

Implementasi *Generalized Vector Space Model* (GVSM) dalam Pencarian Buku di Perpustakaan

I Made Suwija Putra, Ni Putu Ayu Widiari, I Wayan Gunaya

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana

Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: putrasuwija@unud.ac.id, widiariayyu@gmail.com, gunnayaa@gmail.com

Abstrak

Perpustakaan sebagai tempat koleksi buku yang besar tentu akan menemui kesulitan pelayanan pencarian buku jika masih menggunakan teknologi konvensional. Dalam ilmu *Information Retrieval* terdapat beberapa metode dalam pencarian dokumen yang relevan, salah satunya *Generalized Vector Space Model* (GVSM). Metode ini merupakan perkembangan dari *Vector Space Model* yang menggunakan model aljabar. GVSM menggunakan pertimbangan kedekatan antar kata yang dinyatakan dalam vector dua dimensi. Dalam Aplikasi pencarian buku, judul, pengarang, dan sinopsis buku dimasukkan ke dalam database dan dilakukan preprocessing diantaranya tokenisasi, *stopword removal*, dan *stemming*. Hal yang sama untuk kata kunci (*query*) pencarian yang diinputkan user juga dilakukan proses preprocessing. Dalam GVSM, dokumen dan *query* pencarian akan diubah kedalam bentuk vector dan dicari similaritasnya berdasarkan perkalian vector. Hasil dari implementasi metode GVSM dalam aplikasi pencarian buku di perpustakaan didapatkan hasil perbandingan nilai *precision* dan *recall* sebesar 0.703 dan 0.910 yang artinya hasil pencarian yang ditampilkan masih mendekati keinginan user.

Kata Kunci: Pencarian Dokumen, *Generalized Vector Space Model*, Similaritas, *Query*

Abstract

The library as a large collection of books will certainly encounter difficulties in book search services if they are still using conventional technology. In Information Retrieval there are several methods in finding relevant documents, one them is the Generalized Vector Space Model (GVSM). This method is a development of the Vector Space Model uses algebraic models. GVSM uses closeness between words expressed in two-dimensional vectors. In the book search application, title, author, and synopsis the book stored in the database and then preprocessing includes tokenization, stop word removal, and stemming. This preprocessing also being done for search keywords entered by the user. In GVSM, documents and search queries will be converted into vector form and look for similarity based on vector multiplication. The results of the implementation of the GVSM method in book search applications in the library showed that the comparison of precision and recall values was 0.703 and 0.910, which meant that the search results displayed were still closer to the user's desire.

Keywords: Document Search, *Generalized Vector Space Model*, Similarity, *Query*

1. Pendahuluan

Buku merupakan kumpulan dari suatu informasi yang dimana informasi-informasi didalamnya saling mendukung dan memiliki satu tema yang sama[1] sehingga buku dikatakan sebagai jendela informasi dunia. Perpustakaan merupakan tempat mengkoleksi informasi atau buku yang massif saat ini sudah banyak memiliki sistem informasi perpustakaan guna mendukung pelayanan pendataan, pencarian, dan peminjaman buku[2]. Khusus untuk proses pencarian sangat diperlukan saat ini untuk menunjang dan mempermudah dalam memperoleh dokumen buku yang relevan dan sesuai dengan kebutuhan yang diinginkan.

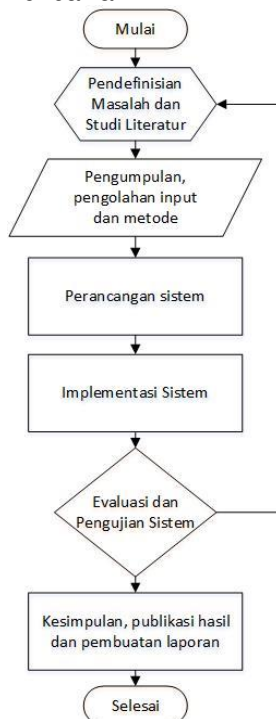
Kebanyakan sistem pencarian dokumen buku di perpustakaan daerah maupun di instansi pemerintah masih menggunakan pencarian biasa yaitu mencocokkan kata kunci yang sama dengan judul buku sesuai dengan *query* kondisi bahasa SQL. Hal ini terkadang menyebabkan

