

APLIKASI SISTEM TEMU KEMBALI ANGGKET MAHASISWA MENGGUNAKAN METODE GENERALIZED VECTOR SPACE MODEL

Suprianto¹, Sunardi², Abdul Fadlil³

¹ Program Studi Sistem Informasi, STMIK PPKIA Tarakanita Rahmawati, ² Program Studi Teknik Elektro, Universitas Ahmad Dahlan, ³ Magister Teknik Informatika, Universitas Ahmad Dahlan
Email: ¹supri@ppkia.ac.id, ²sunardi@mti.uad.ac.id, ³abdulfadlil@mti.uad.ac.id

(Naskah masuk: 26 Oktober 2018, diterima untuk diterbitkan: 4 Desember 2018)

Abstrak

Banyak hal yang dapat dilakukan untuk memajukan sebuah perguruan tinggi, salah satunya adalah dengan melakukan evaluasi terhadap angket Mahasiswa pada setiap semester. Salah satu perguruan tinggi yang ada di Kota Tarakan Kalimantan Utara adalah STMIK PPKIA Tarakanita Rahmawati. Banyaknya data yang terdapat pada angket mahasiswa PPKIA membuat pengguna kesulitan menemukan informasi yang sesuai dengan kata kunci yang diberikan. Angket mahasiswa berisi penilaian mahasiswa terhadap pengajaran dosen, pelayanan administrasi dan fasilitas kampus yang dibuat dalam bentuk form yaitu memilih grade nilai dari sangat tidak baik sampai dengan sangat baik. Selain itu juga terdapat penilaian dalam bentuk esay yaitu berupa saran dan komentar. Pengisian angket dilakukan pada akhir semester berjalan. Penelitian ini bertujuan untuk menemukan informasi data angket yang relevan terhadap kata kunci. Aplikasi dibangun berbasis web dengan bahasa pemrograman PHP. Aplikasi yang dibuat hanya menggunakan basisdata masih mempunyai kekurangan yaitu tidak dapat mengurutkan dokumen sesuai dengan kata kunci, dikarenakan pengurutan dokumen hanya berdasarkan urutan dokumen pada basisdata saja. Dengan memanfaatkan teknik *information retrieval* (IR) yang diterapkan pada aplikasi, pengguna akan sangat terbantu dalam menemukan informasi yang dibutuhkan. Aplikasi yang dibuat dapat menampilkan dan mengurutkan dokumen yang paling mirip dengan kata kunci. Aplikasi dibangun dengan metode Generalized Vector Space Model (GVSM) sebagai dasar untuk menyelesaikan permasalahan yang ada. Metode GVSM adalah IR atau biasa disebut sistem temu kembali untuk mencocokkan *term* atau kata dari kata kunci yang digunakan. Dari hasil uji coba terhadap 5 kata kunci diperoleh nilai precision sebesar 72% dan recall sebesar 100% dengan waktu proses selama 34.4 detik.

Kata kunci: *angket mahasiswa, aplikasi, generalized vector space model*

APPLICATION OF INFORMATION RETRIEVAL FOR OPINION STUDENT QUESTIONNAIRE USING GENERALIZED VECTOR SPACE MODEL METHODS

Abstract

Many things can be done to advance a university, one of which is by evaluating the Student questionnaire every period. One of the universities in Tarakan City, North Kalimantan is STMIK PPKIA Tarakanita Rahmawati. The large amount of data contained in PPKIA student questionnaires makes it difficult for users to find information that matches the given keywords. Student questionnaires contain student assessments of lecturer teaching, administrative services and campus facilities that are made in the form of selecting grades from very bad to very good. In addition there are also assessments in the form of essays, in the form of suggestions and comments. The questionnaire will be filled in at the end of the period. This study aims to find questionnaire data information that is relevant to the keyword. The application is built web-based with the PHP programming language. Applications that are made using only databases still have a disadvantage of not being able to sort documents according to keywords, because sorting documents is only based on the order of documents on the database only. By utilizing information retrieval (IR) techniques that are applied to the application, users will be very helpful in finding the information needed. The application created can display and sort documents that are most similar to keywords. Applications are built with the Generalized Vector Space Model (GVSM) method as a basis for solving existing problems. The GVSM method is IR or commonly called a retrieval system to match the terms or words of the keywords used. From the results of trials on 5 keywords, the precision value of 72% and recall of 100% were obtained with a processing time of 34.4 seconds.

