Available *online* at : http://bit.ly/InfoTekJar

**InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan**

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600

# Analisa Recall dan Precision Menggunakan VSM pada Kasus Text Mining

*Warnia Nengsih*

*Politeknik Caltex Riau, Jl. Umbansari No 1 Rumbai 28265 Pekanbaru Riau Indonesia*

KEYWORDS

Vector Space Model, Preprocessing, recall, precission,,text mining.

CORRESPONDENCE

Phone: +62-81363 -1849- 99

E-mail: warnia@pcr.ac.id

**A B S T R A C T**

Text Mining merupakan proses pengolahan untuk mengetahui pola-pola yang tidak terstruktur. Pola yang tidak terstruktur tersebut bisa ditemukan pada objek seperti jurnal, artikel, novel, buku dan sejenisnya. Implementasi fitur yang sering digunakan adalah teknik pencarian file atau dokumen yang memenuhi unsur efektif dan efisiensi. Pencarian file atau dokumen sangat ditentukan oleh ketepatan dan kesesuaian dokumen yang dipanggil dengan kata kunci yang digunakan. Semakin tepat kata kunci yang diinputkan semakin relevan dengan hasil yang ditampilkan. Agar hasil pencarian sesuai dengan keyword yang dimasukkan maka dibutuhkan algoritma pencarian Vector Space Model. Vector Space Model merupakan algoritma yang digunakan untuk melihat relevansi antara kata kunci dengan hasil pencarian yang ditampilkan. Dari hasil perhitungan recall dan precission, sistem dapat melakukan pengembalian dokumen sesuai dengan kata kunci yang dimasukkan pengguna. Dimana nilai recall yang diperoleh sebesar 100%. Pencarian menggunakan metode Vector Space Model dapat memberikan hasil yang maksimal dalam melakukan pencarian dokumen.

## PENDAHULUAN

Penyimpanan dan pencarian dokumen dalam sebuah database harus mengikuti metode pencarian yang benar sehingga efektifitas dan efisiensi bisa terpenuhi. Permasalahan yang ada, seringkali tak ada ukuran yang pasti akurasi kesesuaian dan ketepatan kata kunci yang dimasukkan dengan hasil pencarian yang ditampilkan.

Seyogyanya penggunaan kata kunci sangat berpengaruh terhadap hasil pencarian. Untuk mendukung proses pencarian yang lebih akurat maka perlu adanya sebuah algoritma. Algoritma yang diimplementasikan adalah *Vector Space Model*. Algoritma *Vector Space Model* merupakan metode pencarian yang menghitung kesesuaian dan ketepatan hasil dengan data input. Algoritma ini akan mencoocokan kata yang ada dalam database per kata sehingga lebih detail dan teliti. Proses perhitungan berdasarkan frekuensi kemunculan data sesuai dengan kata kunci yang ada. Kata tersebut akan dikonversi ke dalam bentuk vektor. Hasil dari perhitungan tersebut akan menjadi basis dalam penentuan kesesuian kata yang dicari

## METODOLOGI PENELITIAN

*A. Objek Dokumen pada Text Mining*

Pada *text mining*, dokumen digital yang akan diproses harus melalui tahapan document *preprocessing*. Tentunya proses ini bertujuan untuk mengurangi volume kosakata, menyeragamkan kata dan menghilangkan noise [1]. Berikut merupakan proses yang terdapat pada document preprocessing:

Tokenizing

Proses Tokenizing yaitu memisahkan deretan kata di dalam kalimat, paragraf atau dokumen menjadi token atau potongan kata tunggal atau termed word. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (case folding)[2].

Filtering

Pada tahapan ini melakuan proses stopword atau penghapusan kata yang tidak perlu seperti kata penghubung, kata ganti dan sebagainya

Stemming

Tahapan ini bertujuan untuk menghapus atau menghilangkan imbuhan-imbuhan yang ada. Setelah document preprocessing selesai maka selanjutnya adalah melakukan inverted index. Inverted index adalah salah satu proses untuk mengidekskan

https://doi.org/10.30743/infotekjar.v5i1.2663

sebuah koleksi teks yang digunakan untuk mempercepat proses pencarian[3]. Selanjutnya dilakukan proses penghitungan nilai bobot menggunakan TF – IDF, yang selanjutnya dilakukan perhitungan tingkat kemiripan kata kunci yang dimasukkan menggunakan vector space model.