hasil pencarian yang ditampilkan tidak mencerminkan keinginan *user* yang sesungguhnya atau keinginan *user* untuk mencari buku yang mengandung kata kunci yang tidak ada dalam judul buku, melainkan ada di dalam sinopsis buku. Berdasarkan hal tersebut, maka diperlukan sebuah sistem perpustakaan yang dapat memberikan fitur pencarian lebih dalam bukan hanya sampai kepada judul buku namun juga isi buku guna menemukan informasi atau dokumen diperoleh secara cepat dan efisien. Namun saat ini mendapatkan buku yang sesuai akan sulit dikarenakan untuk mengetahui informasi yang terkandung didalam suatu buku harus membaca sinopsisnya karena judul belum bisa menggambarkan isi dari buku. Untuk memudahkan hal ini, sistem temu kembali informasi dapat diaplikasikan. Sistem temu kembali informasi dapat digunakan untuk mencari buku dengan syarat dimana tidak hanya buku dengan kueri yang sama yang ditampilkan, namun juga buku yang memiliki relevansi dengan kueri yang dimasukan. Metode yang dapat digunakan dalam *information retrieval* yaitu model klasik Boolean, model klasik vector, klasik probabilistik, dan model terstruktur[3]. Model klasik vector merupakan metode retrieval yang menggunakan konsep ruang vector. Pengembangan dari model ini yaitu Generalized Vector Space Model, dimana pada GVSM digunakan *min-term* dari kueri untuk menentukan kemungkinan frekuensi kemunculan kata.

Penelitian sebelumnya mengenai GVSM sudah dilakukan pada penelitian[4] yang dimana hasil yang didapatkan dengan menggunakan metode GVSM dapat menjalankan fungsi-fungsi pencarian dengan baik dan memiliki keakuratan yang tinggi. Penelitian lain[5] yang menggunakan metode GVSM untuk pencarian informasi pada dokumen di Teknik Elektro di UPT BPI LIPI, menghasilkan kesimpulan GVSM mampu mempermudah pencarian dokumen tertentu yang mengandung kata kunci (*query*) yang dimasukan oleh pengguna. Dalam penelitian ini akan diuji kembali dengan menggunakan contoh data buku dari perpustakaan yang berbeda untuk mendapatkan tingkat *precision* dan *recall* yang merupakan nilai keefektifan dari hasil uji metode GVSM.

2. Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam penelitian ini yaitu menggunakan pendekatan kuantitatif yang menitikberatkan pada aspek numerik sebagai data, baik dalam tahap pengumpulan data maupun hasil penelitian. Berikut ini adalah diagram alir dari tahapan penelitian seperti yang ditunjukkan pada Gambar 1 di bawah ini.



Gambar 1 Diagram alir tahapan penelitian

Penelitian ini dimulai dari mendefinisikan masalah yaitu mengenai bagaimana implementasi GVSM dalam aplikasi pencarian buku, yang dilanjutkan dengan studi literatur mengenai GVSM. Tahapan selanjutnya adalah mengumpulkan data, data yang digunakan pada penelitian ini berupa data buku yang dilengkapi dengan judul dan synopsis. Setelah semua perangkat terkumpul selanjutnya membuat perancangan dan implementasi sistem dan database. Setelah sistem jadi dilakukan pengujian dengan menggunakan data uji dan menarik kesimpulan dari penelitian yang telah dilakukan.

Penelitian ini menggunakan metodologi penelitian eksperimental dimana menitikberatkan pada riset kuantitatif. Penelitian menghasilkan *outcome* berupa hasil pencarian buku yang berdasarkan relevansi sinopsis dengan kueri yang dimasukkan, yang selanjutnya akan dilakukan proses perhitungan nilai *precision* dan *recall* dari sistem temu kembali yang dibuat.

3. Kajian Pustaka

3.1 Sistem Temu Kembali Informasi (*Information Retrieval*)

Sistem temu kembali informasi atau *information retrieval* merupakan sistem yang proses besarnya adalah untuk menemukan informasi yang relevan terhadap kebutuhan pengguna dari suatu sumber informasi yang banyak secara otomatis[6]. Pengaplikasian dari sistem temu kembali informasi adalah *search-engine* atau mesin pencari. Sistem temu kembali menggunakan dokumen sebagai objek datanya yang merupakan sumber dari informasi. Dokumen akan dilakukan pengindeksan menggunakan sistem tertentu, seperti TF-IDF. Sistem temu kembali informasi yang baik mampu menampilkan dokumen-dokumen relevan yang sesuai dengan kueri yang dimasukkan, mengurutkan dokumen tersebut, dan membuang dokumen yang tidak relevan. Terdapat berbagai metode yang dapat digunakan dalam pengindeksan ataupun dalam pencarian dokumen-dokumen relevan. Sistem temu kembali juga menggunakan nilai *precision* dan *recall* sebagai pengukur efektifitas dari sistem temu kembali informasi, dimana sistem temu kembali informasi dianggap baik jika nilai *precision* dengan *recall*-nya sama (1:1).