Keywords: *student questionnaire, application, generalized vector space model*

1. PENDAHULUAN

Teknologi komputer saat ini berkembang dengan sangat pesat. Biaya yang terjangkau dengan spesifikasi yang tinggi mampu mengatasi masalah dari yang mudah sampai yang paling sulit. Dengan adanya teknologi ini, kita dapat melakukan banyak hal, diantaranya adalah mencari data angket mahasiswa. Data angket mahasiswa adalah kuisioner yang diisi oleh mahasiswa pada saat sebelum ujian akhir semester dilakukan. Hal ini dilakukan untuk menilai proses pengajaran, pelayanan dan fasilitas yang telah dilakukan selama satu semester berjalan. Banyaknya angket mahasiswa sama dengan banyaknya jumlah mahasiswa, oleh karena itu untuk melakukan pencarian data sesuai kebutuhan pengguna menjadi sulit dan lama. Terdapat banyak kelebihan yang terdapat pada komputer, salah satunya adalah dapat digunakan untuk membangun sebuah aplikasi.

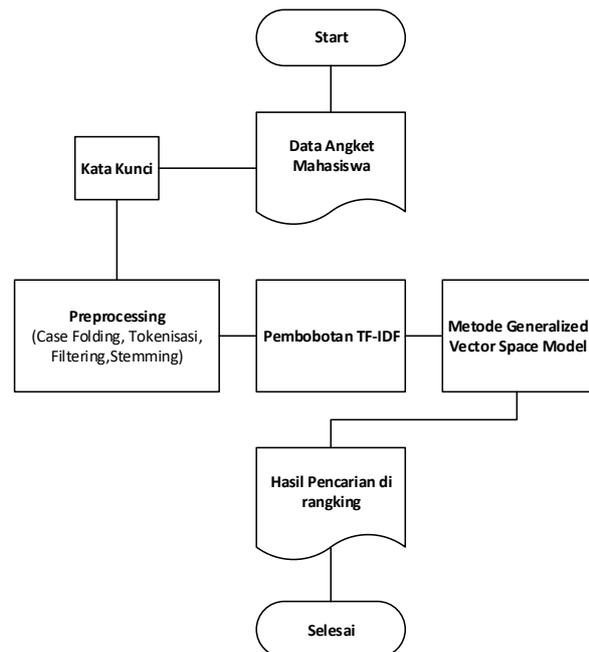
Dari menganalisis permasalahan yang ada, dapat dibangun aplikasi untuk sistem temu kembali opini angket mahasiswa menggunakan metode *generalized vector space model* digunakan untuk mencocokkan tingkat kemiripan dokumen dengan kata kunci yang dicari oleh pengguna. Menurut Dedi, sistem temu kembali informasi (*information retrieval system*) adalah bagian dari ilmu komputer yang berkaitan dengan pengambilan informasi dari dokumen-dokumen berdasarkan isi dari dokumen-dokumen itu sendiri (Dedi & Khusaeri, 2015). *Generalized vector space model* adalah perkembangan dari metode *vector space model* (VSM) yang menambahkan fungsi *sense* dan penilaian terhadap hubungan makna antar *term* (kata atau kumpulan kata merepresentasikan suatu pengertian) dengan lebih akurat. Menurut fatkhul perkembangan penelusuran informasi saat ini menghasilkan *recall* yang tinggi dan *precision* yang rendah (Fatkhul, 2012). Menurut Lamhot aplikasi IR dengan metode GVSM bekerja dengan baik pada jumlah dokumen sedikit maupun pada jumlah dokumen yang banyak (Lamhot, 2015). Menurut Jasman GVSM terjadi dalam dua proses yaitu proses *preprocessing* dan proses menghitung nilai relevansi antara dokumen dan kata kunci berdasarkan nilai *similaritasnya* (Jasman, 2014). Jorg dkk, menyajikan dua pendekatan untuk pencarian semantik dengan memasukkan anotasi dokumen tertaut ke dalam *generalized vector space model* (Jorg dkk, 2015). Menurut (Hendra & Chatlahea, 2015) Proses metode Generalized Vector Space Model didahului dengan memasukkan kata kunci, mengolah kata-kata termasuk pembuangan kata-kata penghenti dan stemming. Aplikasi yang dibuat melakukan pencarian dokumen yang *relevan* dengan pertanyaan, berdasarkan kesamaan. Hasil pencarian yang dipesan berdasarkan nilai kesamaan tertinggi.

