerative Clustering* untuk sistem temu kembali informasi berbahasa Indonesia. Algoritma klustering digunakan untuk mengintegrasikan dokumen-dokumen dengan topik yang berbeda. Perkembangan penelusuran informasi saat ini menghasilkan *recall* yang tinggi dan tingkat keakuratan yang rendah. *Recall* yang tinggi diartikan bahwa dokumen yang dihasilkan dalam penelusuran dokumen adalah banyak, sedangkan tingkat akurasi rendah dapat diartikan bahwa dokumen yang diharapkan dapat ditemukan sedikit atau rendah.

Solusi untuk mengatasi masalah di atas adalah dengan membuat sistem temu kembali informasi menggunakan metode *Vector Space Model (VSM)*. Metode VSM dipilih karena cara kerja model ini efisien, mudah dalam representasi dan dapat diimplementasikan pada *document-matching*.

Identifikasi Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang dapat dirumuskan adalah bagaimana membuat aplikasi dengan algoritma yang efektif dalam pencarian dokumen dengan metode *vector space model* pada dokumen teks berbahasa Indonesia, sehingga pengguna mudah mencari dokumen yang diinginkan. Dalam penelitian ini ada beberapa pembatasan masalah yang dilakukan, yaitu:

a. Dokumen yang digunakan adalah dokumen teks berbahasa Indonesia.
b. Penelitian ini digunakan metode *Vector Space Model*

Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini:

a. Dapat digunakan sebagai alat bantu untuk pencarian dokumen teks.
b. Menghemat waktu pencarian informasi untuk mendapatkan dokumen yang diinginkan.

Tujuan Penelitian

a. Mengembangkan metode pencarian cepat dengan metode *Vector Space Model.*
b. Menguji kinerja temu kembali menggunakan *recall* dan *precision*

Penelitian terdahulu

Penelitian selama 20 tahun menunjukkan bahwa sistem yang berbasis indeks teks dengan menggunakan pembobotan sebuah dokumen yang tepat menghasilkan pencarian lebih tepat. Menurut Salton (1988) hasilnya sangat tergantung dari efektifitas sistem pembobotan dokumen. Penelitian terkait telah dilakukan antara lain Yue W, dkk (2007), dalam penelitiannya mengusulkan algoritma sistem temu kembali informasi *(information retrieval)* berbasis *query expansion* dan klasifikasi. Algoritma tersebut diinduksi dari query yang pendek dan metode pencarian informasi tradisional (traditional retrieval information method) yang menghasilkan presisi yang rendah walaupun tingkat *recall* cukup tinggi. Penelitian ini berusaha untuk mendapatkan lebih banyak dokumen yang relevan dengan *query expansion* dan klasifikasi dokumen.

Penelitian dengan topik *Document Ranking andthe Vector-Space Model*, Dik L. Lee (*Hong Kong University of Science and Technology)*, Huei Chuang, dan Kent Seamonts. *(1997).* Banyaknya informasi berupa teks, mempunyai masalah dengan pengambilan kembali informasi yang ada. Diperlukan teknik pengambilan informasi yang efektif dan efisien mengingat jumlah informasi yang besar. Kebanyakan pengambilan informasi bergantung kepada kata kunci pengindeksan, namun demikian kata kunci atau indeks saja tidak cukup menangkap isi dokumen. Saat ini pengindeksan menggunakan kata kunci banyak digunakan dalam system temu kembali informasi, karena dinilai paling layak. Peneliti mengulas dua permasalaha

yang ada yaitu bagaimana mengidentifikasi istilah indeks dan bagaimana cara mengetahui dokumen sesuai dengan query.

Sistem Temu Kembali Informasi

Sistem temu kembali informasi merupakan bagian dari ilmu komputer yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. Proses dalam sistem temu kembali dapat digambarkan sebagai sebuah proses untuk mendapatkan dokumen yang relevan dari koleksi dokumen melalaui pencarian *query* yang diimputkan user. Salton menjelaskan bahwa sistem temu kembali informasi bertujuan untuk menjembatani kebutuhan informasi *user* dengan sumber informasi yang tersedia dalam situasi seperti dikemukakan 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.