3.2 Text Preprocessing

Text preprocessing merupakan tahapan awal dimana dokumen akan dilakukan proses tokenisasi dan *stopword removal*. Proses ini dilakukan untuk mengubah data menjadi bentuk yang lebih terstruktur. Tokenisasi adalah proses untuk membagi teks dalam kalimat, paragraf, maupun dokumen, menjadi bentuk token atau bagian tertentu. Tahapan tokenisasi diawali dengan pembersihan data dari tanda baca, tag html, dan angka yang selanjutnya tiap *term* yang ada diganti menjadi huruf kecil.

Stopword removal adalah proses yang digunakan untuk menghilangkan *stopword list*[7]. *Stopword list* merupakan kata-kata yang diabaikan dalam pemrosesan data. Karakteristik dari *stopword* yaitu frekuensi kemunculan tinggi dan berupa kata ganti orang atau kata penghubung. Dalam *stopword list*, tidak ada *list* yang pasti dan benar untuk *stopword*. Tujuan dari penghilangan *stopword* ini adalah untuk mengurangi jumlah kata dalam sebuah dokumen yang akan berpengaruh dalam kecepatan dan performa pemrosesan.

3.3 Porter Stemming

Stemming merupakan proses membentuk kata dasar dari kata atau kumpulan kata yang sudah diberikan awalan dan akhiran. *Stemming* menggunakan struktur morfologi Bahasa untuk mengubah kata berimbuhan menjadi kata dasarnya[8]. Proses *stemming* digunakan dalam proses pencarian informasi untuk meningkatkan informasi yang didapatkan lebih berkualitas. Artinya hubungan kata dengan dengan varian kata lainnya didapatkan. Sebagai contoh kata “diculik”, “menculik” (melakukan tindakan culik) dan “penculik” (orang yang menculik) dengan masing-masing imbuhan tersebut memiliki arti yang berbeda-beda, dengan proses *stemming* didapatkanlah hasil sebuah kata “culik” sehingga arti dan varian kata di atas saling berhubungan.

Proses *stemming* untuk *corpus* yang menggunakan Bahasa Indonesia berbeda dengan *corpus* yang menggunakan Bahasa asing atau Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks (akhiran). Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan. *Stemming* pada Bahasa Indonesia didasarkan pada basis aturan (*rule based*), yang artinya akan melakukan pemotongan imbuhan dengan mematuhi aturan-aturan penggunaan suatu imbuhan. *Stemming* ini akan memeriksa setiap kata yang memiliki suatu imbuhan (sufiks ataupun prefiks) sampai benar-benar tidak

ditemukan imbuhan pada kata tersebut. Contoh algoritma *stemming* berbasiskan aturan adalah *Stemmer Porter*[9].

3.4 Generalized Vector Space Model

Generalized Vector Space Model (GVSM) merupakan metode yang dikembangkan dari metode sebelumnya yaitu Vector Space Model dengan mempertimbangkan kesamaan arti antar kata atau *term* dengan lebih akurat dalam mewakili representasi dari dokumen[10]. Wong, et al. (1985) memprakarsai Generalized Vector Space Model (GVSM) dengan memperkenalkan korelasi antar *term*, di mana vektor *index term* dianggap independen secara linier tetapi tidak dianggap berpasangan orthogonal. Vektor *index term* terdiri dari komponen-komponen yang lebih kecil[11][12]. Tahapan dari GVSM adalah sebagai berikut.