Dengan adanya aplikasi ini, diharapkan dapat membantu dalam menyelesaikan masalah dalam

melakukan pencarian data yang dibutuhkan pengguna pada angket mahasiswa. Pada penelitian ini penulis menggunakan pemrograman berbasis WEB dengan bahasa pemrograman PHP (*Personal Home Page* atau *Hypertext Preprocessor*) dan *database MySQL*.

2. METODE PENELITIAN

Penelitian ini memiliki metode penelitian sebagai berikut pada gambar 1.



Gambar 1. Metodologi Penelitian

Pada gambar 1, data angket Mahasiswa merupakan data set yang digunakan dalam pencarian informasi berdasarkan kata kunci yang diberikan. Kata kunci dilakukan proses *preprocessing* yang meliputi *case folding*, *tokenisasi*, *filtering*, dan *stemming* dan kemudian dilakukan proses perhitungan *TF-IDF*. Selanjutnya hasil yang didapatkan digunakan untuk melakukan perhitungan dengan metode *generalized vector space model*. Hasil akhir yang didapatkan yaitu hasil pencarian yang telah diurutkan dari yang paling tinggi nilainya berdasarkan kata kunci yang dimasukkan.

2.1. Preprocessing

Proses teks mining membutuhkan tahapan *preprocessing*, karena dokumen atau teks yang akan dilakukan proses teks mining terdapat simbol-simbol, imbuhan (awalan dan akhiran), karakter kapital, dan kata yang sering muncul (*stopword*). Tahapan *preprocessing* dilakukan pada dokumen yang akan dicari dan *query* pencarian yang pengguna masukkan. Ada beberapa tahapan *preprocessing* yaitu *case folding*, *tokenisasi*, *filtering*, dan *stemming*.

2.1.1. Case Folding

Case Folding merupakan sebuah tahapan yang digunakan untuk merubah teks yang awalnya huruf besar menjadi huruf kecil semua. Tahapan ini dilakukan agar proses *teks mining* menjadi lebih mudah.

2.1.2. Tokenisasi

Tokenisasi ada sebuah tahapan yang digunakan untuk memisahkan atau menghilangkan *string input* berdasarkan setiap kata penyusunnya atau memisahkan setiap kata yang tersusun dalam suatu dokumen. Bagian yang dihilangkan dapat berupa angka, karakter atau simbol, dan tanda baca selain dari huruf alfabet, hal ini dilakukan karena karakter tersebut tidak memiliki pengaruh dalam pemrosesan suatu teks.

2.1.3. Filtering

Filtering atau *stop word removal* yaitu proses untuk menghilangkan kata yang terdapat pada dokumen yang tidak *relevan* dengan cara menggunakan *stoplist*. *Stoplist* berisi kumpulan kata yang tidak *relevan* atau tidak digunakan dalam pemrosesan bahasa alami.

2.1.4. Stemming

Stemming merupakan tahapan proses lanjutan setelah *filtering* yang digunakan untuk membuang imbuhan awalan atau akhiran menjadi kata dasar. *Library* yang digunakan pada program aplikasi ini adalah *stemming* Arifin Setiono.

2.2. TF-IDF

TF-IDF adalah suatu metode pembobotan kata yang umum digunakan untuk melakukan pencarian kata atau *information retrieval*. Menurut dimas dkk, *TF-IDF* merupakan suatu proses *weighting*/pembobotan dimana akan dilakukan perhitungan *weight*/bobot terhadap setiap term *indeks* yang dihasilkan pada tahap *text preprocessing* (Dimas dkk, 2016). Metode ini melakukan perhitungan *term frequency* dan *inverse document frequency* pada setiap kata yang ada di pada setiap dokumen. Metode ini menggunakan rumus pada persamaan 1 dan 2. Berikut rumus *IDF* yaitu: (Abdul, 2015).