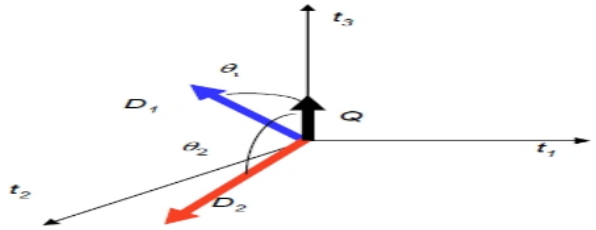
Text Mining

Text mining adalah salah satu bidang khusus dari *data mining*. Sesuai buku *The Text Mining Handbook*, *text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Salah satu elemen kunci dari *text mining* adalah kumpulan dokumen yang berbasis teks. Pada prakteknya, *text mining* ditujukan untuk menemukan pola dari sekumpulan dokumen

yang jumlahnya sangat besar dan bisa mencapai jumlah ribuan bahkan sampai jutaan. Koleksi dokumen bisa statis, dimana dokumen tidak berubah, atau dinamis, dimana dokumen selalu di*update* sepanjang waktu.

Vector Space Model

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan *(smilarity)* 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*.



Gambar 1. Ilustrasi Vector Space Model

dimana
ti = Kata di database
Di = Dokumen
Q = Kata Kunci

Cara kerja dari *vector space model* adalah dengan menghitung nilai cosines sudut dari dua vector, yaitu vektor kata kunci terhadap vektor tiap dokumen. Perhitungan vektor space model menggunakan persamaan (1),(2) dan (3)

$$\boldsymbol{Cosine\ \theta_{D_i} = Sim(Q, D_i)} \quad (1)$$

dimana
Q = query (kata kunci)
Di = dokumen ke-i

$$\boldsymbol{Sim(Q, D_i) = \frac{\sum_j w_{ij}\, w_{qj}}{\sqrt{\sum_j w_{ij}^2}\sqrt{\sum_i w_{qj}^2}}} \quad (2)$$

dimana
Di = dokumen ke-i

Q = query (kata kunci)
J = Kata diseluruh dokumen

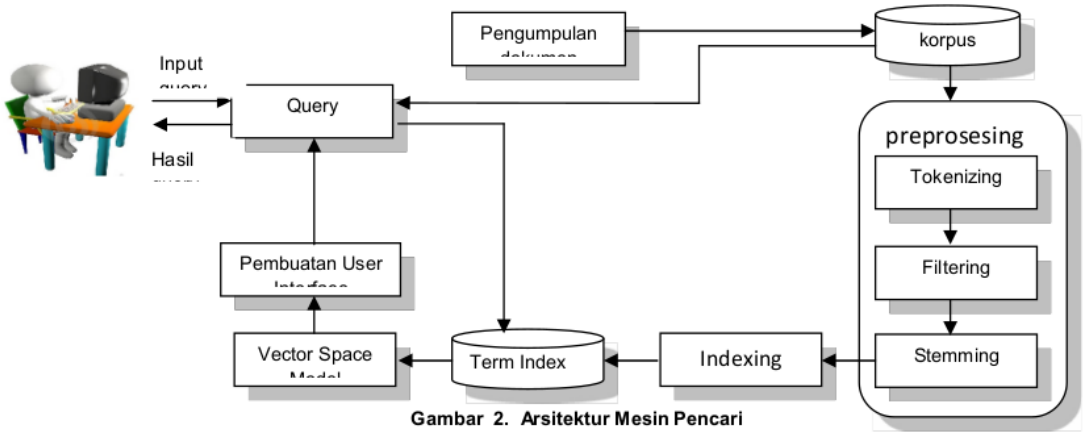
$$Cosine\ \theta_{D_i} = \frac{Q \cdot D_i}{|Q| * |D_i|} \quad (3)$$

