

## Pembangunan Ensiklopedia Kosa Kata Al Qur'an Menggunakan Generalized Vector Space Model dan Semantics Relatedness

Annisa Dian Muktiari<sup>1</sup>, Moch. Arif Bijaksana<sup>2</sup>, Bambang Ari Wahyudi<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>annisadian@students.telkomuniversity.ac.id, <sup>2</sup>arifbijaksana@telkomuniversity.ac.id,

<sup>3</sup>bambangari@telkomuniversity.ac.id

---

### Abstrak

Al Qur'an merupakan kitab suci bagi umat Islam dan menjadi pedoman dan sumber hukum paling utama. Al Qur'an memiliki 30 Juz, 114 Surat dan 6236 ayat. Dalam Al Qur'an terdapat ayat disetiap juz-nya yang masih kurang dipahami bagi kebanyakan orang. Dibutuhkan kamus atau ensiklopedia yang dikhususkan untuk membahas arti kata dalam Al Qur'an untuk memperoleh informasi yang lebih lengkap dengan menggunakan pedoman buku tafsir. Salah satu cara untuk mengukur keterkaitan kata dari setiap potongan kata dalam Al Qur'an adalah dengan menggunakan Generalized Vector Space Model (GVSM). GVSM merupakan metode pengembangan dari Vector Space Model (VSM) yang menambahkan fungsi sense dan penilaian pada makna antar kata dalam dokumen. Nilai yang didapat merupakan nilai *similarity* yang akan menentukan relevannya suatu dokumen dengan *query* yang dimasukkan user. Dalam penelitian ini, dokumen yang dimaksud adalah paragraf. Dari hasil pengujian yang dilakukan, metode GVSM mendapatkan nilai *similarity* yang lebih tinggi dari metode *Vector Space Model* (VSM) dan *Latent Semantics Analysis* (LSA). Dokumen yang relevan dengan *query user* akan menghasilkan nilai *similarity* diatas 0.50.

**Kata kunci :** al qur'an, tafsir, generalized vector space model, similarity.

---

### Abstract

The Qur'an is a holy book for Muslims and the most important guideline and source of law. The Qur'an has 30 Juz, 114 Letters and 6236 verses. In the Qur'an there is a verse in every juz it is still not understood for most people. It takes a dictionary or encyclopedia devoted to discussing the meaning of the word in the Qur'an for obtaining more complete information using the guidebook of the commentary. One of measure the interconnectedness of every word in the Qur'an is to use the Generalized Vector Space Model (GVSM). GVSM is a method of development of the Vector Space Model (VSM) that adds sense function and judgment to the meaning of inter-word in documents. The value obtained is a value of similarity that will be relevant to the document that the user requested. From the results of the tests, the GVSM method gets a higher similarity value than the Vector Space Model (VSM) and Latent Semantics Analyzer (LSA) methods. Relevant documents with the user query will produce a similarity value above 0.50.

**Keywords:** al qur'an, tafseer, generalized vector space model, similarity.

---

## 1. Pendahuluan

### 1.1 Latar Belakang

Al Qur'an merupakan kitab suci bagi umat Islam dan menjadi pedoman dan sumber hukum paling utama. Al Qur'an memiliki 30 Juz, 114 Surat dan 6236 ayat. Dalam Al Qur'an terdapat ayat disetiap juz-nya yang masih krang dipahami bagi kebanyakan orang. Dibutuhkan kamus atau ensiklopedia yang dikhususkan untuk membahas arti kata dalam Al Qur'an untuk memperoleh informasi yang lebih lengkap dengan menggunakan pedoman buku tafsir. Ensiklopedia berisi rangkuman informasi dari suatu ilmu pengetahuan dari semua bidang. Salah satu cara untuk yang dapat dilakukan untuk memahami isi dari Al Qur'an tersebut adalah dengan mencari arti kata dari beberapa ayat dalam buku tafsir Al Qur'an agar diperoleh informasi yang lebih lengkap.

Oleh karena itu, untuk mengatasi masalah tersebut, dalam tugas akhir ini digunakan metode Generalized Vector Space Mode (GVSM). GVSM merupakan metode pengembangan dari Vector Space Model (VSM) yang menambahkan fungsi sense dan penilaian pada makna antar kata dalam dokumen. Penggunaan GVSM ini bertujuan untuk mendapatkan nilai similaritas tiap dokumen dengan query yang di inputkan oleh user sehingga didapatkan hasil pencarian dokumen-dokumen yang relevan dengan query dari *user*.

