Pemanfaatan Aljabar Vektor Pada Mesin Pencari

Anwar Ramadha 13514013
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514013@std.stei.itb.ac.id

ABSTRAK-Sistem yang dapat menghasilkan informasi yang relevan dengan apa yang dicari oleh penggana disebut dengan sistem temu balik informasi. Makalah ini membahasa tentang model yang digunakan oleh mesin pencari yang salah satunya adalah dengan menreapkan model ruang vektor. Tujuan pembuatan makalah ini adalah memahami cara kerja mesin pencari yang menggunakan model ruang vektor. Hasil akan terurut berdasarkan kerelevanan suatu dokumen dengan kata kunci (*Query*). Semakin besar kecocokan dokumen dengan *query*, maka data akan prioritas dokumen semakin besar.

Kata kunci : ruang vektor, sistem temu balik informasi.

I. LATAR BELAKANG

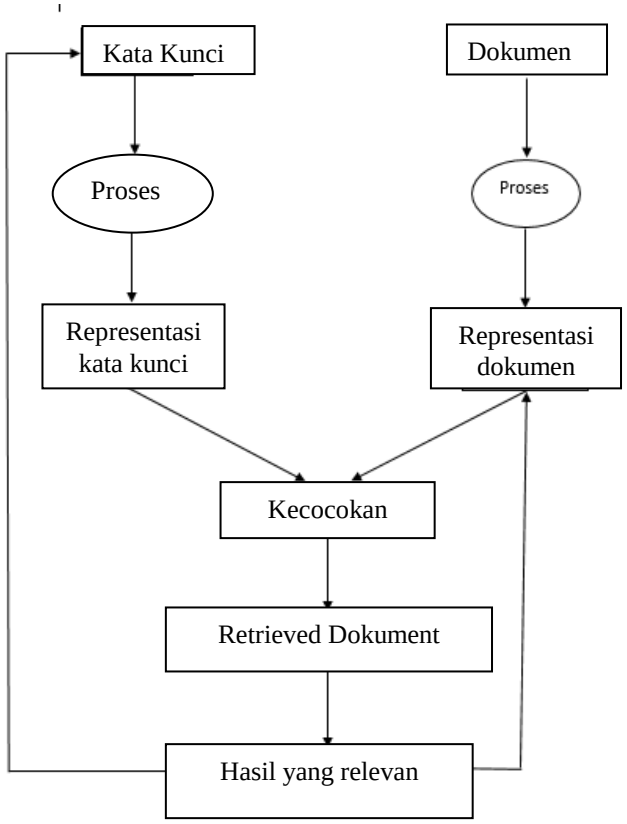
Saat ini, informasi telah menjadi kebutuhan primer bagi setiap umat manusia. Selaras dengan pesatnya perkembangan teknologi, kebutuhan informasi tersebut sekarang sangat mudah di akses terutama melalui internet. Setiap hari informasi yang tersedia di internet semakin banyak.Informasi yang disediakan beragam bentuknya yaitu dapat berupa dokumen, gambar, video, suara, dan file-file lain yang mungkin dibutuhkan oleh seseorang.

Semua informasi yang terdapat di internet dapat dicari melalui sebuah mesin perncari. Msin pencari adalah suatu mesin yang menggunakan metode sistem temu balik informasi. Sistem ini akan menampilkan hasil-hasil yang relevan dengan apa yang dicari oleh pengguna dengan waktu yang relatif singkat.Walaupun kita dapat mencari semua informasi di mesin pencari, tapi hasil yang ditampilkan oleh mesin pencari kadang dirasa kurang memuaskan karena banyaknya data yang harus diolah. Oleh karena itu, kebutuhan akan suatu mekanisme pencarian dokumen yang lebih efektif dirasakan semakin mendesak (Mandala dan Setiawan.2002; Jaya.2007).