- Menentukan *minterm* dari *query* untuk menentukan kemungkinan frekuensi munculnya kata. Panjang *minterm* ini didasarkan pada banyak kata yang dimasukan pada *query*.
- Minterm* kemudian diubah menjadi vektor orthogonal \vec{m}_i sesuai dengan polanya, sehingga dapat didefinisikan sebuah set vektor \vec{m}_i yaitu $\vec{m}_1 = (1, 0, \dots, 0, 0)$, $\vec{m}_2 = (0, 1, \dots, 0, 0)$, hingga $\vec{m}_t = (0, 0, \dots, 0, 1)$.
- Menghitung *index term* dengan persamaan

$$\vec{k}_i = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \vec{m}_r}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c_{i,r}^2}} \quad (1)$$

- Dokumen (d) dan *query* (q) kemudian diubah menjadi vector dengan persamaan

$$\vec{d}_j = \sum_{i=1}^n w_{ij} \times \vec{k}_i \quad (2)$$

$$\vec{q} = \sum_{i=1}^n q_i \times \vec{k}_i \quad (3)$$

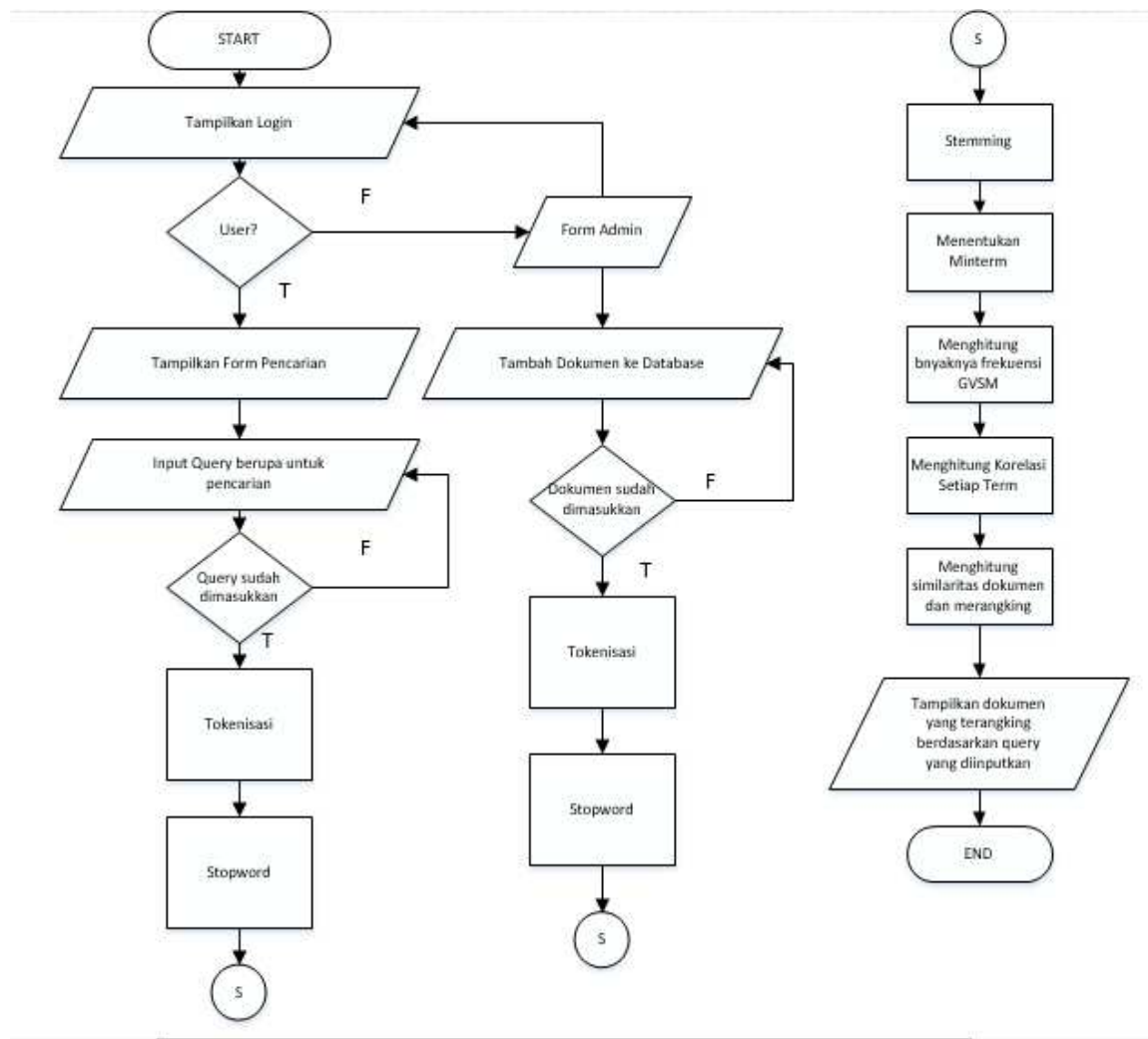
- Dokumen yang telah diubah menjadi vector kemudian diurutkan berdasarkan similaritas dengan menggunakan persamaan sebagai berikut

$$\text{sim}(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{\|\vec{d}_j\| \|\vec{q}\|} \quad (4)$$