Pada penelitian sebelumnya sudah ada yang menggunakan metode GVSM ini, namun penelitian sebelumnya tidak menggunakan tafsir Al Qur'an sebagai pedomannya. Sehingga, pada penelitian kali ini menggunakan buku tafsir Al Qur'an sebagai pedomannya. Dalam penelitian ini, buku tafsir Al Qur'an yang digunakan sebagai pedoman adalah buku Tafsir Al Misbah dari Quraish Shihab dengan menggunakan Bahasa Indonesia dan mengambil 50 kata dalam AL Qur'an sebagai dataset. Diharapkan dengan menerapkan metode GVSM ini dapat mengetahui nilai similarity yang tinggi dari kata-kata yang dimasukkan oleh user.

## 1.2 Topik dan Batasannya

Input yang diberikan adalah sebuah kata dalam Al Qur'an. Sedangkan untuk output-an dari sistem ini adalah nilai similarity tersebut dengan arti dari kata tersebut serta letak surat dan ayat dalam buku tafsir Al Misbah. Contohnya adalah kata "amanah", user akan meng-input-kan kata "amanah" tersebut, lalu sistem akan memberikan output nilai similarity dari kata tersebut, letak surat dan ayat dalam buku tafsir Al Misbah, serta arti dari kata "amanah" tersebut. Sistem akan mengeluarkan lima nilai tertinggi dari nilai similarity-nya. Dalam menyelesaikan masalah pada tugas akhir ini, terdapat beberapa batasan masalah yaitu :

- Dataset yang digunakan adalah bagian-bagian tafsir Al Misbah dari Quirash Shihab dalam bentuk buku yang menggunakan Bahasa Indonesia.
- Terdapat 50 kata Al Qur'an yang akan menjadi query user. Dan terdapat 122 dokumen berformat .txt yang berisi arti dari kata yang menjadi query tersebut.
- Input berupa kata Arab latin, dan output berupa arti kata, letak surat dan ayat, serta nilai similarity dari inputan.
- Output program menggunakan Bahasa Indonesia.

## 1.3 Tujuan

Tujuan dari tugas akhir ini adalah untuk menemukan arti kata dari sebuah *query user* dengan kumpulan dokumen tafsir Al Qur'an. Lalu, untuk mengetahui keterkaitan kata dengan menggunakan metode *Generalized Vector Space Model* melalui nilai similarity-nya serta untuk membandingkan metode GVSM dengan metode *Vector Space Model* dan *Latent Semantic Analysis*.

## 2. Studi Terkait

### 2.1 Tafsir Al Misbah

Tafsir Al Misbah merupakan sebuah tafsir Al Qur'an 30 juz pertama yang turunan dalam kurun waktu 30 tahun terakhir. Quraish Shihab memberi warna yang menarik dan khas serta sangat relevan untuk memperkaya pemahaman umat islam terhadap makna ayat-ayat Allah SWT. Tafsir ini menggunakan gaya penulisan yang mudah dicerna oleh semua kalangan. Pada penelitian ini, menggunakan taafsir Al Misbah volume 1-10 dengan menggunakan Bahasa Indonesia dan mengambil 50 kata sebagai dataset.

### 2.2 Ensiklopedia

Ensiklopedia adalah kumpulan penjelasan dari banyak kata-kata yang memiliki informasi secara luas, lengkap dan tersusun berdasarkan abjad. Ensiklopedia merupakan perkembangan dari kamus, namun berbeda dengan ensiklopedia yang memiliki sumber pengetahuan yang dapat ditemukan didalamnya disemua bidang pengetahuan, kamus hanya berisi definisi suatu kata berdasarkan sudut pandang linguistik. Pada penelitian ini menggunakan ensiklopedia karena akan memberikan penjelasan dari arti kata Al Qur'an secara lengkap.

### 2.3 Text Processing

Suatu proses pengubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan, untuk proses mining yang lebih lanjut. Singkatnya merubah teks menjadi *term index*. Terdapat beberapa tahapan dalam proses ini, yaitu stopword removal, stemming, dan tokenisasi.