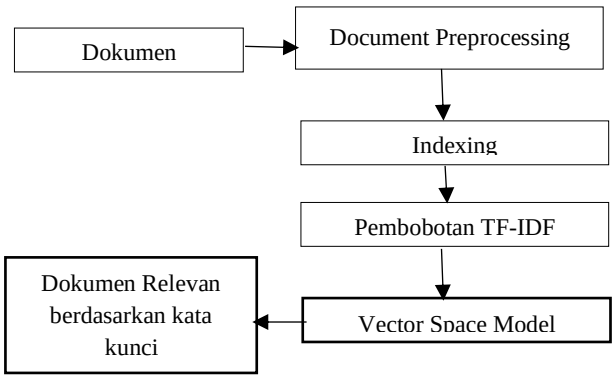
Algoritma Vector Space Model

Vector Space Model (VSM) adalah suatu metode untuk melihat tingkat kedekatan atau kesamaan (similarity) term dengan cara melakukan pembobotan term. Dokumen dan kata kunci dipandang sebagai sebuah vektor yang memiliki jarak dan arah. Relevansi sebuah dokumen ke sebuah query didasarkan pada similaritas diantara vektor dokumen dan vektor query [4][5].
Adapun rumus relevansi yang digunakan adalah

$$\cos\theta\ similarity(\vec{d_j}, \vec{q}) = \frac{(\vec{d_j} \cdot \vec{q})}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

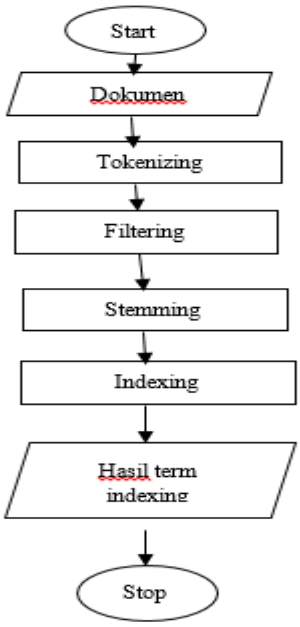
Dimana :
q = bobot kata kunci
d = bobot dokumen
$|\vec{d}|$ = panjang dokumen
$|\vec{q}|$ = panjang kata kunci

Berikut merupakan blok diagram dari pencarian dokumen ini




Gambar 1. Blok Diagram

Objek dokumen yang akan diolah berupa novel. Dokumen – dokumen tersebut harus melewati tahapan document preprocessing sebagai bagian dari tahapan pembersihan file. Selanjutnya dilakukan proses Indexing dan pembobotan TF –IDF sebelum implementasi metode Vector Space Model untuk menampilkan dokumen yang sesuai dengan kata kunci[2][6-10]. Hal yang paling penting pada tahapan ini adalah tahapan document preprocessing untuk memastikan bahwa kata-kata yang akan diolah sudah siap untuk diolah menggunakan metode Vector Space Model. Terdapat beberapa langkah pada tahapan ini yaitu tokenizing, Filtering, Stemming, Indexing.