Perbedaan antara VSM dengan GVSM terletak pada perhitungan kolerasi antara dokumen dengan kueri. GVSM menjadikan seluruh *term* sebagai vector orthogonal untuk menghitung *index term* dan kemudian setiap *term* pada dokumen diubah dengan cara generalizedisasi menjadi vektor ortogonal dengan mengalikan hasil Index *term* dengan *term* dokumen dan *query*. Selanjutnya, masing-masing vektor dilakukan operasi perkalian titik, kemudian hasilnya akan menentukan relevansi *query* terhadap kumpulan dokumen.

4. Percobaan dan Hasil

Perancangan sistem pencarian buku dengan metode Generalized Vector Space Model (GVSM) digambarkan dengan *flowchart* (diagram alir). Adapun perancangan sistem dari sistem pencarian buku dengan metode GVSM ditunjukkan pada Gambar 2 di bawah ini.

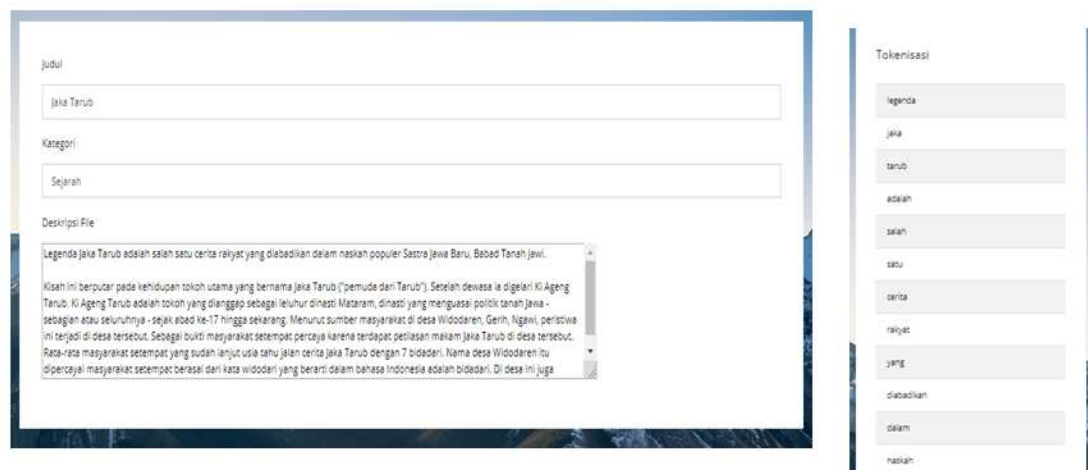


Gambar 2 Flowchart Sistem Pencarian Buku

Gambar 2 merupakan diagram alir dari sistem pencarian buku perpustakaan dengan metode GVSM (*Generalized Vector Space Model*). Tahap pertama dari sistem pencarian buku ini adalah menampilkan *form login* untuk menentukan apakah dia termasuk admin dan user. Halaman admin berisikan halaman untuk tambah abstrak buku, dimana abstrak buku akan dibersihkan dengan proses *preprocessing*. Halaman user berisikan *form* pencarian untuk memasukkan *query* pencarian. *Query* pencarian akan mengalami proses *preprocessing* seperti halnya dokumen. *Parse query* adalah tahapan *preprocessing* yang merubah kalimat *query* ke dalam kelompok kata (*term*). Hal ini melalui beberapa tahapan proses yaitu.

1. Tokenisasi

Tokenisasi merupakan suatu tahapan yang memproses teks input yang berisikan beberapa kalimat dibagi menjadi unit-unit kecil yang disebut token, yang dapat berupa kata, angka atau suatu tanda baca. Dalam sistem ini tahapan *tokenisasi* terdiri dari karakter dari a sampai z, segala simbol dihilangkan, pemecahan-pemecahan kalimat menjadi token dilakukan berdasarkan pada spasi di dalam kalimat tersebut dan hal ini dilakukan secara otomatis.



