# Rancang Bangun *Information Retrieval System* pada *Curriculum Vitae* dengan Metode *Vector Space* Model

[1]Royadi, [2]Richardus Eko Indrajit, [3]Eka Fitriani
E-mail : [1]royadi619@gmail.com, [2]indrajit@post.harvard.edu, [3]Fitrianieka817@gmail.com

**ABSTRAK**

*Curriculum vitae* merupakan gambaraan seseorang akan pegalamannya. *Curriculum vitae* sering dijadikan bahan utama bagian recruitment perusahaan untuk menetapkan posisi kerja calon pegawai. Banyaknya dokumen *curriculum vitae* sering menyulitkan bagian *recruitment* untuk menentukan diterima atau tidaknya calon pegawai. Dengan menggunakan metoda *vector space* model yang diterapkan di dalam program pencarian akan mempermudah mendapatkan data yang dibutuhkan, khususnya dalam hal pencarian data *curiculum vitae* berdasarkan kualifikasi pendidikan. Artikel ini memperlihatkan model konseptual yang dipergunakan dalam mengembangkan prototipa program komputer berbasis information retrieval.

Kata Kunci : *curriculum vitae, information retrieval, vector spac*e model, prototipa.

***ABSTRACT***

*The curriculum vitae is a person's description of the experience. Curriculum vitae is often used as the main ingredient of the company's recruitment department to determine the job positions of prospective employees. The number of documents curriculum vitae often complicate the recruitment to determine whether or not acceptance of prospective employees. Using the vector space model applied in the search program will make it easier to get the data needed, especially in terms of searching curiculum vitae data based on educational qualifications. This article shows a conceptual model used in developing prototype information-based computer retrieval programs.*

*Keywords: curriculum vitae, information retrieval, vector space model, prototype*.

## PENDAHULUAN

### Latar Belakang

*Curriculum vitae* sangat erat kaitannya dengan seleksi tenaga kerja. Adapun seleksi tenaga kerja itu sendiri merupakan suatu proses mencari sumber daya manusia yang tepat dari sekian banyak kandidat atau calon yang ada. Melihat daftar riwayat hidup (*curriculum vitae*) milik pelamar adalah hal pertama yang dilihat. Setelah itu dari data daftar riwayat hidup dilakukan pemilihan sesuai dengan kriteria yang dibutuhkan. Jika syarat yang diinginkan terpenuhi maka kandidat tersebut bisa masuk. Begitu sebaliknya.

Masalah yang sering terjadi dari tahap seleksi tenaga kerja diatas adalah pada tahap penyortiran cv. Tidak sedikit perusahaan yang kesulitan melakukan penyortiran cv Karena banyaknya cv pelamar. Tumpikan cv pelamar membuat penyeleksian tenaga kerja menjadi lebih lama. seleksi tenaga kerja bertujuan untuk mendapatkan sumber daya manusia yang memenuhi kualifikasi yang sesuai dengan kebutuhan organisasi, maka dasar kebijakan dalam seleksi adalah pemenuhan persyaratan kualifikasi yang menjadi dasar dalam proses seleksi.

Merujuk kepada masalah diatas, diperlukan adanya sebuah metode untuk memudahkan mendapatkan data dari *file-file* cv. oleh karena itu, penelitian ini akan menampilkan konsep *information retrival* dengan menerapkan model ruang *vector* (*vector space model)* berdasarkan kualifikasi pendidikan.

## TINJAUAN PUSTAKA

Menurut Amin (2012:78), "*Information retrival* merupakan sistem yang menemukan informasi yang sesuai dengan kebutuhan *user* dari kumpulan informasi secara otomatis. Pakem kerja *Information retrival* bisa dikatakan kumpulan dokumen dari seorang pengguna yang memformulasikan sebuah pertanyaan. Sampai pada akhirnya menghasilkan sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan".

*Information retrival* akan mengambil salah satu dari kemungkinan tersebut. *Information retrival* terbagi dalam dua komponen utama yaitu sistem *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.