Dalam metode sistem temu balik informasi, ada beberapa model yang dapat digunakan untuk menilai kepresisisan suatu hasil pencarian, antara lain yaitu model Boolean dan model ruang vektor. Kedua model ini mempunyai kelebihan dan kelemahan masing-masing. Model Boolean adalah model yang pertama kali digunakan. Implementasinya pun relatif mudah jika dibandingkan dengan model lainnya, tetapi waktu yang dibutuhkan untuk menjalankan (*run-time*) model ini relatif lebih lama. Selanjutnya adalah model ruang vektor. Model ini dapat menghasilkan informasi yang lebih relevan dan terurut. Waktu yang dibutuhkannya pun relatif lebih singkat karena tidak membutuhkan perhitungan yang berlebihan. Makalah ini akan membahas tentang model ruang vektor.

II. TEORI DAN METODOLOGI

2.1 *Flowchart* Sistem Temu Balik Informasi



Gambar 2.1 : Proses Temu Balik Informasi Dokumen Text
Sumber : Jurnal Temu Balik Informasi Pada Dokumen Teks Berbahasa Indonesia Dengan Metode Vector Space Retrieval Model, 2012

2.2 Model Ruang Vektor / *Vektor Space Retireval Model* (VSM)

Dalam sistem temu balik informasi ada 2 proses, yaitu proses *indexing* yang terdiri dari :

1. *Parsing* dokumen yaitu proses pengambilan kata-kata dalam dokumen.
2. *Tokenizing* adalah pembuangan tanda baca sehingga kata-kata pada dokumen merupakan kata yang berdiri sendiri.
3. *Stoplist* yaitu proses untuk membuang kata-kata yang tidak layak untuk dijadikan pembanding seperti tetapi,sedangkan,dll.
4. *Stemming* yaitu proses untuk mengubah kata menjadi kata dasarnya, contoh "Pembuangan" menjadi "buang".
   Algoritma yang dapat digunakan pada proses ini salah satunya adalah algoritma stemming yang dibahas pada bagian bantuan.
5. Menghitung frekuensi lokal dari suatu term (kata kunci/keyword) i dalam dokumen j (TF_{ij}) dan frekuensi global dari suatu term I dalam dokumen j (IDF_{ij}) serta dimasukkan ke dalam indeks.

Proses yang kedua yaitu proses pengurutan dokumen. Pada proses ini setiap kata diberi bobot sesuai dengan rumus $W_{ij}=TF_{ij}*IDF_{ij}$…. (1), dimana

W_{ij} : bobot istilah kata I dalam dokumen j.
TF_{ij} : frekuensi istilah kata I dalam dokumen j.

Contoh 1 :

Terdapat beberapa dokumen seperti berikut:
D1 : Teknik Informatika adalah salah satu program studi yang bermarkas di labtek 5.

D2 : Teknik Informatika adalah salah satu program studi terbaik di Institut Teknologi Bandung.

D3 : Semua mahasiswanya adalah mahasiswa yang jujur

Misalnya kita akan mencari "Institut Teknologi Bandung", maka langkah pertama yang harus di lakukan adalah membuat tabel yang menunjukkan frekuensi kemunculan kata-kata pada setiap dokumen.

Tabel 2.1 : Tabel kemunculan kata pada setiap dokumen.

Kata	D_1	D_2	D_3
Teknik	1	1	0
Informatika	1	1	0
Salah	0	1	0
Program	0	1	0
Labtek	1	0	0
5	1	0	0
Satu	0	1	0
Baik	0	1	0
Institut	0	1	0
Teknologi	0	1	0
Bandung	0	1	0
Mahasiswa	0	0	1
Semua	0	0	1
Jujur	0	0	1
Studi	0	1	0

Lalu hitung bobot global , yaitu IDF = $\log(\frac{N}{DF})$…(2), dimana N adalah jumlah dokumen. Pada dokumen diatas, diasumsikan jumlah dokumen adalah 1. Sehimgga IDF = $\log(\frac{N}{DF})$ = 2.

Setelah itu hitung bobot dokumen dengan rumus W=TF * IDF. Sehingga menghasilkan tabel dibawah ini :

Tabel 2.4 : Tabel hasil perkalian IDF dengan TF.