dimana
Di = dokumen ke-i
Q = query (kata kunci)
|Q| = Vektor Q
|Di| = Vektor Di

Arsitektur Sistem Temu Kembali Informasi

Sistem temu kembali informasi menggunakan metode *Vector Space Model* sebagai suatu sistem memiliki beberapa proses (modul) yang membangun system secara keseluruhan. Modul system temu kembali informasi terdiri dari : modul pengumpulan dokumen, modul tokenisasi (*tokenizing*), modul pembuangan *stopword* (*filtering*), modul Pengubahan kata dasar (*stemming*), modul Pengindeksan kata (*indexing*), modul *Vector Space Model* (*term similarity*) dan modul pembuatan *user interface.*

Arsitektur sistem temu kembali informasi bisa dilihat pada gambar 1.



Gambar 2. Arsitektur Mesin Pencari

Implementasi Vector Space Model

Modul sistem temu kembali informasi terdiri dari : modul pengumpulan dokumen, modul tokenisasi (*tokenizing*), modul pembuangan *stopword* (*filtering*), modul Pengubahan kata dasar (*stemming*), modul Pengindeksan kata (*indexing*), dan modul *Vector Space Model* (*term similarity*). Modul Pengumpulan Dokumen.

Proses pengumpulan dokumen-dokumen yang dipilih untuk disimpan dalam korpus. Dokumen-dokumen yang dipilih adalah dokumen teks berbahasa Indonesia. Pada penelitian ini digunakan 500 abstraksi skripsi mahasiswa Universitas Indonesia.

Modul Tokenisasi (tokenizing). *Tokenizing* adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Umumnya setiap kata yang teridentifikasi atau terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses ini menggunakan spasi pada dokumen untuk melakukan pemisahan kata. Modul Pembuangan *stopword* (*filtering*). Tahap *filtering* adalah proses pembuangan term yang tidak memiliki arti atau tidak relevan. Term yang diperoleh pada tahap tokenisasi dicek dalam suatu daftar *stopword*, jika term masuk dalam daftar stopword maka term tidak akan diproses lebih lanjut, tapi jika term tidak termasuk dalam daftar *stopword* maka term akan diproses lebih lanjut. Contoh *stopwords* adalah "yang", "dan", "di", "dari" dan seterusnya. Daftar *stopword* bisa dilihat pada lampiran 1.

Modul Pengubahan Kata Dasar (*Stemming*). Proses *stemming* adalah tahap mencari kata dasar (*root)* dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam bentuk kata dasar. Proses stemming pada penelitian ini menggunakan Algoritma Porter Stemmer for Bahasa Indonesia [Tala, 2003]. Stemming digunakan untuk untuk mereduksi bentuk term untuk menghindari ketidakcocokan sehingga dapat mengurangi recall.

Modul Indexing (*inverted index*). Proses Indexing adalah tahap pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Seluruh dokumen dalam koleksi disimpan dalam satu file dengan format tertentu sehingga antara dokumen satu dengan dokumen yang lain bisa dibedakan. Setelah kata telah dikembalikan dalam bentuk kata dasar, kemudian disimpan dalam tabel basis data. Proses indexing menghasilkan *database index* .Metode yang digunakan dalam penelitian ini adalah *inverted index*. Modul *Vector Space Model (similarity analysis)*. Hasil indexing selanjutnya dihitung tingkat kemiripannya dengan *query* menggunakan metode *vector space model*.

