Implementasi Aljabar Vektor pada Sistem Temu Kembali Informasi untuk *Customer Information*

Ratnadira Widyasari 13514025
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ratnadira.widyasari@std.itb.ac.id

Customer information **tentu digunakan di berbagai perusahaan. Saran dan kritik dari pengguna jasa sangat diperlukan untuk mngembangkan suatu perusahaan. Namun, seringkali saran dan kritik tersebut diabakan karena membutuhkan waktu yang lama untuk menangani dan menglompokan kritik dan saran yang ada. Jumlahnya yang banyak serta kesulitan untuk menyusun data karena data yang tidak terstruktur merupakan persoalan yang cukup rumit. Penambahan persoalan pun terjadi karena kritik dan saran terus bertambah. Proses pencarian dokumen akan membutuhkan waktu yang sangat lama apabila membuka dokumen satu persatu. Persoalan rumit tersebut dapat diselesaikan menggunakan suatu metode yang bernama sistem temu kembali informasi yang dihasilkan dari implementasi aljabar vektor. Aplikasi dari sistem temu kembali infomasi tersebut akan dapat menglompokkan data serta mengurutkan data yang ada sesuai dengan masukan pengguna aplikasi. Bahasa yang digunakan oleh pengguna pun adalah bahasa natural.**

Keywords—customer information**, sistem temu kembali informasi, aljabar vektor,** ***generalized vector space model, vector space model***

I. PENDAHULUAN

Pesatnya perkembangan teknologi saat ini, selalu diiringi dengan bertambah banyaknya berkas digital yang disimpan. Setiap harinya berkas-berkas tersebut selalu bertambah, informasi yang ada juga harus selalu diperbarui sesuai data yang ada. Untuk mendapatkan suatu data, apabila pencarian dilakukan secara manual, maka akan menghabiskan waktu yang lama.

Metode pencarian telah dikembangkan sepanjang waktu untuk dapat menemukan hasil pencarian dengan cepat dan tepat. Basis data adalah salah satu salah satu penyelesaiannya. Namun, data yang ada dalam database adalah data terstruktur, serta *query* yang diberikan bukanlah bahasa natural. Pada basis data terdapat relasi antar *query* yang bersifat buatan (*artificial*), sehingga pengguna tidak memiliki wewenang untuk memodifikasi *query*.

Penulis merasa bahwa dibutuhkan suatu metode yang dapat menemukan serta menyusun data secara cepat walau data tersebut tidak terstruktur, serta *query* yang dapat diterima dari bahasa natural. Banyak metode pencarian yang digunakan dan dikembangkan, salah satunya adalah *find first*. Namun seiring banyaknya berkas yang terus bertambah hal ini menyebabkan *find first* tidak lagi digunakan karena data yang dicek satu per satu akan memakan banyak waktu. Setelahnya, ditemukan sistem temu balik informasi yang menjadi solusi yang tepat untuk permasalahan tersebut. Sistem temu balik informasi akan membebaskan cara penyimpanan serta *query* yang diberikan sehingga pengguna dan penyimpan data akan lebih leluasa menggunakannya.

Customer Information merupakan tema yang diangkat dalam makalah ini. Customer dari suatu perusahaan dapat memberikan saran atau kritik pada suatu perusahaan, data ini merupakan data yang tidak terstruktur, maka aplikasi untuk *customer information* dapat diproses menggunakan sistem temu balik informasi untuk menemukan data dan dapat menyusunnya hingga terurut sesuai dengan kedekatan pada kata kunci yang dicari. Salah satu dari model temu balik informasi adalah model ruang vektor, teori pada aljabar vektor digunakan pada metode ini.

II. DASAR TEORI

A. Sistem Temu Balik Informasi

Sistem temu balik informasi adalah suatu sistem yang mampu melakukan penyimpanan, pencarian, dan pemeliharaan informasi (Kowalski, 2000)

Search engine adalah salah satu aplikasi yang memanfaatkan sistem temu balik informasi. Contoh lainnya dapat dilihat dari sistem informasi perpustakaan.