Gambar 3 Proses Tokenisasi

2. Proses Stopword Removal

Beberapa token atau kata yang dihasilkan dari proses proses *parse* dipilih dan dibuang kata-kata mana yang termasuk ke dalam kategori *stop word* dalam Bahasa Indonesia.

3. Proses Stemming

Setelah proses *filtering* pada *stop word filtration* maka token-token *query* yang dihasilkan akan dirubah ke bentuk dasarnya menjadi *stemmed term query*. Sistem akan membandingkan *Stemmed term query* yang dihasilkan tersebut dengan koleksi dokumen sehingga mengembalikan dokumen-dokumen yang relevan. Setelah proses *parse query* dilakukan tahapan selanjutnya adalah memodelkan *query* dengan kumpulan *term* menggunakan salah satu metode dari *information retrieval*. Metode yang digunakan pada penelitian ini adalah *Generalized Vector Space* model. Cara kerjanya adalah untuk masing-masing *term* atau kata yang ditemukan pada corpus dan *query* yang sudah melalui proses tokenisasi dan *stop word removal* diberi bobot dan disimpan sebagai salah satu elemen vektor untuk nantinya dihitung nilai kemiripan antara *query* dan dokumen dengan menggunakan operasi perkalian vektor. Setelah itu perankingan dokumen dapat dilakukan berdasarkan kemiripan antara *query* dan dokumen. Semakin tinggi nilai kemiripan maka dokumen tersebut yang akan dimunculkan pada peringkat teratas.

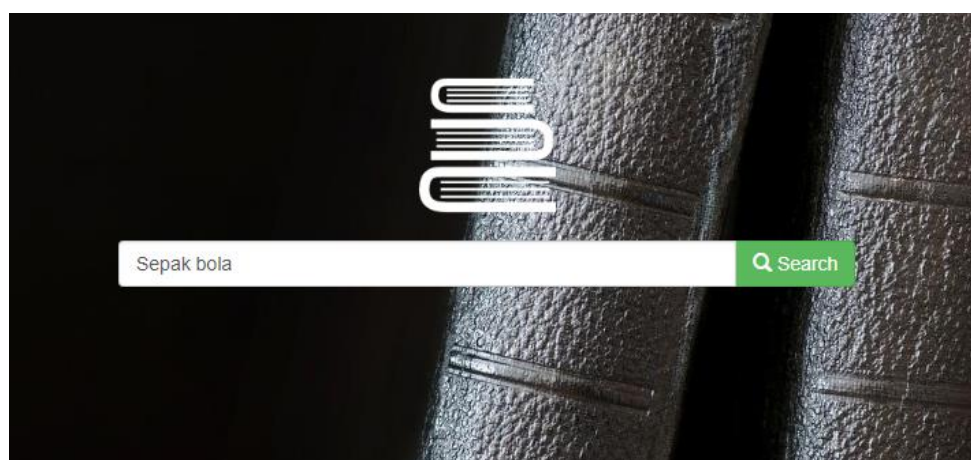
Hasil Stemming	
nyil	6
blorong	5
wujud	4
rupa	3
manusia	3
ular	3
ratu	2
sebut	2
budak	2
raksasa	2
konon	2
sakti	2
plilik	2
emas	2
kidul	2

Gambar 4 Hasil *Stopword Removal*, *Stemming*, dan Penghitungan *term* pada dokumen

Gambar 4 merupakan gambaran hasil sebuah corpus sudah melalui proses *stopword removal*, *stemming*, dan penghitungan *term* pada dokumen. Pada proses *stemming* didapatkan *term* kata dasar pada dokumen yang diinputkan. *Term* tersebut dihitung jumlah frekuensi kemunculannya dan setiap hasil *term* yang didapatkan semua dimasukkan pada *database* yang sudah dibuat sebelumnya.

4.1 Implementasi Sistem

Perancangan yang telah dilakukan kemudian diimplementasikan sesuai dengan desain diletakkan pada sistem di perpustakaan. Hasil implementasi kemudian diuji cobakan guna memastikan *output* yang dikeluarkan oleh aplikasi pencarian buku dengan metode GVSM (Generalized Vector Space Model) dengan inputan *query* dan *output* berupa data buku yang terkait dengan *query* yang diinputkan sudah sesuai.

**Gambar 5** Tampilan *Interface Search*

Gambar 5 merupakan tampilan *search* pada aplikasi pencarian buku dengan metode Generalized Vector Space Model (GVSM). *Query* yang diinputkan adalah kata kunci “sepak bola”.