$$IDF = \log_2(D/df) \quad (1)$$

Dimana:

D : jumlah dokumen

df : jumlah dokumen yang memiliki kata yang dicari

$$W_{dt} = tf_{df} * IDF_t \quad (2)$$

Dimana:

W : bobot dokumen ke-d terhadap kata yang dicari pada sebuah dokumen

d : dokumen ke-d

t : kata ke-t dari kata kunci

tf : banyak kata yang dicari pada sebuah dokumen

IDF: *Inversed Document Frequency*

2.3. Metode Generalized Vector Space Model

Metode generalized vector space model adalah sebuah metode yang menggunakan konsep dari ruang *vektor*. Kata kunci yang dimasukkan oleh pengguna dan kumpulan dokumen yang tersedia diubah menjadi *vektor-vektor* yang kemudian dilakukan operasi perkalian titik dan hasilnya akan menjadi acuan yang digunakan untuk menemukan relevansi dari kata kunci yang di masukkan terhadap dokumen. Pada tahun 1985, Wong, dkk menyajikan suatu alternatif dari *IR VSM*, Wong berpendapat bahwa *GVSM* menghindari asumsi yang ada dengan menggunakan dokumen-dokumen sebagai dasar ruang dan *vektor* dari pada *term* atau kata kunci (Wong, 1985).

Menurut Baeza, metode *generalized vector space model* dapat digunakan untuk menemukan hasil dari pencarian berdasarkan kata kunci yang dimasukkan dengan langkah sebagai berikut (Baeza,1999):

1. Kata depan dan penghubung dihilangkan.
2. Menggunakan stemming untuk membuang kata dasar atau imbuhan.
3. Membuat *miniterm* sesuai kata kunci yang diinputkan, kemudian di ubah menjadi vector orthogonal sesuai pola panjangnya kata kunci. Contohnya sebagai berikut :
 $M_1 = (0,0)$
 $M_2 = (1,0)$
 ...
 $M_n = (0,0)$
4. Melakukan perhitungan kemunculan kata yang terdapat pada dokumen berdasarkan kata kunci.
5. Perhitungan *index term* dapat dinyatakan pada persamaan 3.

$$\vec{k}_1 = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \vec{m}_r}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c_{i,r}^2}} \quad (3)$$

Dimana:

\vec{k}_1 : *index term* ke-1

\vec{m}_r : *vector orthogonal* sesuai pola *miniterm*

$C_{i,r}$: *faktor korelasi* antara *index term* ke-i dengan *miniterm* r

kemudian *faktor korelasi* dapat dinyatakan pada persamaan 4.

$$C_{i,r} = \sum_{d_j | g_j(\vec{a}_j) = g_i(m_r)} w_{i,j} \quad (4)$$

Dimana:

$C_{i,r}$: 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

- Melakukan pengubahan kata kunci dan dokumen dengan persamaan 5 dan 6.

$$\vec{d}_j = \sum_{i=1}^n w_{ij} x_{k_i} \quad (5)$$

$$\vec{q} = \sum_{i=1}^n q_i x_{k_i} \quad (6)$$

Dimana:

\vec{d}_j : vektor dokumen ke-j

\vec{q} : vektor query

w_{ij} : berat *index term* i pada dokumen j

q_i : berat *index term* pada query i

x_{k_i} : *index term*

n : jumlah *index term*

- Langkah terakhir yaitu melakukan perhitungan vektor untuk mengurutkan dokumen berdasarkan tingkat persamaan atau *similaritas* dengan persamaan 7.

$$sim(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| |\vec{q}|} \quad (7)$$

Dimana:

\vec{d}_j : vektor dokumen ke-j

\vec{q} : vektor query

2.4. Recall dan Precision

Recall dan *Precision* digunakan untuk mengukur efektifitas kinerja dari suatu IR. *Recall* adalah perbandingan antara jumlah dokumen *relevan* yang ditampilkan terhadap jumlah seluruh dokumen yang *relevan*. Perhitungan *recall* dinyatakan pada persamaan 8.

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} \quad (8)$$

Precision adalah perbandingan antara jumlah dokumen *relevan* yang ditampilkan terhadap jumlah seluruh dokumen yang ditampilkan. Perhitungan *Precision* dinyatakan pada persamaan 9.