### *Vector Space Model*

Menurut Isa (2013:231), "*Vector space model* sering dipakai untuk mempresentasikan kumpulan dokumen dalam suatu ruang. Dalam pemahaman ini, *Vector Space Model (VSM*) adalah metode untuk melihat tingkat kedekatan atau kesamaan similarity dengan cara pembobotan term. Dokumen dipandang sebagi sebuah vektor yang memiliki jarak dan 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".

*Vector Space Model* memberikan sebuah kerangka pencocokan parsial adalah mungkin. Masalah ini dimungkinkan 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*).

### *Curriculum Vitae*

Menurut Hariwijaya (2017:40), "CV adalah daftar riwayat hidup yang berisi ringkasan perjalanan pendidikan serta aktivitas professional seseorang. Dalam Kamus Besar Bahasa Indonesia CV mempunyai pengertian

atau definisi yakni uraian tentang segala sesuatu yang telah dialami (dijalankan) seseorang".

## METODE PENELITIAN

### Objek Penelitian

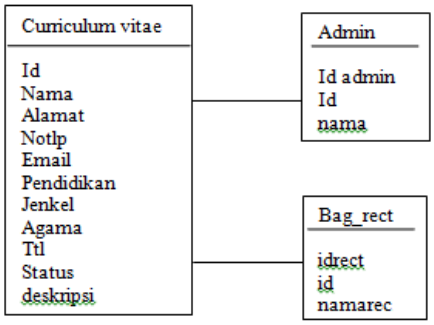
Sumber data yang menjadi objek penelitian adalah kumpulan dokumen *curriculum vitae* disebuah perusahaan. Dokumen *curriculum vitae* tersebut dikhususkan yang dikirim melalui *e-mail. Curriculum vitae* yang masuk kebagian *recruitmen* sebuah perusahaant. Pada akhurnya dokumen *curriculum vitae* tersebut akan *diretrive* berdasarkan kualifikasi Pendidikan.

### Format dan karakteristik data

Karakteristik data pada *curriculum vitae* adalah data yang terstruktur dan tidak terstruktur. Disebut data terstruktur Karena didalam sebuah *curriculum vitae* berisikan informasi atau deskripsi seseorang. Biasanya dibuat dengan struktur yang sama seperti membuat biodata. Ada nomor identitas, nama, alamat yang dibuat terstruktur. Tidak hanya terstruktur *curriculum vitae* juga bisa tidak terstruktur. Karena didalam sebuah curriculum vitae biasanya ada form khusus yang menjelaskan gambaran diri seseorang. Form khusus tersebut diluar dari struktural sebuah *curriculum vitae.* Form khusus tersebut ditulus dengan bebas oleh sipembuat *curriculum vitae*. Bahkan dibuat semenarik mungkin untuk meyakinkan perusahaan.

### Organisasi Penyimpanan Data

Organisasi penyimpanan data pada penelitian ini adalah sebagai berikut :




Gambar 1. Organisasi Penyimpanan Data

### Sintaks Pencarian Data

Sintaks pencarian data penelitian ini adalah dengan memasukan kata kunci salah satu kuaifikasi pendidikan. Misaslnya "SMA", "D3" dan sebagainya.

### Algoritma

Pada tahapan ini dijelaskan proses yang terjadi pada metode *Vector Space Model*. Proses tersebut dibagi menjadi tiga, yaitu proses *distance* atau penentuan jarak kedekatan antara dokumen, proses pembobotan dan yang terakhir penghitungan *Similarity*.

Tabel 1. Proses dan Metode *Vector Space Model*

| Proses | Vector Space Model |
|---|---|
| *Distance* | 1 |
| | *Hamming distance* |
| Bobot | Pembobotan TF – IDF |
| *Similarity* | *Cosine Similarity* |

Pada tabel 1, proses VSM dimulai dengan penghitungan jarak (*distance*) untuk mengetahui kedekatan antara *query* dengan dokumen *database*. Pada proses ini dibagi menjadi dua jenis, yaitu menggunakan *Hamming Distance* dan tanpa menggunakan *Hamming Distance* yaitu jarak dianggap 1 atau sama untuk semua dokumen. Langkah selanjutnya adalah proses pembobotan dengan menggunakan TF-IDF. Setelah dilakukan pembobotan, selanjutnya dihitung nilai *similarity* dengan metode *Cosine Similarity*.