Kata	D_1	D_2	D_3	W		
				D_1	D_2	D_3
Teknik	1	1	0	2	2	0
Informatika	1	1	0	2	2	0
Salah	0	1	0	0	2	0
Program	0	1	0	0	2	0
Labtek	1	0	0	2	0	0
5	1	0	0	2	0	0
Satu	0	1	0	0	2	0
Baik	0	1	0	0	2	0
Institut	0	1	0	0	2	0
Teknologi	0	1	0	0		0
Bandung	0	1	0	0	2	0
mahasiswa	0	0	1	0	0	2
Semua	0	0	1	0	0	2
Jujur	0	0	1	0	0	2
Studi	0	1	0	0	2	0

Setelah proses diatas selesai, proses selanjutnya adalah mengurutkan dokumen berdasarkan hasil yang paling relevan dengan kata kunci.

Sebelumnya akan dijelaskan sedikit mengenai *Retireval Model*. *Retrieval model* dapat di deskripsikan sebagai

proses komputasi, contohnya bagaimana dokumen di urutkan dan bagaimana dokumen serta indeks di letakkan pada suatu implemnetasi. *Retireval Model* juga dapat mencoba mendeskirpsikan apa yang dininginkan manusia akan sebuah informasi serta menampilkan hasil yang relevan.

VSM adalah model yang menggunakan pembobotan kata dan pengurutan dokumen.Hasil *retrieval* yang didapat dari model ini adalah dokumen terurut yang relevan dengan kata kunci.,.

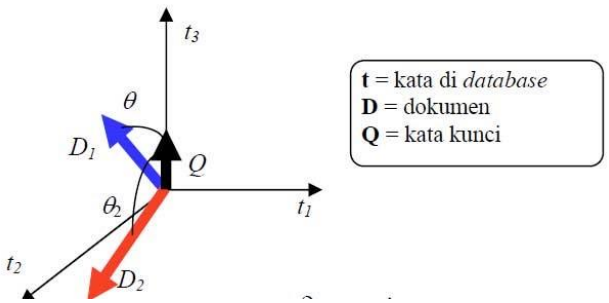
Model ruang vector memiliki keuntungan dibanding dengan model boolean, diantaranya adalah

1. Model yang sederhana karena merupakan aljabar linear.
2. Term Weight / kata tidak biner.
3. Memungkinkan menghitung berkelanjutan secara bersamaan antara kata kunci dan dokumen.
4. Pengurutan dokumen lebih relevan.
5. Memungkinkan pencocokan parsial.

Pada ruang vektor, dokumen direpresentasikan sebagai matriks kata-dokumen. Nilai elmen w_{ij} adalah bobot kata *I* dalam dokumen *j*. Misalkan terdapat kata *T* dalam jumlah *n*, yaitu T = $[T_1,T_2,T_3,\ldots,T_n]$ dan dokumen *D* dalam jumlah m, yaitu M = $[D_1,D_2,\ldots,D_m]$. Maka representasi matrik kata-dokumennya adalah sebagai berikut.

$$\begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_m \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \ldots. & T_n \end{array} \\ \begin{pmatrix} w_{11} & w_{21} & \ldots & w_{n1} \\ w_{12} & w_{22} & \ldots & w_{n2} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ w_{1m} & w_{2m} & \ldots & w_{nm} \end{pmatrix} \end{array}$$

Gambar 2.2 : matriks kata-dokumen
Sumber : Jurnal Pencarian Dokumen Berdasarkan Kombinasi Antara Ruang Vektor dan Model Domain Ontologi, 2010



Gambar 2.3 : Sudut yang dibentuk oleh vektor D dan vektor Q
Sumber:
https://commanderx78.wordpress.com/2012/09/24/model-vector-space/
Relevansi dokumen dengan kata kunci dapat dilihat dari besarnya sudut antara vektor doumen dan vektor kata kunci (*Query)*. Semakin kecil sudut antara 2 vektor ini, maka kemiripan akan semakin besar.

Besarnya sudut didapat dari perkalian titik antara Q dan D persamaan,, sehingga

Q . D=|Q| . |D| cosΘ, $|D|=\sqrt{\sum_{i=1}^{n} Di^2}$, $|Q|=\sqrt{\sum_{i=1}^{n} Qi^2}$...(4)

Rumus yang digunakan untuk menghitung kesamaan adalah sebagai berikut

$$\text{Sama}(Q,D)=\cos(Q,D)=\frac{Q \cdot D}{|Q||D|}\ldots\ldots(5)$$

Besarnya sudut merupakan suatu indikasi kesamaan antara dokumen dan kata kunci, semakin besar nilai cosinus artinya kedekatan antara kata kunci dan dokumen semakin besar.