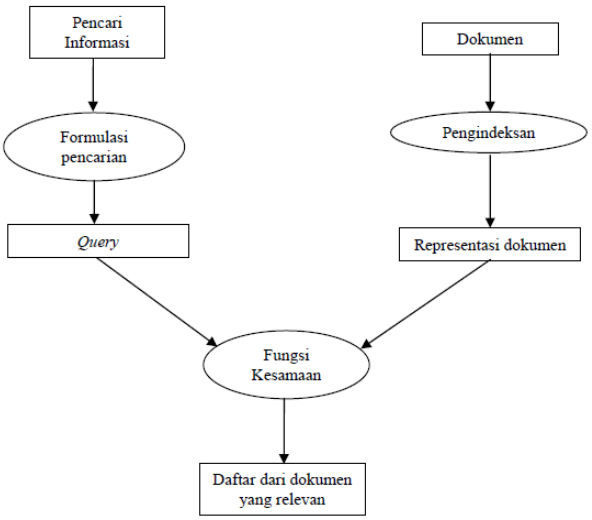
Hal yang membedakan sistem temu balik informasi dengan sistem pencarian pada database adalah dari ekspresi kebutuhan pengguna yang disebut *query* yang tidak memiliki struktur, serta data yang disediakan yang

juga tidak memiliki struktur. Perbedaan sistem temu kembali data (*data retrieval*) dan sistem temu kembali informasi (*information retrieval*) secara lebih lanjut dapat dilihat dari tabel dibawah ini:

	Data Retrieval	***Information Retrieval***
Matching	*Exact Match* (sama persis)	*Partial (best) Match* (hasil terbaik)
Inference	Deduksi	Induksi
Model	Deterministik	Probabilistik
Klasifikasi	*Monothetic*	*Polythetic*
Bahasa *Query*	*Artificial*	*Natural*
Spesifikasi *Query*	Lengkap	Tidak Lengkap
Item yang diinginkan	*Matching*	Relevan
Respon Error	Sensitif	Tidak Sensitif

Tabel 1 Perbedaan data retrieval dan information retrieva[3] *(Rijsbergen, 1979)*

Berikut adalah gambar dari kerangka sederhana sistem temu balik informasi:



Gambar 2 Kerangka information retrieval sederhana[3] *(Ingswersen, 1992)*

B. Algebraic Model

Sistem temu balik informasi dibedakan menjadi tiga model besar yaitu

1. *Probabilistic model*, proses pengambilan dokumen yang diperlakukan sebagai *probabilistic inference*
2. *Set-theoritic* models, dokumen direpresentasikan sebagai himpunan kata atau frase
3. *Algebraic model*.

Aljabar yang digunakan pada sistem temu balik informasi adalah aljabar linear. Aljabar linear yang didalamnya terkandung sistem persamaan linier dan solusinya, vektor, serta transformasi linier dan tidak dilupakan matriks dan operasinya. Contoh dari model ini adalah *Vector Space Model* (VSM) dan *Generalized Vector Space Model* (GVSM).

Seluruh dokumen dan ekspresi kebutuhan pengguna digambarkan (direpresentasikan) dalam vektor. Hal ini dilakukan untuk menemukan kesamaan dari ekspresi dengan dokumen yang ada. Acuan untuk pengurutan dokumen akan didapatkan dari nilai skalar vektor.

C. Vector Space Model

Vector Space Model (VSM) akan mengibaratkan seluruh dokumen dan *query* sebagai vektor n-dimensi, dengan menghitung derajat kesamaan antar tiap dokumen dengan *query* lalu disimpan dalam sistem.

Langkah yang dilakukan untuk menggunakan VSM yang pertama adalah *query* dilakukan proses *break into token*, *filtration*, dan *stemming* sehingga didapatkan *root word*.