Perhitungan VSM digunakan pembobotan TFIDF dan perhitungan nilai similarity dengan menggunakan *Cosine* Similarity. Metode TF-IDF adalah cara untuk memberikan bobot hubungan suatu term terhadap dokumen. Metode ini menggabungkan dua konsep perhitungan bobot yaitu frekuensi kemunculan kata dalam suatu dokumen dan *inverse* dari frekuensi yang mengandung kata tersebut. Persamaan dalam perhitungan TF-IDF terdapat pada rumusan (2) dan (3) sebagai berikut:

$W(t,d) = TF(t,d)x\ IDF$ (2)
$W(t,d) = TF(t,d)xLog\ D\ DFt$ (3)

Dimana :
W(t,d) : bobot term t pada dokumen d
TF (t,d) : total kemunculan term t pada dokumen d
D : total seluruh dokumen
DFt : total dokumen yang memiliki term t

***Text Prosessing***

Tahapan *Preproccesing* merupakan tahap persiapan yang dilakukan untuk menyiapkan dokumen sebelum diolah. Penggunaan *text preproccesing* dilakukan karena dokumen teks tidak dapat diproses langsung oleh algoritma pencarian, sehingga diperlukan proses untuk menghasilkan data numerik yang akan digunakan dalam perhitungan. tahapan *text preproccesing* pada penelitian ini meliputi :

a. Penghapusan format dan markup dalam dokumen
b. *Tokenizing*
c. *Filtering*
d. *Stemming*

***Cosine similarity***

Metode ini digunakan untuk menghitung nilai cosinus sudut antara dua *vector* dan mengukur kemiripan antar dua dokumen. Metode *Cosine Similarity* menggambarkan suatu kesamaan antara vektor *query* dan vektor dokumen dengan dilihat dari sudut yang paling kecil. Perhitungan *Cosine Similarity* dirumuskan pada persamaan berikut ini :

$$sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t}(W_{ij} \cdot W_{iq})}{\sqrt{\sum_{i=1}^{t} W_{ij}^2} \cdot \sqrt{\sum_{i=1}^{t} W_{iq}^2}}$$

Dimana:
$q$ = bobot query
$|q|$ = panjang query
$\vec{dj}$ = bobot dokumen
$|\vec{df}|$= panjang dokumen

## HASIL PENELITIAN

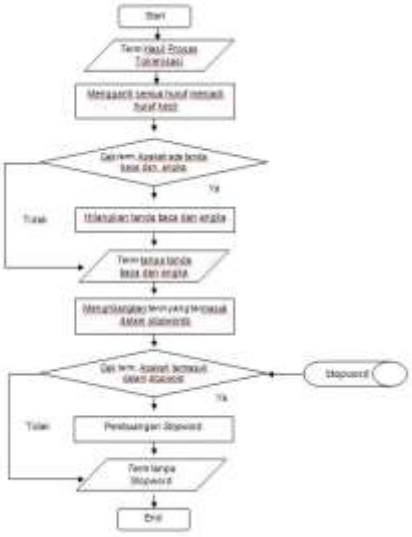
***Flowchart Tokenisasi***

Proses Tokenisasi dirancang untuk dapat memisahkan dokumen menjadi *term-term* yang akan diproses pada tahap *filtering*. Proses tokenisasi diawali dengan *scanner* dokumen yang ada pada korpus kemudian diproses menjadi *term*. *Flowchart* tokenisasi bisa dilihat pada gambar dibawah ini :