Contoh :

Kita akan melanjutkan contoh 1 untuk mencari urutan dokumen berdasarkan hasil yang paling relevan dengan kata kunci.

Pertama hitung dulu jumlah bobot dokumen (misal sum) yang bersesuaian dengan kata kunci. Perhitungan bobot dapat diamati dibawah ini.

Sum = $D_1+D_1+D_1$ = 0 + 0 + 0 = 0
Sum = $D_2+D_2+D_2$ = 2 + 2 + 2 = 6
Sum = $D_3+D_3+D_3$ = 0 + 0 + 0 = 0

Setelah jumlah bobot ditentukan, tentukan besar nilai Q yaitu dengan membagi bobot perkata dengan jumlah bobot.

Nilai Institut = $\frac{W}{sum} = \frac{2}{6}$ = 0.33

Nilai Teknologi = $\frac{W}{sum} = \frac{2}{6}$ = 0.33

Nilai Bandung = $\frac{W}{sum} = \frac{2}{6}$ = 0.33

hitung sudut yang dihasilkan dengan menggunakan persamaan (5), Sehingga kita akan mendapatkan tabel sebagai berikut.

Tabel 2.5 : Tabel hasil perkalian titik Q dan D.

Kata	Q	D_1	D_2	D_3	Q.D		
					D_1	D_2	D_3
Teknik	0	1	1	0	0	0	0
Informatika	0	1	1	0	0	0	0
Salah	0	0	1	0	0	0	0
Program	0	0	1	0	0	0	0
Labtek	0	1	0	0	0	0	0
5	0	1	0	0	0	0	0
Satu	0	0	1	0	0	0	0
Baik	0	0	1	0	0	0	0
Intitut	0.33	0	1	0	0	0.33	0
Teknologi	0.33	0	1	0	0	0.33	0
Bandung	0.33	0	1	0	0	0.33	0
Mahasiswa	0	0	0	1	0	0	0
Semua	0	0	0	1	0	0	0
Jujur	0	0	0	1	0	0	0
studi	0	0	1	0	0	0	0

Kemudian cari nilai |Q| . |D| sesuai dengan persamaan (4)

$|Q| = \sqrt{\sum_{i=1}^{n} Qi^2} = \sqrt{(0.33 + 0.33 + 0.33)^2} = 0.5716$

$|D| = \sqrt{\sum_{i=1}^{n} Qi^2} = \sqrt{(0.33 + 0.33 + 0.33)^2} = 0.5716$

Lalu tentukan nilai cosinus masing-masing dokumen dengan menggunakan persamaan (5)

Untuk dokumen D_1 $\cos\Theta = \frac{Q \cdot D}{|Q||D|} = \frac{0}{0.33} = 0$

Untuk dokumen D_1 $\cos\Theta = \frac{Q \cdot D}{|Q||D|} = \frac{0.33}{0.33} = 1$

Untuk dokumen D_3 $\cos\Theta = \frac{Q \cdot D}{|Q||D|} = \frac{0}{0.33} = 0$

Didapat sudut untuk D_1=90, D_2=0, D_3=90. Sehingga urutan data yang relevan dengan kata kunci adalah D_2,D_1,D_3. Urutan tersebut memperlihatkan bahwa dokumen yang paling sesuai dengan kata kunci adalah D_2

III. BANTUAN

1. Rumus

   Untuk mencari sebuah kesamaan antara kata kunci dan dikumen yang bersesuaikan dibuthkan beberapa persamaan yang dapat dilihat dibawah ini

   a. Untuk mengurutkan suatu dokumen kita perlu tahu bobot dari dokumen tersebut yang dapat dihasilkan dari perkalian antara frekuensi lokal term I dalam dokumen j (TF_{ij}) dengan frekuensi global term i dalam dokumen j (IDF_{ij}). Frekuensi global merupakan logaritma dari pembagian jumlah dokumen dibagi dengan jumlah dokumen yang mengandung kata kunci.