Break into token adalah teks yang ada akan diproses menjadi unit-unit yang lebih kecil yaitu kata. Karakter dan simbol selain a-z pada proses *break into token* dihilangkan, pemecahan kalimat dan kata dilakukan berdasarkan pada spasi di dalam kalimat tersebut. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua kata ke bentuk huruf kecil (*lower case*). Contoh dari *break into token* adalah:
Input: Kamu memang satu-satunya untukku tapi aku bukanlah satu-satunya untukmu!
Output: kamu memang satu satunya untukku tapi aku bukanlah satu satunya untukmu

Filtration adalah proses untuk menghilangkan kata yang tidak relevan. *Filtration* dilakukan dengan cara membandingkan kata dengan *stop word*. *Stop word* adalah data dari kata yang tidak relevan namun sering ada pada dokumen. Contoh dari proses *filtration* adalah:
Input: kamu adalah satu untuk kamu adalah aku
Output: kamu satu kamu aku

Stemming untuk menghilangkan imbuhan dari suatu kata. Imbuhan bahasa Indonesia lebih kompleks dari pada bahasa inggris, hal ini terjadi karena di dalam bahasa Indonesia terdapat awalan (prefiks), infiks (sisipan), akhiran (sufiks), konfiks (gabungan prefiks dan sufiks). Contoh dari proses *stemming* adalah:
Input: kamu memang satu satunya untukku tapi aku bukanlah satu satunya untukmu
Output: kamu memang satu satu untuk tapi aku bukan satu satun untuk

Selanjutnya, *minterm* dapat ditentukan, dan penentuan frekuensi kata yang muncul pada dokumen dapat dihitung. Pada model ini, bobot dari *query* dan dokumen dinyatakan dalam bentuk
vektor, seperti:

$Q = (wq1, wq2, wq3, \ldots, wqt)$

$Di = (wi1, wi2, wi3, \ldots, wit)$

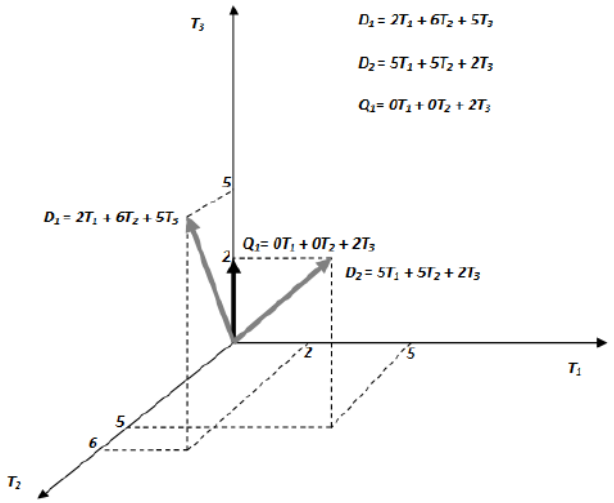
Dengan *wqj* dan *wij* sebagai bobot istilah *Tj* dalam *query Q* dan dokumen *Di*. Selanjutnya, dengan formula inner product koefisien kesamaan antara *query* dan dokumen dapat diperoleh, sebagai berikut:

$$sim(Q, D_i) = \sum_{j=1}^{t} w_{qj} \cdot w_{ij}$$

Formula diatas dapat digantikan dengan formula ternormalisasi, yaitu:

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} \cdot w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \times \sum_{j=1}^{t} (w_{qj})^2}}$$

Berikut adalah contoh dari model ruang vektor:



Gambar 3 Contoh model ruang vektor[2]

Terlihat dari gambar bahwa *query* $Q_1=0T_1+0T_2+2T_3$, dokumen $D_1=2T_1+6T_2+5T_3$ serta $D_2=5T_1+5T_2+2T_3$. Nilai sinus yang didapatkan dari formula yang telah diberikan sebelumnya adalah sebagai berikut:

$$\text{Sim}\left(\vec{d_1}.\vec{q}\right) = \frac{(2x0)+(6x0)+(5x2)}{(\sqrt{4+36+25})(\sqrt{0+0+4})} = \frac{10}{\sqrt{65,4}} = 0,62$$

$$\text{Sim}\left(\vec{d_2}.\vec{q}\right) = \frac{(5x0)+(5x0)+(2x2)}{(\sqrt{25+25+4})(\sqrt{0+0+4})} = \frac{4}{\sqrt{54,4}} = 0,27$$

Nilai diatas memperlihatkan bahwa dokumen D2 lebih mirip dengan dengan *query* dibanding dokumen D1, terlihat dari sudut D2 yang lebih kecil.