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} \quad (9)$$

Penotasian *precision* dan *recall* dalam bentuk *contingency table* ditunjukkan pada tabel 1.

Tabel 1. *Contingency Table Precision dan Recall*

	Relevant	Non Relevant
Retrieved	True Positives (tp)	False Positives (fp)
Non Retrieved	False Negatives (fn)	True Negatives (tn)

Dari tabel 1 diperoleh sebuah rumus untuk menghitung *Precision* dan *Recall*, seperti yang ditunjukkan pada persamaan 10 dan 11.

$$Precision = \frac{tp}{(tp+fp)} \quad (10)$$

$$Recall = \frac{tp}{(tp+fn)} \quad (11)$$

Kedua ukuran di atas biasanya diberi nilai dalam bentuk prosentase, 1 sampai 100% (Dwiyantoro, 2017).

3. HASIL DAN PEMBAHASAN

3.1 Perhitungan Generalized Vector Space Model

Berdasarkan hasil perhitungan menggunakan algoritma *GVSM*, terdapat sebuah *query* atau kata kunci (KK) serta 5 dokumen yaitu D1, D2, D3, D4, D5 sebagai berikut:

KK : Terlambat mengajar

D1 : Bahasa Pengajaran dipermudah, tidak lambat memberikan materi.

D2 : Metode mengajar sudah sangat baik dan mudah dimengerti.

D3 : Dosen sudah tidak pernah terlambat mengajar.

D4 : Disarankan agar dosen tidak terlambat masuk kelas.

D5 : Materi yang diajarkan lebih lambat dari kelas lain.

Setelah dilakukan tahap *case folding* dan *tokenizing*, serta *filtering*, dan *stemming*, maka dihasilkan kata sebagai berikut pada tabel 2.

Tabel 2. <i>Term</i>		
No.	Term	Dokumen
1	lambat	D1, D3, D4, D5
2	ajar	D1, D2, D3, D5
3	bahasa	D1
4	mudah	D1, D2
5	beri	D1
6	materi	D1
7	saran	D4, D5
8	dosen	D3, D4, D5
9	masuk	D4, D5
10	kelas	D4, D5

Pada tabel 2, terdapat 10 (sepuluh) *term* yang dihasilkan oleh proses *preprocessing* dari kata kunci dan 5 (lima) dokumen. *Term* "lambat" terdapat pada dokumen D1, D3, D4, dan D5. Dari kesepuluh *term* yang terbentuk digunakan untuk tahapan selanjutnya

yaitu proses pembobotan kata menggunakan metode *TF-IDF* dengan persamaan 1, yaitu pada tabel 3.

Tabel 3. Perhitungan Nilai IDF

Term	TF						D f	D/d f	IDF
	K	D1	D2	D3	D4	D5			
lambat	1	1	0	1	1	1	4	1,25	0,097
Ajar	1	1	1	1	0	1	4	1,25	0,097
bahasa	0	1	0	0	0	0	1	5	0,699
mudah	0	1	1	0	0	0	2	2,5	0,398
beri	0	1	0	0	0	0	1	5	0,699
materi	0	1	0	0	0	0	1	5	0,699
saran	0	0	0	0	1	1	1	5	0,699
dosen	0	0	0	1	1	1	2	2,5	0,398
masuk	0	0	0	0	1	1	1	5	0,699
kelas	0	0	0	0	1	1	2	2,5	0,398

Tabel 3 merupakan hasil perhitungan IDF dari sepuluh term yang ada. Kolom Df pada term “lambat” diperoleh dari jumlah dokumen yang mengandung term “lambat” yaitu sebanyak 4 (empat) dokumen. Kolom D/df diperoleh dari nilai Df dibagi dengan jumlah dokumen. Kolom IDF diperoleh dari hasil perhitungan *LOG* dari nilai D/df. Setelah dilakukan perhitungan nilai *IDF*, maka kemudian dapat dilakukan perhitungan nilai *weight* (W) pada tabel 4 dengan menggunakan persamaan 2.