#### 2.3.1 Stopword Removal

Pada proses ini merupakan proses untuk menghapus atau membuang kata yang tidak penting atau kata yang tidak memiliki makna yang sering muncul dalam dokumen. Salah satu contoh stopword removal adalah kata depan (di, ke, dari), dan kata penghubung (dan, atau, tapi).

**Tabel 1 Contoh Stopword Removal**

Sebelum Stopword Removal	Setelah Stopword Removal
Bencana yang mereka duga tidak akan terjadi itu dipahami sebagai bencana duniawi karena, sejak semula, mereka telah menganggap ringan dan sebentar siksa ukhrawi	Bencana mereka duga terjadi dipahami bencana duniawi, sejak semula, mereka telah menganggap ringan siksa ukhrawi.

### 2.3.2 Stemming

Setelah proses stopword removal, selanjutnya dilakukan proses mengubah kata yang memiliki imbuhan menjadi kata dasar. Contoh stemming adalah “membetulkan” menjadi “betul”, “makanan” menjadi “makan”.

**Tabel 2 Contoh Stemming**

Sebelum Stemming	Setelah Stemming
Bencana mereka duga <b>terjadi dipahami</b> bencana duniawi, sejak <b>semula</b> , telah mereka <b>mengganggap</b> ringan siksa ukhrawi.	Bencana mereka duga jadi paham bencana duniawi, sejak mula, telah mereka anggap ringan siksa ukhrawi.

### 2.3.3 Tokenisasi

Selanjutnya dilakukan proses tokenisasi yaitu mengubah kalimat atau paragraf menjadi term-term. Selain itu, spasi digunakan sebagai pemisah antar kata tersebut.

**Tabel 3 Contoh Tokenisasi**

Sebelum Stemming	Setelah Stemming
Bencana mereka duga jadi paham bencana duniawi, sejak mula, telah mereka anggap ringan siksa ukhrawi.	Bencana Mereka Duga Jadi Paham Bencana Duniawi Sejak Mula Telah Mereka Anggap Ringan Siksa Ukhrawi

## 2.4 Pembobotan Term Frequency-Inverse Document Frequency (TF-IDF)

Pada proses ini, dilakukan untuk mencari TF dari suatu term dan melakukan pembobotan term. TF adalah frekuensi dari kemunculan sebuah term dalam dokumen.<sup>[1]</sup> Perhitungan pembobotan yang sering digunakan adalah TF-IDF. Adapun rumusnya TF-IDF sebagai berikut :

$$Wdt = TF \times IDF$$

$$Wdt = TF \times \log \frac{n}{df}$$

Keterangan :

Wdt : bobot term atau kata terhadap dokumen

TF : jumlah kemunculan term dalam dokumen

n : jumlah semua dokumen yang terdapat dalam dataset

df : jumlah term yang ada dalam semua dokumen

Tabel 4 Contoh Perhitungan TF-IDF

Term	Tf				Idf ( $\log \frac{n}{df}$ )	Wdt		
	d1	d2	d3	df		d1	d2	d3
Bencana	1	0	1	2	0.176	0.176	0	0.176
Mereka	1	0	1	2	0.176	0.176	0	0.176
Duga	1	0	0	1	0.477	0.477	0	0
Paham	1	0	0	1	0.477	0.477	0	0
Duniawi	0	1	0	1	0.477	0	0.477	0
Anggap	0	2	0	1	0.477	0	0.954	0
Ringan	0	1	1	2	0.176	0	0.176	0.176
Siksa	0	1	1	2	0.176	0	0.176	0.176

Tf merupakan *term frequency* yaitu jumlah frekuensi kata yang muncul pada setiap dokumen. Untuk d1, d2, dan d3 merupakan representasi dari dokumen. Dokumen tersebut berupa paragraf dalam format file .txt yang telah dibuat sebelumnya. Sedangkan untuk df merupakan representasi untuk jumlah keseluruhan kata yang muncul pada semua dokumen.