No	Judul Buku	Kategori	Deskripsi
1	Bola	Scifi	Secara umum, hanya penjaga gawang saja yang berhak menyentuh bola dengan tangan atau lengan di dalam...
2	sepak bola	Scifi	sepak bola adalah olah raga yang paling populer di dunia yang menjadikan banyak sekali penggemarnya....
3	Persib Bandung	Scifi	Klub sepak bola nasional Persib Bandung memiliki riwayat yang kurang baik ketika ditukangi oleh pela...
4	Sepak Bola	Scifi	Sepak bola (bahasa Inggris: Football atau Soccer) adalah cabang olahraga yang menggunakan bola yang ...
5	Sepak Bola Modern	Scifi	Sejarah olahraga sepak bola (permainan menendang bola) dimulai sejak abad ke-2 dan ke-3 sebelum Mase...
6	peraturan sepak bola	Scifi	Kejuaraan sepak bola internasional terbesar ialah Piala Dunia yang diselenggarakan oleh Fédération...
7	sejarah sepak bola	Scifi	Sejarah sepak bola di Indonesia diawali dengan berdirinya Persatuan Sepak Bola Seluruh Indonesia (PS...
8	Persib Bandung 2	Scifi	Sebab sang pelatih baru sekali menukangi klub sepak bola di negara asalnya, yakni Montenegro. Dimana...

Gambar 6 Tampilan Hasil *Search*

Gambar 6 merupakan tampilan hasil search dari uji coba pencarian buku dengan metode GVSM (Generalized Vector Space Model). Hasil *search* dari *query* yang diinputkan adalah dimunculkan daftar dokumen buku yang judul, deskripsi yang memiliki tingkat kedekatan dengan *query* yang diinputkan yaitu Sepak Bola sebanyak 8 dokumen.

4.2 Hasil Pengujian

Untuk menguji ketepatan dan keakuratan implementasi metode GVSM dalam sistem pencarian buku perpustakaan, dilakukan penghitungan nilai *precision* dan *recall* untuk setiap pencarian yang dilakukan *user*. *Precision* mengindikasikan tingkat kesesuaian antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem, tetapi tidak memandang total jumlah dokumen yang relevan dalam kumpulan dokumen. Dengan kata lain nilai *precision* (P) didapatkan dengan membandingkan jumlah dokumen yang relevan bagi user dengan total jumlah dokumen yang ditemukan. Sedangkan *recall* adalah tingkat keberhasilan sistem menemukan kembali semua dokumen yang sesuai. Nilai *recall* (R) diperoleh rasio jumlah dokumen yang relevan bagi user dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan [13].

$$\text{Precision} = \frac{|(a) \cap (b)|}{|(b)|} \quad (5)$$

$$\text{Recall} = \frac{|(a) \cap (b)|}{|(a)|} \quad (6)$$

Diketahui:

- a : jumlah dokumen relevan
b : jumlah dokumen yang ada dalam *database*

Hasil pengujian nilai *precision* di-*retrieve* oleh sistem yang telah dilakukan dengan menggunakan 10 kata kunci dan penilaian dokumen buku yang dimunculkan sistem bernilai relevan atau tidak didasarkan dari penilaian yang diberikan oleh *user* dalam hal ini adalah mahasiswa. Berikut ini nilai masing-masing kata kunci untuk *precision* dilihat dalam Tabel 1 di bawah ini.

Tabel 1 Tabel nilai *precision* pada sistem pencarian buku dengan metode GVSM

No	Query	Relevant items retrieved	Total relevant items retrieved	P
1	Harry Potter	7	12	0.58
2	Olahraga Sepak Bola	8	8	1
3	Tubuh Manusia	6	15	0.4
4	Nyi Roro Kidul	3	4	0.75
5	Bencana Tsunami	5	5	1
6	Anatomi Manusia	6	13	0.46
7	Hantu roh	9	12	0.75
8	Kisah Mahabharata	8	12	0.67
9	Laskar Pelangi	1	1	1
10	Cinta Remaja	3	7	0.42
Rata-rata nilai <i>Precision</i>				0.703

Hasil pengujian nilai *recall* di-*retrieve* oleh sistem yang telah dilakukan dengan menggunakan 10 kata kunci dan penilaian dokumen buku yang relevan diambil dari nilai "*relevant items retrieved*" yang ada pada Tabel 1. Hasil nilai *recall* seluruh kata kunci dapat dilihat dalam tabel 2 di bawah ini.