   $W_{ij} = TF_{ij} * IDF_{ij}$
   $IDF_{ij} = \log(\frac{n}{DFij})$

   Ket :
   W_{ij} = Bobot term i dalam dokumen j.
   TF_{ij} = Frekuensi lokal kata kunci i dalam dokumen j.
   IDF_{ij} = Frekuensi global kata kunci i dalam dokumen j.
   DF_{ij} = Jumlah dokumen yang mengandung kata kunci i.

   b. Setelah menentukan bobot dokumen, langkah selanjutnya adalah mengurutkan dokumen berdasarkan hasil yang paling relevan dengan kata kunci. Persamaan yang digunakan adalah persamaan yang memanfaatkan perkalian titik pada vektor yaitu :

   $Q \cdot D = |Q||D| \cos\Theta$

   dimana Θ adalah sudut yang dibentuk oleh vektor Q dan D. Q adalah vektor yang menunjukkan kata kunci sedangkan D adalah vektor yang menunjukkan dokumen serta |Q| adalah besar vektor Q dalam skalar, begitu juga |D|.

   $|D|=\sqrt{\sum_{i=1}^{n} Di^2}$, $|Q|=\sqrt{\sum_{i=1}^{n} Qi^2}$

   Dari penurunan persamaan diatas didapat

   $\cos(\Theta)= \frac{Q \cdot D}{|Q||D|}$

   Nilai dari cos(Θ) inilah yang menjadi acuan untuk mengurutkan dokumen. Semakin besar nilai cos(Θ), semakin besar pula kesamaan antara dokumen dengan kata kunci.

2. Tabel

   Algoritma stemming adalah suatu proses untuk membuang imbuhan suatu kata agar menjadi kata dasar. Algoritma ini dibutuhkan dalam model ruang vektor. Algoritma ini mempunyai langkah-langkab sebagai berikut :

1. Cari kata yang akan diubah didalam kamus, jika ditemukan maka kata itu adalah kata dasar dan algoritma akan berhenti.
2. Buang *inflection suffix* (“-lah”, “-kah”,”-mu”,”nya”). Jika merupakan *particle* (“-lah”, “-kah”,”-tah”,”pun”) maka langkah ini akan diulangi dan jika ditemukan *Possesive pronoun* (“-ku” ,”-mu”,”-nya”) maka imbuhan tersebut akan dihilangkan.
3. Hilangkan *Derivate Suffix* (“-i”,”-au”,”kan”). Proses akan berhenti jika kata tersebut terdapat dalam kamus. Jika kata tidak ditemukan, maka lanjutkan ke proses 3a.
    a. Jika kata “-an” telah dihapus kemudian terdapat huruf “-k” diakhir kata, maka “-k” juga dihapus. Kemudian cek kata dalam kamus. Jika ditemukan, maka algoritma akan berhenti, jika tidak lanjutkan ke proses 3b.
    b. Akhiran (“-i”,”-kan”,”an”) yang telah terhapus dikembalikan kemudian lanjutkan ke proses 4.
4. Hilangkan *Derivation Suffix*. Jika pada proses 3 ada *suffix* yang dihapus maka lanjutkan ke proses 4a, jika tidak lanjut ke proses 4b.
    a. Cocokkan kata dengan tabel kombinasi awalan dan akhiran yang diizinkan, jika ditemukan, maka algoritma berhenti, lanjut ke langkah 4b jika tidak.
    b. Lakukan *looping* dari 1 sampai 3 temukan tipe dan hapus awalan. Jika kata dasar belum dapat ditentukan, lanjutkan ke langkah 5. Algoritma akan berhenti jika kata dasar sudah dapat ditentukan atau jika awalan pertama sama dengan awalan kedua.
5. Lakukan recoding.
6. Jika semua langkah sudah dilakukan tetapi kata dasar belum ditemukan didalam kamus, maka kata awal diasumsikan sebagai kata dasar.