Gambar 2. *Flowchart Tokenisasi*

***Flowchart Filtering***

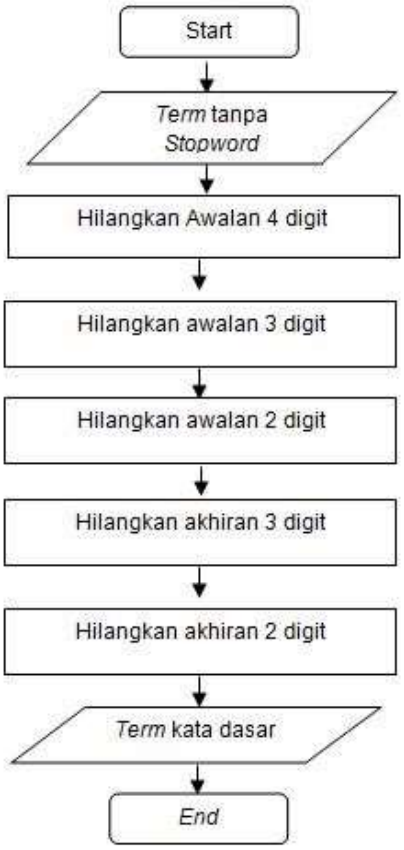
Proses *Filtering* dirancang untuk menghasilkan *term* tanpa *stopwords*. *Flowchart filtering* dimulai dengan mengganti huruf kapital menjadi huruf kecil, menghilangkan tanda baca dan angka.



Gambar 3. *Flowchart Filtering*

***Flowchart Stemming***

Proses *stemming* dirancang agar *term* hasil *filtering* diubah menjadi *term* kata dasar. Proses *stemming* dimulai dengan menghilangkan awalan dan akhiran. Proses ini juga dirancang dapat melakukan *replace* ketika awalan dihilangkan dan menggantinya dengan huruf yang sesuai. Proses menghilangkan awalan, akhiran, dan *replace* sisipan dilakukan dalam satu tahap proses.




Gambar 4. *Flowchart Steaming*

***Flowchart Indexing***

*Term* kata dasar hasil proses *stemming* selanjutnya dimasukkan dalam tabel untuk diproses pada perhitungan *Vector Space Model*. Proses *indexing* menggunakan metode *inverted indexing*, yaitu dengan membedakan letak tiap term dalam dokumen.


Gambar 5. *Flowchart Indexing*

**Deskripsi Data**

Data pada penelitian akan menggunakan 100 data *curriculum vitae* yang ada pada sebuah perusahaan. Dimana data cv tersebut akan disimpan ke dalam *database* yang terdiri dari parameter id, nama, alamat, no telepon, email, Pendidikan, jenis kelamin, agama, ttl dan deskripsi.

**Proses *Query* Input**

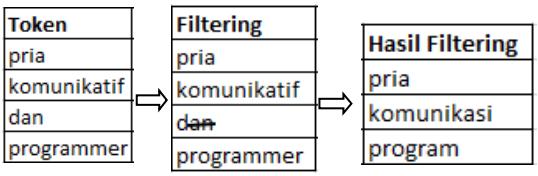
Proses *preprocessing query* input bertujuan untuk menyiapkan *query* input yang akan dibandingkan dengan dokumen yang ada di database. Pada proses ini dilakukan proses penghapusan format *markup*, *tokeninzing*, *filtering* dan *stemming* . Setelah terbentuk term dari query input dihitung frekuensi term yang akan digunakan sebagai *query* pembanding. Berikut contoh proses *Preproccessing Query* Input dengan input user "Pria komunikatif dan *programmer*".

Tabel 2. Proses *Query Input*

| **Query** | **Penghapusan format** | **Tokenizing** |
|---|---|---|
| pria komunikatif dan programer | pria komunikatif dan programer | Pria |
| | | komunikatif |
| | | Dan |
| | | programmer |

Proses yang pertama adalah penghapusan format dan *markup* pada *query input* yang dapat dilihat pada tabel 2. Proses ini mengembalikan *query* input dari *user* ke dalam bentuk huruf kecil untuk dilanjutkan ke proses selanjutnya. Selain mengembalikan ke huruf

kecil, pada proses ini juga menghapus beberapa format *tag* yang tidak diperlukan. Selanjutnya, pada proses *tokenizing*, dilakukan pemecahan *query* menjadi beberapa *term* yang dipisahkan berdasarkan spasinya.