Tabel 4. Perhitungan Nilai Weight (W)

Term	W					
	KK	D1	D2	D3	D4	D5
lambat	0,097	0,097	0	0,097	0,097	0,097
ajar	0,097	0,097	0,097	0,097	0	0,097
bahasa	0	0,699	0	0	0	0
mudah	0	0,398	0,398	0	0	0
beri	0	0,699	0	0	0	0
materi	0	0	0	0	0	0,699
saran	0	0	0	0	0,699	0
dosen	0	0	0	0,398	0,398	0
masuk	0	0	0	0	0,699	0
kelas	0	0	0	0	0,398	0,389

Tabel 4 merupakan hasil perhitungan *weight* (W) yang diperoleh dari jumlah term pada setiap dokumen dikalikan nilai IDF. Untuk term “lambat” pada dokumen D1 terdapat 1 (satu) term, diperoleh nilai $W = 1 \times 0,097$. Berikut ini merupakan perhitungan dokumen berdasarkan kata kunci yang dicari yang memiliki nilai *similaritas* dokumen dari paling tinggi ke yang terendah dengan metode *generalized vector space model*.

1. Melakukan *preprocessing* berupa *case folding*, *tokenizing*, *filtering*, dan *stemming*.
2. Menentukan *minterm* yang muncul pada dokumen yang tersedia berdasarkan banyak kata yang diinputkan oleh pengguna, penulis menggunakan dua kata kunci sebagai berikut:
 M_x : lambat
 M_y : ajar

3. Hasil perhitungan bobot (W) pada tabel 4 akan digunakan nilainya ke dalam *vektor orthogonal* yang sudah dibentuk berdasarkan pola *minterm*. Hasil yang didapatkan terdapat pada tabel 5.

Tabel 5. Penentuan Vektor Ortogonal

Dokumen	Lambat	Ajar	Vektor Orthogonal
D1	0,097	0,097	\rightarrow m_1
D2	0	0,097	\rightarrow m_2
D3	0,097	0,097	\rightarrow m_3
D4	0,097	0	\rightarrow m_4
D5	0,097	0,0967	\rightarrow m_5
Q	0,097	0,097	

4. Menghitung *indeks term* dengan menggunakan persamaan 3, hasil perhitungan terdapat pada tabel 6.

Tabel 6. Hasil Perhitungan Indeks Term

Dokumen	Indeks Term	
	K1	K2
D1	0,5	0,5
D2	0	0,5
D3	0,5	0,5
D4	0,5	0
D5	0,5	0,5

Tabel 6 merupakan hasil perhitungan indeks term pada setiap dokumen untuk masing-masing kata kunci yang digunakan. Nilai *indeks term* dokumen D1 terhadap kata kunci “lambat” sebesar 0.5 dan terhadap kata kunci “ajar” sebesar 0.5.

5. Menghitung *korelasi* setiap *terms* dengan menggunakan persamaan 4. Hasil *korelasi term* dapat dilihat pada tabel 7.

Tabel 7. Hasil Perhitungan Korelasi Term

Dokumen	Korelasi Terms (C)	
	K1	K2
D1	0,097	0,097
D2	0	0,097
D3	0,097	0,097
D4	0,097	0
D5	0,097	0,097

Tabel 7 merupakan hasil perhitungan korelasi term pada setiap dokumen untuk masing-masing kata kunci yang digunakan. Nilai *korelasi term* dokumen D1 terhadap kata kunci “lambat” sebesar 0.097 dan terhadap kata kunci “ajar” sebesar 0.097.

6. Melakukan pengubahan kata kunci dan dokumen dalam bentuk *vektor* dengan menggunakan persamaan 5 dan 6, hasil pengubahan dapat dilihat pada tabel 8.

Perhitungan *vektor* dokumen dan kata kunci menggunakan nilai korelasi dan *indeks term* yang telah diperoleh pada tabel 6 dan 7. *Vektor* dokumen D1 terhadap D1, diperoleh dari nilai korelasi D1 dan nilai *indeks term* D1. *Vektor* dokumen D1 terhadap

D2, diperoleh dari nilai *korelasi* D2 dan nilai *indeks term* D1.