Tipe awalan dapat ditentukan melalui langkah berikut :

1. Jika sebuah kata mempunyai awalan “-di”, ”-ke”,”-se”, maka tipe awalannya secara berturut-turut adalah “-di”, ”-ke”,”-se”,
2. Jika “-te”,”=me”,”-be”,”-pe” adalah awalan sebuah kata, maka dibutuhkan proses lain untuk menentukan tipe awalan.
3. Jika 2 karakter pertama bukan merupakan imbuhan yang telah disebutkan diatas, maka proses berhenti.
4. Jika “none” adalah awalan, maka proses berhenti. Jika bukan, maka cek tabel 2.

Tabel 1 : Kombinasi awalan-akhiran yang tidak diperbolehkan

Awalan	Akhiran yang tidak diperbolehkan
be-	-i
di-	-an
ke-	-i,-kan
me-	-an
se-	-i,-kan

Tabel 2 : Cara menentukan tipe awalan untuk kata yang diawali dengan “-te”

karakter		Tipe Awalan
bagian 1	bagian 2	
“-r”	“-r”	Tidak ada
“-r”	Vokal	Ter-luluh
“-r”	bukan (vokal or “-r-”)	ter
“-r”	bukan (vokal or “-r-”)	ter-
“-r”	bukan (vokal or “-r-”)	ter
bukan (vokal or “-r-”)	“-er”	Tidak ada
bukan (vokal or “-r-”)	“-er”	te

Karakter		Tipe Awalan
Bagian 3	Bagian 4	
-	-	Tidak ada
-	-	Ter-luluh
“-r”	vokal	ter
“-r”	bukan vokal	ter-
Bukan “-r”	bukan (vokal or “-r-”)	ter
vokal	-	none
bukan vokal	-	te

Tabel 3 : Tipe awalan berdasarkan jenis awalan

Tipe Awalan	Awalan yang harus dihilangkan
di-	di-
ke-	ke-
se-	se-
te-	te-
ter-	ter-
ter-luluh	ter

3. Daftar Singkatan dan Istilah
    1. Term : Kata kunci
    2. Query : Kata kunci
    3. DF : Document Frequency
    4. TF : Term Frequency
    5. IDF : Invers of Document Frequency

IV. KESIMPULAN

Ternyata dalam aplikasinya, aljabar geometri dapat digunakan sebagai salah satu cara membangkitkan metode model ruang vektor untuk mencari kesamaan antara kata kunci dan dokumen yang tersedia. Hasil yang didapat pun lebih relevan jika dibanding dengan model lainnya. Penggunaann model ini bisa dibilang cukup mudah karena hanya memanfaatkan ilmu aritmatika. Rumus lain yang digunakan, yaitu perkalian titik juga telah dipelajari sejak SMA. Tetapi tentu saja algoritma yang harus dituliskan lebih kompleks daripada model yang lain.

DAFTAR PUSTAKA

[1] Karyono, Giat., Utomo, Fandi Steyo (2012). Temu Balik Informasi Pada Dokumen Teks Berbahasa Indonesia Dengan Metode Vector Space Retrieval Model, Semarang. Volume 2, No. 1, http://publikasi.dinus.ac.id/index.php/semantik/article/view/141, 29 November 2015.

[2] Hadhiatma, Agung (2010). Pencarian Dokumen Berdasarkan Kombinasi Antara Ruang Vektor dan Model Domain Ontologi, Yogyakarta.jurnal.upnyk.ac.id/index.php/semnasif/article/view/1189. 29 November 2015.

[3] "Model Vektor Space". 29 November 2015. https://commanderx78.wordpress.com/2012/09/24/model-vector-space/

[4] Agusta, Ledy (2010). Perbandungan Algoritma Stemming Dengan Algoritma Nazief & Adriani Untuk Stemming DokumenTeks BahasaIndonesia,Bali. https://yudiagusta.files.wordpress.com/2009/11/196-201-knsi09-036-perbandingan-algoritma-stemming-porter-dengan-algoritma-nazief-adriani-untuk-stemming-dokumen-teks-bahasa-indonesia.pdf. 4 Desember 2015

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 04 Desember 2015

ttd

Anwar Ramadha 13514013