D. Generalized Vector Space Model

Generalized Vector Space Model akan menerjemahkan *query* serta dokumen menjadi vektor-vektor. Selanjutnya, dilakukan operasi perkalian terhadap vektor-vektor tersebut untuk menentukan relevansi *query* terdahap dokumen. Hasil yang didapatkan dari *Generalized Vector Space Model*, dilakukan dengan beberapa proses yaitu:

a. Kata depan dan penghubung dibuang.
b. Imbuhan awalan dan akhiran dibuang.
c. *Minterm* digunakan untuk menentukan kemungkinan pola frekuensi kata. Jumlah kata pada *query* akan menentukan panjang minterm.
d. Frekuensi kata pada dokumen yang sesuai dengan *query* dihitung.
e. *Index term* dihitung, dengan formula sebagai berikut:

$$\overrightarrow{Ki} = \frac{\sum_{\forall r, gi(Mr)=1} Ci,r.\overrightarrow{Mr}}{\sqrt{\sum_{\forall r, gi(Mr)=1} Ci,r^2}}$$

$\overrightarrow{Ki}$: *Index term* ke-i

$\overrightarrow{Mr}$: vektor ortogonal sesuai pola *minterm* yang terpakai

Ci,r: Faktor korelasi antara *Index term* ke-i dengan *minterm* r

$$C_{i,r} = \sum_{d_j | g_i(\overrightarrow{d_j}) = g_i(M_r)} W_{i,j}$$

$C_{i,r}$: Faktor korelasi antara *Index term* i dengan *minterm* r

$W_{i,j}$: Berat *Index term* i pada dokumen j

$g_i(M_r)$: bobot *Index term* K_i dalam *minterm* M_r

f. Dokumen dan *query* diubah menjadi vektor

$$\overrightarrow{d_j} = \sum_{i=1}^{n} W_{i,j} x \overrightarrow{K_i}$$

$$\vec{q} = \sum_{i=1}^{n} q_i x \overrightarrow{K_i}$$

$\overrightarrow{d_j}$: vektor dokumen ke-j

$\vec{q}$:vektor *query*

$W_{i,j}$:berat *Index term* i pada dokumen j

Q_i :berat *Index term* pada *query* i

n :jumlah *Index term*

g. Dokumen diurutkan menggunakan hasil perkalian vektor dengan formula, sebagai berikut:

$$\text{Sim}\left(\overrightarrow{d_j}.\vec{q}\right) = \frac{\overrightarrow{d_j}.\vec{q}}{|\overrightarrow{d_j}||\vec{q}|}$$

$\overrightarrow{d_j}$: vektor dokumen ke-j

$\vec{q}$: vektor *query*

Perbedaan antara *vector space model* dan *generalized vector space model* terletak pada perhitungan korelasi antar *query* dan dokumen. *Generalized vector space model* menjadikan seluruh *term* sebagai vektor ortogonal untuk menghitung *Index term* dan setelah itu setiap *term* pada dokumen diubah dengan cara generalisasi menjadi vektor ortogonal dengan mengalikan hasil *Index term* dengan *term* dokumen dan *query*. Selanjutnya, setiap vektor tersebut dikenakan operasi perkalian titik dan hasilnya menjadi acuan dalam menentukan relevansi *query* terhadap kumpulan dokumen.