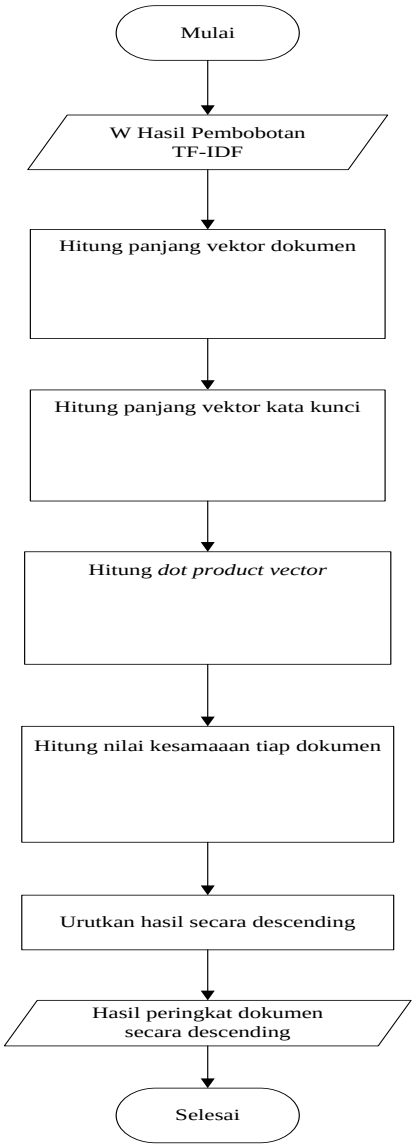

Gambar 2. Diagram Alir Document Preprocessing

Metode Vector Space Model melakukan pengolahan hasil dari pembobotan TF-IDF dengan menghitung panjang dari vektor setiap dokumen, di samping itu panjang vektor dari kata kunci juga haruss ditentukan,Jika sudah diperoleh hasil dot poduct vector maka hitung nilai kesamaan tiap dokumen, lakukan proses descending untuk menampilkan hasil peringat dokumen seperti yang ditunjukkan pada diagram alir gambar 3.

Berikut merupakan diagram Alir algoritma Vector Space Model.

Tabel 1. Hasil document preprocessing

| Dokumen | Halaman | Tahapan Document Preprocessing Tokenizing & Filtering: Jumlah kata setelah proses token dan filter | Waktu | Stemming: Jumlah kata setelah stemming | Waktu | Indexing: Jumlah Term | Waktu |
|---|---|---|---|---|---|---|---|
| a | 261 | 50177 kata | 135,84s | 50176 kata | 85,76s | 101388 | 161.5s |
| b | 164 | 21057 kata | 45.14s | 21057 kata | 34.37s | 4772 | 93.76s |
| c | 4 | 475 kata | 0.44s | 475 kata | 0.915s | 183 | 4.95s |
| d | 110 | 28612 kata | 85.97 | 28612 kata | 47.80 s | 3541 | 134.4s |
| e | 453 | 51864 kata | 102.17 s | 51864 kata | 81.76 s | 9871 | 159.9s |




Gambar 3. Diagram Alir Vector Space Model

## HASIL DAN PEMBAHASAN

Pengujian menggunakan lima dokumen dengan spesifikasi jumlah halaman yang berbeda. Setiap dokumen sudah melalui tahapan *document preprocessing* dengan spesifikasi jumlah kata yang dihasilkan dan waktu perolehan seperti yang ditunjukkan pada tabel 1.

Recall atau disebut juga dengan perolehan merupakan kemampuan sistem untuk menampilkan hasil sesuai dengan kata kunci yang dimasukkan. *Precission* atau ketepatan merupakan kemampuan tidak menampilkan dokumen yang tidak sesuai dengan kebutuhan pengguna. Pengujian dilakukan dengan memasukkan kata kunci yang terdiri atas satu, dua dan tiga suku kata.

Tabel 2. Hasil pencarian dokumen

| Dokumen | Nilai Kemiripan ( Similarity) | Relevan |
|---|---|---|
| A | 0.55 | ya |
| B | 0.35 | ya |
| C | 0.27 | ya |
| D | 0.20 | tidak |
| E | 0.19 | tidak |

Tabel 2 menunjukkan hasil pencarian dokumen. Terdapat beberapa dokumen yang relevan dan tidak relevan. Dokumen yang tidak relevan memiliki bobot yang tinggi hal itu dipengaruhi oleh frekuensi kemunculan kata-kata sehingga menyebabkan bobot tiap kata lebih tinggi. Sebaliknya dokumen relevan memiliki bobot rendah, hal itu dipengaruhi oleh kata-kata yang tidak memiliki kecocokan terhadap kata kunci.