Tabel 8. Hasil Perhitungan *Vektor Dokumen*

Dokumen	Vektor Dokumen					Q
	D1	D2	D3	D4	D5	
D1	0,09 7	0,04 8	0,097	0,048	0,097	0,097
D2	0,04 8	0	0,048	0,048	0,048	0,048
D3	0,09 7	0,04 8	0,097	0,048	0,097	0,097
D4	0,04 8	0,04 8	0,048	0	0,048	0,048
D5	0,09 7	0,04 8	0,097	0,048	0,097	0,097

7. Melakukan perhitungan nilai *similaritas* atau kemiripan dokumen dengan kata kunci dengan persamaan 7, kemudian melakukan perangkingan data sesuai data terbesar sampai dengan terkecil. Hasil perhitungan nilai *similaritas* dokumen dapat dilihat pada tabel 9.

Tabel 9. Hasil Perhitungan *Similaritas Dokumen*

Similaritas Dokumen					
D1	D2	D3	D4	D5	
1	0,9354	1	0,9354	1	

Dokumen yang tampil dari hasil pencarian *query* yang dimasukkan oleh pengguna adalah sebagai berikut, pada pencarian yang tampil adalah dokumen 1,2,3,4, dan 5, karena hanya dokumen tersebut yang mempunyai kata yang *relevan* dengan *query* pengguna.

Berikut ini adalah urutan dokumen yang sesuai dengan *query* pengguna berdasarkan nilai hasil *similarity* tertinggi terdapat pada tabel 10.

Tabel 10. Hasil Perangkingan Dokumen

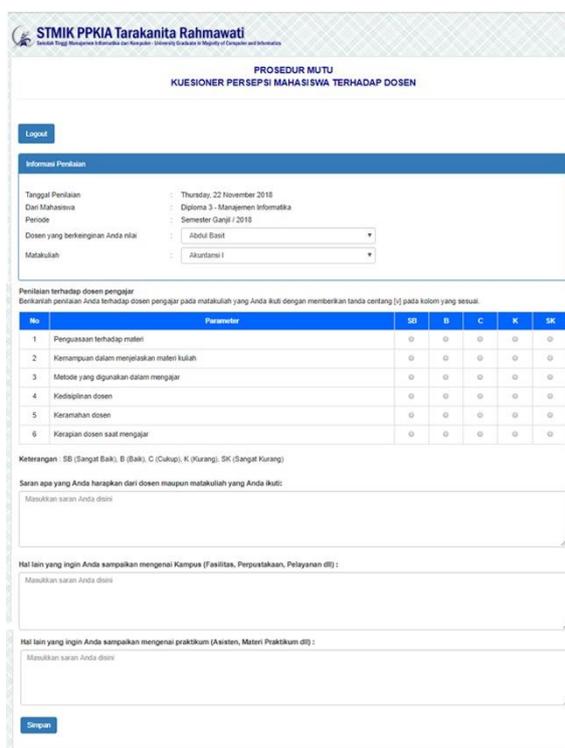
No Urut.	No. Dokumen	Nilai
1	D1	1
2	D3	1
3	D5	1
4	D2	0,9354
5	D4	0,9354

3.2 Aplikasi Pencarian Angket Mahasiswa

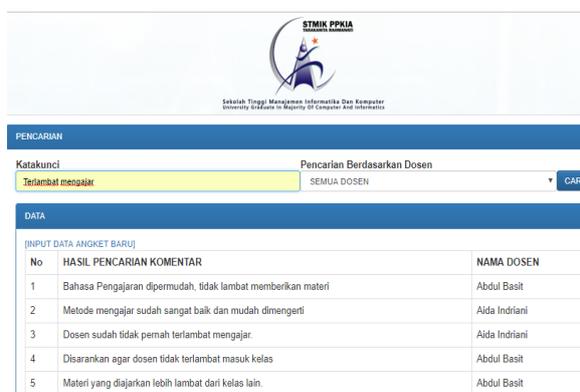
Setelah melakukan perhitungan secara manual, maka dibuatlah aplikasi pencarian angket mahasiswa berbasis *web* dengan menggunakan pemrograman *php*. Data angket yang telah diisi oleh Mahasiswa kemudian dimasukkan kedalam basis data, tampilan aplikasi yang digunakan untuk memasukkan data angket ditunjukkan pada gambar 2.