## 2.5 GVSM

Pada tugas akhir ini akan menggunakan metode GVSM untuk mendapatkan nilai similarity dari masing-masing dokumen. GVSM mempertimbangkan kedekatan sense antar term yang lebih akurat untuk mempresentasikan dokumen. Ada beberapa proses untuk mendapatkan hasil dari query yang dimasukkan oleh user menggunakan metode Generalized Vector Space Model, seperti :

2.5.1 Melakukan proses text processing terlebih dahulu.

2.5.2 Menentukan minterns untuk menentukan kemungkinan pola frekuensi kata. Panjang minterns ini didasarkan pada banyak kata yang diinputkan pada query lalu diubah menjadi vektor orthogonal. Pola yang akan muncul seperti :

$$m1 = (0, 0, 0, \dots)$$

$$m2 = (1, 0, 0, \dots)$$

...

2.5.3 Menghitung banyaknya frekuensi atau kemunculan kata dalam kumpulan dokumen yang sesuai dengan query.

2.5.4 Menghitung index term dengan rumus :

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

Keterangan :

$\vec{k}_i$  : index term ke-i

$\vec{m}_r$  : vektor orthogonal sesuai pola mintern

$c_{i,r}$  : faktor korelasi antara korelasi antara index term i dengan mintern r

Untuk menghitung faktor korelasi sebagai berikut :

$$c_{i,r} = \sum_{d_j | g_i(\vec{d}_j) = g_i(m_r)} w_{i,j}$$

Keterangan :

$C_{i,r}$  : faktor korelasi antara 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  $m_r$

2.5.5 Mengubah dokumen dan query menjadi vektor

$$\vec{d}_j = \sum_{i=1}^n w_{ij} \times \vec{k}_i \qquad \vec{q} = \sum_{i=1}^n q_i \times \vec{k}_i$$

Keterangan :

$\vec{d}_j$  : 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

$\vec{k}_i$  : index term

n : jumlah term index

2.5.6 Mengurutkan dokumen berdasarkan similaritas menggunakan perkalian vektor

$$sim(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \bullet \vec{q}}{\|\vec{d}_j\| \|\vec{q}\|}$$

Keterangan :

$\vec{d}_j$  : vektor dokumen j

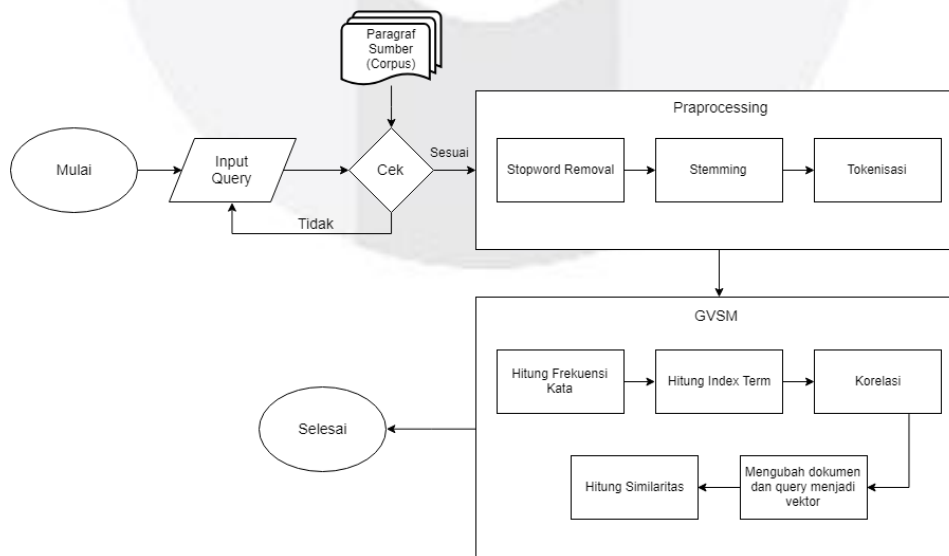
$\vec{q}$  : vektor query

Dari proses diatas akan menghasilkan nilai similarity dari dokumen yang paling relevan dengan query user. Nilai similarity akan diurutkan dari yang paling tinggi nilainya.