E. Customer Information

Definisi *customer* menurut kamus bahasa inggris Oxford;

1. a person or organization that buys something from a shop or business

2. a person of the specified type

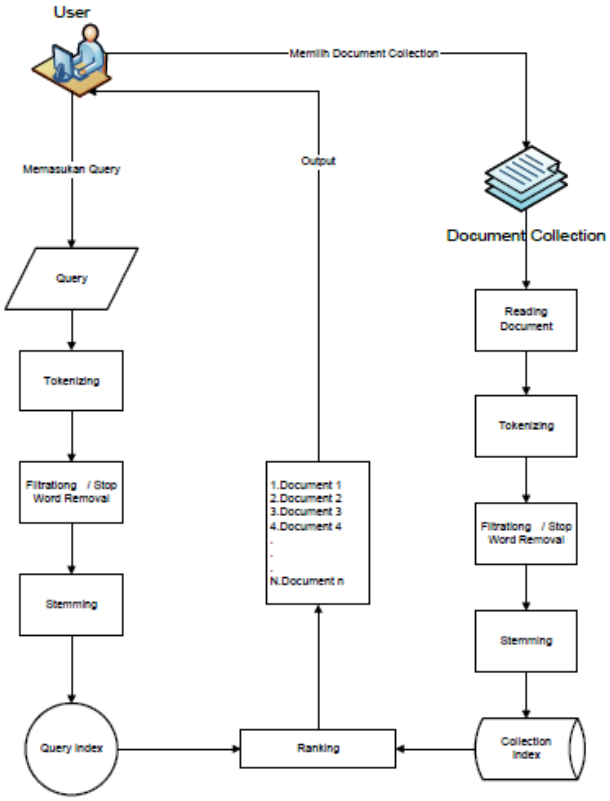
Adapun definisi informasi menurut kamus besar bahasa indonesia;

1. penerangan;
2. pemberitahuan; kabar atau berita tentang sesuatu;
3. *Ling* keseluruhan makna yang menunjang amanat yang terlihat dalam bagian-bagian amanat itu;

Customer information adalah data yang berisi tentang informasi dari pengguna jasa suatu perusahaan. Data tersebut dapat berisi banyak hal, antara lain informasi seperti nama, tempat tanggal lahir dan semacamnya. Informasi lain yang tersedia adalah kesan, pesan, saran dan kritik yang diberikan pengguna jasa untuk suatu perusahaan. Data tersebut merupakan data yang tidak terstruktur, sehingga untuk mendapatkan hasil pencarian dari kesan dan pesan dapat digunakan sistem temu balik informasi.

III. APLIKASI SISTEM TEMU KEMBALI INFORMASI PADA *CUSTOMER INFORMATION*

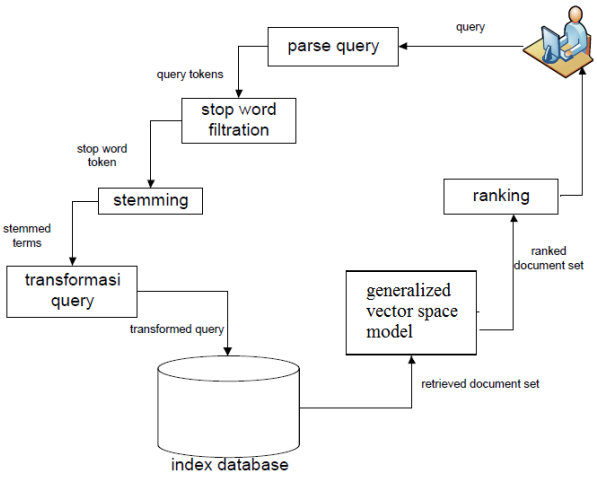
Sistem temu kembali dapat diterapkan pada *customer information*. Berikut merupakan contoh rancangan sistem temu kembali menggunakan *generalized vector space model*:



Gambar 4 Rancangan sistem temu kembali[2]

Document didapatkan dari hasil masukan para pengguna jasa dari suatu perusahaan. Dalam hal ini data yang tidak terstruktur berisi data dari kesan, pesan kritik maupun saran dari para pengguna jasa. Input *query* didapatkan dari masukan pengguna aplikasi sistem temu balik tersebut. Selanjutnya, *output* dari sistem temu balik informasi tersebut adalah *list* terurut yang diurutkan berdasarkan kedekatan dokumen tersebut dengan *query* yang telah dimasukkan. Aplikasi sistem temu balik tersebut terdiri atas dua proses operasi yaitu proses operasi *document* serta proses operasi *query*.

Searching process terdiri atas dua bagian utama yaitu *parse query* dan tahapan pemodelan menggunakan *generalized vector space model*.