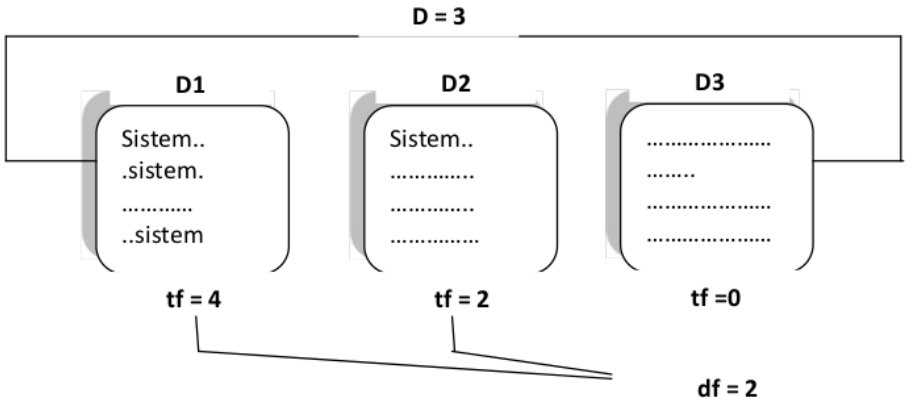
Modul *Vector Space Model (similarity analysis)*

Tahapan proses analisa *VSM* adalah sebagai berikut; Menghitung bobot dokumen dengan *tf-idf,* Menghitung jarak tiap query dan dokumen, Menghitung dot produk, Menghitung similaritas, dan Membuat ranking. Hasil indexing selanjutnya dihitung tingkat kemiripannya dengan *quer*y menggunakan metode *vector space model*. Tahapan implementasi *vector space model* agar lebih jelas dibuat contoh *query* (Q) dan dokumen (D = 3), seperti di bawah ini:

Contoh :

Query (Q) = Sistem
Dokumen 1 (D1) = Sistem adalah kumpulan elemen
Dokumen 2 (D2) = adalah kumpulan elemen yang saling berinteraksi
Dokumen 3 (D3) = Sistem berinteraksi untuk mencapai tujuan

Melalui proses tokenizing selanjutnya masuk pada proses *filtering (stopword removal)*, maka kata adalah pada D1, kata adalah dan yang pada D2, serta kata untuk pada D3 dihapus.



Gambar 3. Ilustrasi Algoritma Text Mining

Keterangan :

D1, D2, D3 = Dokumen
tf = banyak kata yang dicari pada sebuah dokumen
D = total dokumen
df =Banyak dokumen yang mengandung kata yang dicari

Selanjutnya, kumpulan kata dasar yang telah terpilih dilakukan proses pembobotan dokumen melalui beberapa perhitungan di bawah ini.

Menghitung bobot dokumen dengan *tf-idf*

Tabel 1. Tabel perhitungan *tf*

Token	tf				df
	Q	D1	D2	D3	
sistem	1	1	0	1	2
kumpul	0	1	1	0	2
elemen	0	1	1	0	2
saling	0	0	1	0	1
interaksi	0	0	1	1	2
capai	0	0	0	1	1
tuju	0	0	0	1	1

Setelah hasil perhitungan *tf* didapatkan, langkah selanjutnya dilakukan perhitungan *inverse document frequency (idf)* tiap token untuk menghitung bobot token.

Rumus *idf*

$$Idf = log\ (N/df)$$

Keterangan :

N = jumlah dokumen dalam korpus

Melalui perhitungan *idf* didapatkan tabel hasil perhitungan *idf.*

Tabel 2. Tabel perhitungan *idf*

Token	tf				df	D/df	IDF log(D/df
	Q	D1	D2	D3			
sistem	1	1	0	1	2	1.5	0.176
kumpul	0	1	1	0	2	1.5	0.176
elemen	0	1	1	0	2	1.5	0.176
saling	0	0	1	0	1	3	0.477
interaksi	0	0	1	1	2	1.5	0.176
capai	0	0	0	1	1	3	0.477
tuju	0	0	0	1	1	3	0.477

Selanjutnya, setelah nilai *tf* dan *idf* telah didapatkan, kemudian dimasukkan dalam perhitungan *tf-idf weighting* untuk menghitung bobot hubungan suatu token di dalam dokumen [Robertson,2005].