Dari hasil tabel 2 diperoleh nilai *recall* dan *precission* sesuai dengan perhitungan berikut:

Recall

$$= \frac{5}{0+5} \times 100\ \% \ = 100\ \%$$

Precission

$$= \frac{3}{(3+2)} \times 100\ \% \ = 60\ \%$$

Dari hasil perhitungan *recall* dan *precission*, sistem dapat melakukan pengembalian dokumen sesuai dengan kata kunci yang dimasukkan pengguna. Dimana nilai *recall* yang diperoleh sebesar 100% dan nilai *precission* sebesar 60%. Sehingga dapat disimpulkan bahwa pencarian menggunakan metode *Vector Space Model* dapat memberikan hasil yang maksimal dalam melakukan pencarian dokumen.

## KESIMPULAN

Terdapat beberapa dokumen yang relevan dan tidak relevan. Dokumen yang tidak relevan memiliki bobot yang tinggi hal itu dipengaruhi oleh frekuensi kemunculan kata-kata sehingga menyebabkan bobot tiap kata lebih tinggi. Sebaliknnya dokumen releven memiliki bobot rendah. Hal itu dipengaruhi oleh kata-kata yang tidak memiliki kecocokan terhadap kata kunci. Dari hasil perhitungan recall dan precission, sistem dapat melakukan pengembalian dokumen sesuai dengan kata kunci yang dimasukkan pengguna. Dimana nilai recall yang diperoleh sebesar 100% dan nilai *precission* sebesar 60%. Sehingga dapat disimpulkan bahwa pencarian menggunakan metode *Vector Space Model* dapat memberikan hasil yang maksimal dalam melakukan pencarian dokumen.

## DAFTAR PUSTAKA

[1] Dao, S. D. & Marian, R. 2011. Optimisation Of Precedence-Constrained Production Sequencing And Scheduling Using Genetic Algorithms. *Proceedings Of The International Multi Conference Of Engineers And Computer Scientists*, 16-18 March, Hong Kong.

 https://doi.org/10.30743/infotekjar.v5i1.2663

[2] Gen, M. & Cheng, R. 2000. *Genetic Algorithms And Engineering Optimization.* John Wiley & Sons, Inc., New York.

[3] Liliana, D. Y. & Mahmudy, W. F. 2006. Penerapan Algoritma Genetika Pada Otomatisasi Penjadwalan Kuliah. *Laporan Penelitian Dpp/Spp*. Fmipa Universitas Brawijaya, Malang.

[4] Marian, R. M., Luong, L. & Dao, S. D. 2012. Hybrid Genetic Algorithm Optimisation Of Distribution Networks—A Comparative Study. *Dalam:* Ao, S. I., Castillo, O. & Huang, X. (Editor.) *Intelligent Control And Innovative Computing*. Springer, Us.

[5] Phanden, R. K., Jain, A. & Verma, R. 2013. An Approach For Integration Of Process Planning And Scheduling. *International Journal Of Computer Integrated Manufacturing*, 26(4), 284-302.

[6] Ridok, A. 2014. Peringkasan Dokumen Bahasa Indonesia Berbasis Non-Negative Matrix Factorization. *Jurnal Teknologi Informasi Dan Ilmu Komputer (Jtiik)*, 1(1), 39-44.

[7] Tala, F. Z. 2003. A Study Of Stemming Effects On Information Retrieval In Bahasa Indonesia. *Ph.D. Thesis*. Universiteit Van Amsterdam.

[8] Wang, L. 2007. *Process Planning And Scheduling For Distributed Manufacturing.* Springer, London.

[9] Wibawa, A. P., Nafalski, A. & Mahmudy, W. F. 2013. Javanese `Speech Levels Machine Translation: Improved Parallel Text Alignment Based On Impossible Pair Limitation. *Ieee International Conference On Computational Intelligence And Cybernetics*, 3-4 December, Yogyakarta, Indonesia. 16-20.

[10] Liliana, D. Y. & Mahmudy, W. F. 2006. Penerapan Algoritma Genetika Pada Otomatisasi Penjadwalan Kuliah. *Laporan Penelitian Dpp/Spp.* Fmipa Universitas Brawijaya, Malang.