Gambar 5 Skema proses seraching[2]

Dokumen-dokumen yang relevan akan dikembalikan dengan sistem yang membandingkan *stemmed term query* yang dihasilkan tersebut dengan koleksi dokumen. Selanjutnya dilakukan proses memodelkan *query* dengan kumpulan *term* menggunakan salah satu metode sistem temu kembali informasi. Tiap *term* atau kata yang ditemukan pada dokumen dan *query* diberi bobot dan disimpan sebagai salah satu elemen vektor dan dihitung nilai kemiripan antara *query* dan dokumen. Setelah itu perankingan dokumen dapat dilakukan berdasarkan kemiripan antara *query* dan dokumen.

Untuk mengukur efektivitas, rasio umum yang biasa digunakan ada dua yaitu *precision* dan *recall*. *Recall* adalah kemampuan suatu sistem untuk menampilkan seluruh dokumen yang relevan. Sedangkan, *precision* adalah ukuran kemampuan suatu sistem untuk menampilkan hanya dokumen relevan.

$$Precision = \frac{\text{jumlah dokumen relevan yang berhasil ditemukan}}{\text{Jumlah dokumen yang ditemukan}}$$

$$Recall = \frac{\text{jumlah dokumen relevan yang berhasil ditemukan}}{\text{Jumlah dokumen relevan dalam koleksi}}$$

Nilai *recall* dan *precision* selalu berbanding terbalik, semakin tinggi nilai *recall* semakin rendah nilai *precision*, begitu juga sebaliknya semakin rendah nilai *recall* semakin tinggi nilai *precision*. *Precision* dapat dihitung pada berbagai titik *recall*. Secara umum, semakin tinggi nilai *recall* semakin banyak jumlah dokumen yang harus

dicari. Pada mesin pencarian yang sempurna, hasil pencarian semuanya merupakan dokumen yang relevan atau dengan kata lain pada setiap nilai *recall*, nilai *precision* selalu satu. Pada kenyataannya, ada dokumen yang tidak relevan juga diambil oleh mesin pencari.

IV. KESIMPULAN

Aplikasi sistem temu kembali dapet menggunakan berebagai macam model, salah satunya *Generalized Vector Space Model*. Sistem temu kembali tersebut salah satunya dapat diterapkan pada *customer information*. *Customer infromation* memungkinkan pengguna jasa untuk memasukan kritik, saran, kesan maupun pesan. Data tersebut merupakan data yang tidak terstruktur, sehingga tidak bisa diolah menggunakan *database*.

Aplikasi sistem temu kembali pada *customer information* mampu menemukan kembali serta mengurutkan data yang paling sesuai dengan *query* yang telah dimasukan. Sistem temu kembali ini dapat bekerja dengan baik pada jumlah dokumen yang sedikit maupun banyak.

Perusahaan kadang lalai untuk memikirkan kritik dan saran dari pengguna jasa, dengan aplikasi sistem temu kembali tersebut, pegawai perusahaan dapat dengan cepat dan tepat memeriksa data dari pengguna jasa dan mengatasinya. Kritik dan saran dapat dikelompokkan secara tepat dengan aplikasi tersebut, tanpa harus menggunakan waktu yang lama untuk mengelompokannya secara manual.

REFERENSI

[1] Rinaldi Munir, *Aplikasi Aljabar Vektor pada Sistem Temu-balik Informasi IF2123: Aljabar Geometri*, Bandung, 2015.

[2] Lomhot Robinson, *Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI*, Bandung, 2014.

[3] Muhammad Erwin Ashari Hariyono, Wahyudi, *customer information gathering menggunakan metode temu kembali informasi dengan model ruang vektor*, Yogyakarta, 2005.

[4] Firnas Nadirman, *Sistem Temu-Kembali Informasi dengan Metode Vector Space Model pada Pencarian File Dokumen Berbasis Teks*, Yogyakarta, 2006.

[5] Hendra Bunyamin, *Algoritma Umum Pencarian Informasi dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen*, Bandung, 2005.

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, 13 Desember 2015

Ratnadira Widyasari, 13514025