Rumus *tf-idf weighting*

$$wf_{(t,d)} = tf_{t,d} * idf_t$$

Keterangan :
d = dokumen ke-d
t = kata ke-t dari kata kunci
W= bobot dokumen ke-d terhadap kata ke-t

Hasil perhitungan tf-idf weighting bisa diliha pada tabel 3.

Tabel 3. Tabel perhitungan *tf-idf*

Token	tf				df	D/df	IDF log(D/df	W			
	Q	D1	D2	D3				Q	D1	D2	D3
sistem	1	1	0	1	2	1.5	0.176	0.176	0.176	0	0.176
kumpul	0	1	1	0	2	1.5	0.176	0	0.176	0.176	0
elemen	0	1	1	0	2	1.5	0.176	0	0.176	0.176	0
saling	0	0	1	0	1	3	0.477	0	0	0.477	0
interaksi	0	0	1	1	2	1.5	0.176	0	0	0.176	0.176
capai	0	0	0	1	1	3	0.477	0	0	0	0.477
tuju	0	0	0	1	1	3	0.477	0	0	0	0.477

3.3.4.2 menghitung jarak tiap dokumen dan query

$$|Q| = \sqrt{\sum_i W_{q,j}^2}$$

Perhitungan vector dari query

$$Sqrt(Q) = Sqrt(\sum_{j=1}^{n} Q_j^2)$$

Keterangan
j = kata di database

$$Sqrt(Q) = Sqrt(\sum_{j=1}^{n} Q_j^2)$$
$$= \sqrt{(0.176)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$$
$$= \sqrt{(0.301 + 0 + 0 + 0 + 0 + 0 + 0)}$$
$$= \sqrt{(0.301)}$$
$$= \mathbf{0.549}$$

$$|D_i| = \sqrt{\sum_i W_{i,j}^2}$$

Perhitungan vector dari Dokumen

$$Sqrt(D_i) = Sqrt(\sum_{j=1}^{n} D_{i,j}^2)$$

Keterangan
j = kata di database

Dokumen 1 (D1)

$$Sqrt(D_1) = Sqrt(\sum_{j=1}^{n} D_{1,j}^2)$$
$$= \sqrt{(0.176)^2 + (0.176)^2 + (0.176)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2}$$
$$= \sqrt{(0.301 + 0.301 + 0.301 + 0 + 0 + 0 + 0)}$$
$$= \sqrt{(0.903)}$$
$$= \mathbf{0.95}$$

Dokumen 2 (D2)

$$Sqrt(D_2) = Sqrt(\sum_{j=1}^{n} D_{2,j}^2)$$

$$= \sqrt{(0)^2 + (0.176)^2 + (0.176)^2 + (0.477)^2 + (0.176)^2 + (0)^2 + (0)^2}$$

$$= \sqrt{(0 + 0.301 + 0.301 + 0.288 + 0.301 + 0 + 0)}$$

$$= \sqrt{(1.131)}$$

$$= \mathbf{1.063}$$

Dokumen 3 (D3)

$$Sqrt(D_3) = Sqrt(\sum_{j=1}^{n} D_{3,j}^2)$$

$$= \sqrt{(0.176)^2 + (0)^2 + (0)^2 + (0)^2 + (0.176)^2 + (0.477)^2 + (0.477)^2}$$

$$= \sqrt{(0.301 + 0 + 0 + 0 + 0.301 + 0.288 + 0.288)}$$

$$= \sqrt{(1.058)}$$

$$= \mathbf{1.029}$$

Tabel 4. Tabel hasil perhitungan jarak dokumen dan query

Token	Q^2	$D1^2$	$D2^2$	$D3^2$
sistem	0.301	0.301	0	0.301
kumpul	0	0.301	0.301	0
elemen	0	0.301	0.301	0
saling	0	0	0.228	0
interaksi	0	0	0.301	0.301
capai	0	0	0	0.228
tuju	0	0	0	0.228
	Sqrt(Q)	**Sqrt(Di)**		
	0.549	0.95	1.063	1.029