### 3. Sistem yang Dibangun

#### 3.1 Gambaran Umum Sistem

Pada penelitian ini, keluaran yang diharapkan berupa dokumen yang relevan dengan query user. Proses untuk mendapatkan dokumen yang relevan dimulai dari *user* yang melakukan *input* sebuah query berupa kata dalam Al Qur'an. Kemudian sistem akan melakukan pengecekan apakah query tersebut sesuai dengan corpus yang ada atau tidak. Jika query tersebut tidak ada dalam corpus maka user harus input kembali query yang lain. Jika query user sesuai, maka proses akan berlanjut ke tahap *praprocessing*. Kemudian tahap selanjutnya adalah menghitung dengan metode GVSM. Setelah semua proses dalam metode GVSM selesai, maka proses telah selesai dengan hasil nilai similarity dari tiap dokumen yang muncul. Alur sistem sesuai dengan gambar 1 seperti dibawah.



Gambar 1 Gambar Alur Sistem

Dari alur sistem diatas dapat dijelaskan bahwa :

1. User memasukkan *query* berupa kata.
2. Tahap selanjutnya adalah melakukan pengecekan apakah *query* yang dimasukkan sesuai dengan corpus atau tidak. Jika *query user* sesuai maka akan masuk ke proses selanjutnya, jika tidak sesuai maka user harus memasukkan *query* yang lain.
3. Jika *query* sesuai, maka akan masuk ke tahap *praprocessing*. *Query* yang dimasukkan *user* akan di *praprocessing* dan akan melalui beberapa tahap yang ada dalam *praprocessing*, yaitu :
  - a. Stopword removal, menghilangkan kata yang tidak penting dan kata penghubung.
  - b. Stemming, mengubah kata yang berimbuhan menjadi kata dasar.
  - c. Tokenisasi, memisahkan kata berdasarkan spasi.
4. Setelah tahap *praprocessing* selesai, maka akan masuk ke tahap perhitungan menggunakan metode GVSM. Dalam GVSM juga terdapat beberapa tahap di dalamnya, seperti :
  - a. Menghitung frekuensi kata yang muncul. *Query* akan dihitung jumlah kemunculannya dalam tiap dokumen yang ada.
  - b. Menentukan korelasi untuk setiap term.
  - c. Menghitung index term.
  - d. Lalu mengubah dokumen dan *query* ke dalam bentuk vektor
  - e. Menghitung nilai *similarity*.
5. Setelah semua tahap dilakukan maka dokumen yang relevan dengan *query user* akan muncul beserta dengan nilai *similarity*-nya. Dokumen yang memiliki nilai *similarity* yang paling tinggi akan menempati posisi teratas, dan menjadi dokumen yang paling relevan dengan *query user*.

### 3.2 Analisis Sistem

#### 3.2.1 Analisis Masukan

Data masukan yang digunakan dalam penelitian ini adalah :

1. Korpus tafsir dalam Bahasa Indonesia yang letak surat dan ayatnya sudah dibuat sebelumnya.
2. Kata dan arti tafsir diambil secara acak untuk dijadikan korpus.

#### 3.2.2 Analisis Sistem

Proses yang terjadi pada penelitian ini adalah :

1. Proses untuk mengumpulkan dokumen korpus yang akan dijadikan dataset.
2. *Praprocessing*, ada tiga tahap pada *praprocessing* ini, yaitu stopword removal (membuang kata yang tidak penting, kata depan, dan kata penghubung), stemming (mengubah kata yang berimbuhan menjadi kata dasar), tokenisasi (mengubah kalimat atau paragraf menjadi term-term), dan menghitung jumlah kemunculan kata pada tiap dokumen yang ada.
3. Menerapkan model ruang vektor. Setiap dokumen dan *query* akan direpresentasikan oleh vektor dimensi. Satu term mewakili setiap dimensi pada vektor tersebut. Menghitung nilai *similarity* untuk mendapatkan dokumen yang paling relevan.
4. Pencarian arti kata, mencari arti kata dalam tafsir yang sesuai dengan *query user*.

#### 3.2.3 Analisis Output

Output yang diharapkan pada penelitian ini adalah *user* dapat menemukan dokumen apa saja yang relevan dengan *query* yang dimasukkan. Dokumen tersebut berupa paragraf dalam format file .txt yang telah dibuat sebelumnya. Dokumen yang paling relevan dengan *query user* akan berada di posisi paling atas dengan nilai *similarity*-nya yang paling tinggi. Contoh output-an yang diharapkan adalah ketika *user* memasukkan *query* berupa kata “amanah” maka output yang dihasilkan berupa kumpulan dokumen tafsir yang mengandung kata “amanah” tersebut beserta letak surat, ayat, dan paragrafnya dengan format “surat-ayat-paragraf”.