Tabel 2 Tabel nilai *recall* pada sistem pencarian buku dengan metode GVSM

No	Query	Relevant items retrieved	Total relevant items in database	R
1	Harry Potter	7	7	1
2	Olahraga Sepak Bola	8	8	1
3	Tubuh Manusia	6	10	0.6
4	Nyi Roro Kidul	3	4	0.75
5	Bencana Tsunami	5	5	1

6	Anatomi Manusia	6	8	0.75
7	Hantu roh	9	9	1
8	Kisah Mahabharata	8	8	1
9	Laskar Pelangi	1	1	1
10	Cinta Remaja	3	3	1
Rata-rata nilai <i>Recall</i>				0.910

Hasil Pengujian nilai *Precision* dan *Recall* terlihat pada tabel 4.1 dan tabel 4.2 bahwa rata-rata hasil pengujian *precision* dan *recall* pada sistem pencarian buku dengan metode GVSM adalah 0.703 dan 0.910. Dalam bentuk perbandingan nilai *precision* dan nilai *recall* dari sistem pencarian buku dengan metode GVSM adalah 7:9.

5. Kesimpulan

Metode Generalized Vector Space Model adalah metode turunan dari Vector Space Model. Tahapan dari sistem pencarian dengan metode Generalized Vector Space Model adalah proses tokenisasi, *stopwords removal*, *stemming*, dilanjutkan dengan proses menentukan korelasi antar *term* dengan metode GVSM dengan merubah seluruh *term* menjadi vector orthogonal, dan hasilnya diurutkan berdasarkan tingginya similaritas dokumen. Hasil uji coba pada sistem pencarian buku dengan metode Generalized Vector Space Model ini menghasilkan nilai *precision* yang lebih rendah dari nilai *recall* dengan perbandingan 7:9. Hasil uji coba ini membuktikan bahwa sistem pencarian dengan metode GVSM ini masih efektif digunakan untuk pencarian buku di perpustakaan walaupun belum sempurna untuk dijadikan pengembangan metode sistem pencarian kedepannya.

Daftar Pustaka

- [1] Pusat Bahasa Kemdikbud, "Kamus Besar Bahasa Indonesia (KBBI)," *Kementerian Pendidikan dan Budaya*, 2016. .
- [2] A. Firman, H. Wowor, and X. Najoran, "Sistem Informasi Perpustakaan Online Berbasis Web," *E-journal Tek. Elektro dan Komput.*, 2016.
- [3] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. 2008.
- [4] M. A. P. Slamet Handoko, Idhawati Hestningsih, Achmat Ainul Afif, "Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval," *J. Inform.*, vol. 6, no. 1, pp. 599–611, 2012.
- [5] L. Robinson, "Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI," *J. Ilm. Komput. dan Inform. (KOMPUTA)*, 2014.
- [6] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. 1999.
- [7] N. K. Widyasanti, I. K. Gede, D. Putra, N. Kadek, and D. Rusjyanti, "Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia," *Merpati*, vol. 6, no. 2, pp. 119–126, 2018.
- [8] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian," in *Conferences in Research and Practice in Information Technology Series*, 2005.
- [9] P. Bagus, S. Wiguna, and B. S. Hantono, "Peningkatan Algoritma Porter Stemmer Bahasa Indonesia berdasarkan Metode Morfologi dengan Mengaplikasikan 2 Tingkat Morfologi dan Aturan Kombinasi Awalan dan Akhiran," vol. 2, no. 2, pp. 1–6, 2013.
- [10] J. Singh and S. Dwivedi, "Analysis of Vector Space Model in Information Retrieval," *IJCA Proceedings on National Conference on Communication Technologies & its impact on Next Generation Computing 2012*. pp. 14–18, 2012.
- [11] J. Pardede, "Implementasi Multithreading untuk Meningkatkan Kinerja Information Retrieval dengan Metode GVSM." 2014.
- [12] M. Abual-Rub, R. Abdullah, and N. Rashid, "A Modified Vector Space Model for Protein Retrieval," *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 9, pp. 85–89, 2007.
- [13] W. Hardi, "Mengukur Kinerja Search Engine: Sebuah Eksperimentasi Penilaian Precision And Recall Untuk Informasi Ilmiah Bidang Ilmu Perpustakaan Dan Informasi," *Visi Pustaka [National Libr. Indones.*, p. 7, 2008.