3.3.4.3 menghitung dot produk

Selanjutnya setelah jarak dari dokumen dan query didapatkan, dilakukan perhitungan dot produk.

$$Sum(Q * D_i) = \sum_{j=1}^{n} Q_j \; D_{i,j} \qquad (4)$$

Keterangan
j = kata di database

Dokumen 1

$$Sum(Q * D_1) = \sum_{j=1}^{n} Q_j \; D_{1,j}$$

$$= 0.091 + 0.091 + 0.091 + 0 + 0 + 0 + 0$$

$$= \mathbf{0.272}$$

Dokumen 2

$$Sum(Q * D_1) = \sum_{j=1}^{n} Q_j \, D_{2,j}$$
$$= 0 + 0.091 + 0.091 + 0.069 + 0.091 + 0 + 0$$
$$= \mathbf{0.340}$$

Dokumen 3

$$Sum(Q * D_1) = \sum_{j=1}^{n} Q_j \, D_{3,j}$$
$$= 0.091 + 0 + 0 + 0 + 0.091 + 0.069 + 0.069$$
$$= \mathbf{0.228}$$

Tabel 5. Tabel perhitungan Vector Space Model

Token	Q^2	$D1^2$	$D2^2$	$D3^2$	Q*D1	Q*D2	Q*D3
sistem	0.301	0.301	0	0.301	0.091	0	0.091
kumpul	0	0.301	0.301	0	0.091	0.091	0
elemen	0	0.301	0.301	0	0.091	0.091	0
saling	0	0	0.228	0	0	0.069	0
interaksi	0	0	0.301	0.301	0	0.091	0.091
capai	0	0	0	0.228	0	0	0.069
tuju	0	0	0	0.228	0	0	0.069
	Sqrt(Q)	**Sqrt(Di)**			**Sum(Q* Di)**		
	0.549	0.95	1.063	1.029	0.272	0.340	0.228

Menghitung similaritas

Langkah selanjutnya adalah menghitung nilai Cosinus sudut antara vector kata kunci dengan tiap dokumen dengan rumus :

$$\boldsymbol{Cosine\ \theta_{D_i} = \frac{Q \cdot D_i}{|Q| * |D_i|}}$$

$$Cosine\ \theta_{D_1} = \frac{Q \cdot D_1}{|Q| * |D_1|} = Cosine\ \theta_{D_1} = \frac{0.272}{0.549 * 0.95} = \mathbf{0.522}$$

$$Cosine\ \theta_{D_2} = \frac{Q \cdot D_2}{|Q| * |D_2|} = Cosine\ \theta_{D_2} = \frac{0.34}{0.549 * 1.063} = \mathbf{0.583}$$

$$Cosine\ \theta_{D_3} = \frac{Q \cdot D_3}{|Q| * |D_3|} = Cosine\ \theta_{D_1} = \frac{0.288}{0.549 * 1.029} = \mathbf{0.404}$$

Membuat ranking

Dari Analisa Vector Space Model diperoleh hasil untuk ketiga dokumen di atas adalah sebagai berikut.

Tabel 6. Hasil perhitungan Vector Space Model

	D1	D2	D3
Cosine	0.522	**0.583**	0.404
	Rank 2	Rank 1	Rank 3

Hasil perhitungan Cosine diketahui bahwa Dokumen 2 (D2) memiliki tingkat similaritas tertinggi kemudian disusul dengan D1 dan D2.

Kesimpulan

Berdasarkan implementasi contoh kasus keyword atau kata kunci (query) sistem dengan 3 dokumen yang ada, vector space model menghasilkan Dokumen 2 (D2) sebagai dokumen yang paling mendekati kemiripannya atau tingkat kedekatannya dengan query diikuti dengan Dokumen 1 (D1) dan Dokumen 3 (D3). Recall yang dihasilkan rendah dan presisi yang dihasilkan tinggi, artinya dokumen yang diharapkan muncul dengan tingkat akurasi tinggi dapat ditemukan dengan tepat dan dokumen yang dihasilkan sedikit.