#### 4. Evaluasi

##### 4.1 Pengujian Sistem

Pada penelitian ini, sistem yang dibangun akan menghasilkan perhitungan relevansi dokumen tafsir dan *query user*. Pengujian menggunakan 50 *query* dan 122 dokumen tafsir. Berikut adalah contoh *query* dan dokumen tafsir dapat dilihat dari Tabel 5, dan selengkapnya terdapat pada lampiran.

**Tabel 5 Contoh Query/Kata dan Dokumen tafsir yang Relevan**

No	Kata/Query	Penjelasan	Kandungan dalam Surat
1	Dafiq	<p>Menurut Ibnu Zakaria, mempunyai satu arti denotatif yaitu mendorong sesuatu dari belakang.</p> <p><b>Al Ashfahani menyebutkan makna <i>dafiq</i> adalah mengalir dengan cepat. Thabathabai mengartikan sebagai meluap dan mengalirnya air sekaligus dan secara cepat. Al Qurthubi memberikan arti kata tersebut sebagai dari air mani (sperma) yang dicurahkan ke dalam rahim perempuan (istri).(QS. Ath Thariq 86:6)</b></p>	<b>QS. Ath Thariq 86:6</b>
2	Lakum	<p>Memang Nabi Muhammad SAW sangat ingin agar seluruh manusia beriman kepada Allah. Usaha beliau sedemikian keras, sampai-sampai Allah berfirman menggambarkan kasih-Nya.</p> <p><b>Kata <i>lakum/kepada kaum</i> pada ayat ini sepintas tidak diperlukan tetapi sebenarnya ia dicantumkan untuk menjadikan kata <i>yu'мину/percaya</i> mengandung makna mengakui. (QS. Al Baqarah 2:75)</b></p>	<b>QS. Al Baqarah 2:75</b>

Dapat dilihat pada Tabel 5, terdapat kalimat atau paragraf yang bercetak tebal yang merupakan arti kata yang relevan dengan *query* atau pada kalimat tersebut mengandung kata yang user masukkan. Pada kolom “kandungan dalam Surat” merupakan letak arti dari kata atau *query* yang user masukkan.

##### 4.2 Hasil Pengujian

Hasil pengujian ini dapat dilihat pada tabel 6 dibawah ini. Pada penelitian ini, sistem mengeluarkan dokumen yang sesuai dengan *query user*. Dengan total dokumen yang ditampilkan sebanyak lima nilai *similarity* tertinggi.

**Tabel 6 Hasil pencarian dari beberapa query**

Kata Kunci	Semua Dokumen Hasil Pencarian	Nilai Similarity	Dokumen yang Relevan dari Hasil Pencarian	Jumlah Dokumen yang Relevan
Amanah	Dok 8-27-2 Dok 2-283-2 Dok 70-32-2 Dok 63-41-1 Dok 8-27-1	<b>0.88</b> <b>0.82</b> <b>0.67</b> 0.24 0.17	Dok 8-27-2 Dok 2-283-2 Dok 70-32-2	3 dokumen
Fath	Dok 5-52-2 Dok 61-13-2 Dok 10-11-1 Dok 11-52-1 Dok 25-46-1	<b>0.90</b> <b>0.69</b> 0.15 0.14 0.12	Dok 5-52-2 Dok 61-13-2	2 dokumen

Samiyyan	Dok 19-7-2	<b>0.95</b>	Dok 19-7-2	1 dokumen
	Dok 19-8-1	0.29		
	Dok 19-7-1	0.29		
	Dok 13-8-2	0.15		
	Dok 75-28-1	0.14		
Juruzan	Dok 18-8-2	<b>0.98</b>	Dok 18-8-2	1 dokumen
	Dok 13-8-1	0.17		
	Dok 6-99-1	0.17		
	Dok 61-13-1	0.11		
	Dok 18-8-1	0.11		

Dapat dilihat pada Tabel 6, ketika *user* memasukkan kata “fath” maka sistem akan mengeluarkan lima dokumen yang dianggap relevan. Namun dokumen yang paling relevan dengan *query user* hanya dua dokumen saja, karena dokumen tersebut mengandung kata “fath”. Penjelasan selengkapnya terdapat pada lampiran.