Gambar 6. Proses *Filtering Query Input*

Proses *filtering* bertujuan untuk menghilangkan tanda baca dan simbol yang dianggap tidak penting dan digunakan dalam perhitungan. Pada proses ini juga dilakukan penghapusan *stopwords* yang tidak digunakan dalam perhitungan. hasil proses Stemming yang bertujuan untuk mengembalikan token yang sudah dipilih kedalam bentuk kata dasarnya. Selanjutnya term tersebut yang akan digunakan untuk dihitung frekuensinya.

Tabel 3. Perhitungan *Query*

| Term | Frekuensi |
|---|---|
| pria | 1 |
| komunikasi | 1 |
| program | 1 |

Langkah yang terakhir dalam proses *preprocessing* adalah menghitung frekuensi dari masing-masing *term query input*. Hasil perhitungan frekuensi dapat dilihat pada tabel 3. Frekuensi kemunculan inilah yang selanjutnya akan digunakan dalam proses pembobotan.

**Proses VSM**

*Information Retrieval System* akan melakukan proses perhitungan dimulai dari menghitung tfidf, menghitung jarak *query* dan jarak dokumen, menghitung similaritas produk, dan menghitung bobot dokumen. *Query* yang di *input* oleh user selanjutnya akan dilakukan pencarian pada tabel freq kemudian dilakukan perhitungan pembobotan menggunakan metode *Vector Space Model*. Perhitungan dilakukan dalam sistem pencarian, sistem pencarian akan melakukan perhitungan kemudian akan menampilkan hasilnya. Hasil pencarian akan menampilkan nama dokumen di korpus, kemudian bobot similaritas dan disusun berdasarkan perankingan. Bobot terbesar akan menempati ranking teratas pada hasi pencarian.

**Aplikasi**

Studi kasus pada aplikasi *Information Retrieval System* ini menggunakan dokumen-dokumen CV yang sudah terkomputerisasi.


Gambar 7. *User interface* Aplikasi

## KESIMPULAN DAN SARAN

**Kesimpulan**

Penggunaan metode *vector space model* sangat berpengaruh dalam membantu mempercepat waktu eksekusi sistem. Hal ini dikarenakan pada metode *Vector Space Model* apabila terdapat dokumen yang tidak memiliki kedekatan dengan *query* maka tidak akan dilanjutkan ke proses perhitungan selanjutnya.

**Saran**

Berdasarkan hasil penelitian tersebut, penulis mencoba memberikan saran dan hal-hal yang masih perlu diperhatikan :

a. Agar proses tidak terlalu lama, pada term yang memiliki frekuensi di semua dokumen dianggap sebagai stopword.
b. Proses *stemming* yang ada masih belum bisa sepenuhnya membuat semua *term* kedalam bentuk *term* kata dasar dengan benar. Proses ini akan mempengaruhi hasil untuk proses indexing, sehingga akan mempengaruhi hasil akhir perhitungan.

## DAFTAR PUSTAKA


Amin, F. (2012). Sistem Temu Kembali Informasi dengan Metode Vector Space Model, *2*, 78–83.

Hariwijaya, M (2017). 30 Hari Mencari Kerja. Elmarika. 2017

Isa, T. M., Abidin, F., Matematika, J., Kuala, U. S., Syech, J., No, A., Aceh, B. (2013). Mengukur Tingkat Kesamaan Paragraf Menggunakan Vector Space Model untuk Mendeteksi Plagiarisme. Pada, S., Joint, K., Body, O., Java, P. E., Ningrum, W., Sunuharyo, B. S., … Karyawan, K. (n.d.). Pengaruh pendidikan dan pelatihan terhadap kinerja karyawan (, *6*(2).