Daftar Pustaka

Chang, P. 2010. *Deriving a Categorical Vector Space Model for Web Page Recommendations Based on Wikipedia's Content.* ASIS&T '10: Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47. USA. University of Hawaii.

Erk, Katrin; & Pad´o, Sebastian. 2008. *A Structured Vector Space Model forWord Meaning in Context.* EMNLP '08**:** Proceedings of the Conference on Empirical Methods in Natural Language Processing. USA. University of Texas; & Stanford University.

Pressman R, 1997, Software Engineering, Mc Graw Hill, USA.

Salton, G., 1989, *Automatic Text Processing, The Transformation, Analysis, and Retrieval of information by computer*, Addison – Wesly Publishing Company, Inc. All rights reserved.

Salton, G., and Buckley, 1988, *Term Weigting Approaches in Automatic Text Retrieval*, Department of Computer Science, Ithaca, NY 14853, USA. Cornell University.

dinamika teknik implementasi SE 2011.PDF

ORIGINALITY REPORT

19%	16%	8%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.uinsu.ac.id Internet Source	1%
2	Yoel Panjaitan, Muhammad Ihsan Zul, Ibnu Surya. "Sistem Pemilah Topik Diskusi pada Forum Diskusi Mahasiswa PCR Berbasis Web Menggunakan Algoritma KNN", Jurnal Sistem dan Teknologi Informasi (JUSTIN), 2019 Publication	1%
3	unisbank.ac.id Internet Source	1%
4	sistemtemukembaliinformasi.blogspot.com Internet Source	1%
5	text-id.123dok.com Internet Source	1%
6	vdokumen.com Internet Source	1%
7	Robbi Rahim, Nuning Kurniasih, Muhammad Dedi Irawan, Yustria Handika Siregar et al. "Latent Semantic Indexing for Indonesian Text	1%

	Similarity", International Journal of Engineering & Technology, 2018 Publication	
8	mafiadoc.com Internet Source	1%
9	Martin Martin, Lala Nilawati. "Recall dan Precision Pada Sistem Temu Kembali Informasi Online Public Access Catalogue (OPAC) di Perpustakaan", Paradigma - Jurnal Komputer dan Informatika, 2019 Publication	1%
10	lib.ui.ac.id Internet Source	1%
11	Submitted to Universitas 17 Agustus 1945 Surabaya Student Paper	1%
12	anggaradana.blogspot.com Internet Source	1%
13	nusamandiri.ac.id Internet Source	1%
14	www.acsij.org Internet Source	1%
15	ojs.stmikpringsewu.ac.id Internet Source	<1%
16	eprints.umm.ac.id	

	Internet Source	<1%
17	Submitted to Universitas Brawijaya Student Paper	<1%
18	Submitted to UIN Syarif Hidayatullah Jakarta Student Paper	<1%
19	Submitted to Universitas Dian Nuswantoro Student Paper	<1%
20	edoc.site Internet Source	<1%
21	www.jatit.org Internet Source	<1%
22	www2.hawaii.edu Internet Source	<1%
23	eprints.undip.ac.id Internet Source	<1%
24	Submitted to UIN Sunan Gunung DJati Bandung Student Paper	<1%
25	www.vedcmalang.com Internet Source	<1%
26	www.coursehero.com Internet Source	<1%
27	repository.unisba.ac.id:8080 Internet Source	<1%

28	www.bingkaiberita.com Internet Source	<1%
29	Blanco, Eduardo, and Dan Moldovan. "Composition of semantic relations : Theoretical framework and case study", ACM Transactions on Speech and Language Processing, 2013. Publication	<1%
30	Submitted to Forum Komunikasi Perpustakaan Perguruan Tinggi Kristen Indonesia (FKPPTKI) Student Paper	<1%
31	Submitted to Politeknik Kesehatan Kemenkes Semarang Student Paper	<1%
32	jurnal.uns.ac.id Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		