Selanjutnya, pengujian dilakukan dengan membandingkan metode GVSM dengan metode Vector Space Model (VSM) dan Latent Semantics Analysis. Berikut tabel pengujiannya :

**Tabel 7 Tabel Perbandingan Metode GVSM, VSM, dan LSA**

Kata Kunci	Nilai Similarity			Jumlah Dokumen yang Relevan
	GVSM	VSM	LSA	
Amanah	0.88	0.70	0.70	3 dokumen
Fath	0.90	0.57	0.64	2 dokumen
Samiyyan	0.95	0.57	0.73	1 dokumen
Juruzan	0.98	0.57	0.68	1 dokumen

Dapat dilihat dari Tabel 7 diatas, bahwa sistem pencarian dengan metode GVSM memiliki nilai *similarity* yang lebih tinggi dibandingkan metode VSM dan LSA.

#### 4.2 Analisis Hasil Pengujian

Pada Tabel 6 dapat dilihat hasil nilai *similarity* antara *query* yang dimasukkan oleh *user* dengan semua dokumen yang ada dalam sistem. Dokumen yang relevan memiliki nilai *similarity* antara 0 sampai 1. Jika nilai semakin mendekati 1 maka dokumen tersebut yang paling relevan dengan *query* yang dimasukkan *user* dan menampilkan lima dokumen dengan nilai *similarity* paling tinggi. Namun jika *query* yang dimasukkan oleh *user* tidak sesuai maka nilai *similarity* nya tidak akan keluar, karena dalam dokumen tidak terdapat kata tersebut. Dilihat dari nilai *similarity* pada Tabel 6, jika nilai *similarity*-nya diatas 0.50 maka dapat dikatakan dokumen tersebut relevan dengan *query* yang dimasukkan. Namun jika nilai *similarity*-nya dibawah 0.50, dokumen tersebut tidak relevan dengan *query* yang dimasukkan. Batas nilai *similarity* diambil dari pengujian yang telah dilakukan untuk semua *query*, dan menghasilkan nilai *similarity* untuk *query* yang relevan dengan dokumen tafsir adalah 0.50, sehingga nilai tersebut menjadi batas nilai relevan atau tidaknya dokumen tersebut dengan *query user*.

Pada Tabel 7 merupakan tabel perbandingan metode GVSM dengan metode VSM dan LSA. Dengan membandingkan nilai *similarity* masing-masing metode. Nilai *similarity* dari metode GVSM lebih tinggi dibandingkan dengan metode VSM dan LSA dikarenakan pada metode GVSM terdapat perhitungan korelasi antar *query* dan dokumen.

#### 5. Kesimpulan

Pada penelitian ini sistem dapat melakukan pencarian arti kata dalam Al Qur'an melalui tafsir AL Misbah dari sebuah kata kunci dari *user*. Dari Tabel 6 dapat disimpulkan bahwa sistem akan mengeluarkan lima dokumen yang dianggap relevan. Namun yang benar-benar relevan dengan kata kunci dari *user* yang memiliki nilai



*similarity* diatas 0.50. Jika nilai *similarity*-nya dibawah 0.50 maka dokumen tersebut tidak relevan dengan *query user*. Lalu, Tabel 7 dapat dilihat bahwa dengan menggunakan metode GVSM akan mendapatkan nilai *similarity* yang lebih tinggi dibandingkan dengan metode VSM dan LSA.

### Daftar Pustaka

- [1] Hendra Bunyamin, C. P. (t.thn.). Aplikasi Information Retrieval (IR) CATA Dengan Metode.
- [2] Logi, I. (2017, Juli 11). Pembobotan Kata atau Term Weighting TF-IDF.
- [3] Rahayu, S. C. (t.thn.). Implementasi Generalized Vector Space Model pada Pencarian Produk Berbasis Opini Produk.
- [4] Robinson, L. (t.thn.). Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI.
- [5] Setyawan, A. (2016, Juli 18). Perbedaan Kamus dengan Ensiklopedia.
- [6] Wikipedia. (2017, Desember 2). Tafsir Al-Misbah.