Gambar 2 merupakan form yang digunakan untuk melakukan input data angket yang telah diisi oleh Mahasiswa. Setelah data angket dimasukkan, langkah selanjutnya memilih tombol simpan yang

nantinya akan tersimpan dalam basis data untuk kemudian dapat digunakan sebagai data set dalam pencarian informasi data angket sesuai dengan kata kunci. Tampilan awal dari aplikasi pencarian ditunjukkan pada gambar 3. Pada menu tampilan awal program ini dapat dilihat bahwa terdapat menu untuk melakukan pencarian berdasarkan kata kunci yang diinputkan, selain itu program ini bisa mencari berdasarkan nama dosen atau semua dosen Terdapat no urut dokumen, hasil pencarian komentar dan juga nama dosen yang dikomentari oleh mahasiswa. Setelah kata kunci diinputkan, kemudian pilih tombol cari untuk memproses pencarian. Hasil pencarian dapat dilihat pada gambar 4.



Gambar 2. Tampilan Input Data Angket



Gambar 3. Tampilan Awal

Pada gambar 4, dapat dilihat hasil dari pencarian yang dilakukan. Dokumen yang memiliki tingkat kemiripan tertinggi dengan kata kunci akan otomatis berada diposisi paling atas. Berikut hasil perhitungannya dapat dilihat pada menu hasil perhitungan yang akan disajikan pada gambar 5.

No	HASIL Pencarian Komentar	NAMA DOSEN
1	Bahasa Pengajaran dipermudah, tidak lambat memberikan materi	Abdul Basit
2	Dosen sudah tidak pernah terlambat mengajar.	Aida Indriani
3	Materi yang diajarkan lebih lambat dari kelas lain.	Abdul Basit
4	Disarankan agar dosen tidak terlambat masuk kelas	Abdul Basit
5	Metode mengajar sudah sangat baik dan mudah dimengerti	Aida Indriani

Gambar 4. Hasil Pencarian

DOKUMEN	NILAI
D001	1
D002	0.935414
D003	1
D004	0.935414
D005	1

Gambar 5. Hasil Perhitungan Program

Dari hasil perhitungan program pencarian angket mahasiswa tersebut hasilnya sama dengan perhitungan manual yang dilakukan.

3.3 Recall dan Precision

Uji coba dilakukan terhadap 50 data angket yang tersebar dari pengajaran, pelayanan dan fasilitas.

Untuk mendapatkan nilai *recall* dan *precision*, digunakan 5 (lima) kata kunci yang diberikan oleh pengguna. Adapun kata kunci yang diberikan untuk melakukan uji coba *recall* dan *precision* adalah “terlambat mengajar”, “ruang kelas panas”, “terlalu cepat menjelaskan”, “sering tambahan” dan “laptop lambat”. Nilai TP, FP, TN dan FN dari masing-masing kata kunci terhadap seluruh dokumen ditunjukkan pada tabel 11.

Tabel 11. Nilai Contingency Table

Kata Kunci	TP	FP	TN	FN
Terlambat mengajar	4	5	41	0
Ruang kelas panas	2	0	46	0
Terlalu cepat menjelaskan	4	0	46	0
Sering tambahan	1	2	47	0
Laptop lambat	4	1	45	0

Dari tabel 10, diperoleh nilai TP yaitu dokumen yang *relevan* dengan kata kunci sebanyak 4 dokumen, sedangkan nilai FP yaitu dokumen yang dihasilkan tetapi tidak *relevan* dengan kata kunci sebanyak 5 dokumen. Dari nilai *contingency table* pada tabel 11 dilakukan perhitungan *recall* dan *precision* dengan menggunakan persamaan 10 dan 11. Sehingga diperoleh nilai seperti ditunjukkan pada tabel 12.

Tabel 12. Nilai Recall dan Precision

Kata Kunci	Precision	Recall	Waktu
Terlambat mengajar	44%	100%	40 detik
Ruang kelas panas	100%	100%	30 detik
Terlalu cepat menjelaskan	100%	100%	36 detik
Sering tambahan	33%	100%	30 detik
Laptop lambat	80%	100%	36 detik
Total	72%	100%	34.4 detik

Dari hasil perhitungan *recall* dan *precision* seperti yang ditunjukkan pada tabel 12, diperoleh nilai rata-rata *precision* untuk 5 uji coba sebesar 72% dan rata-rata *recall* sebesar 100%. Selain itu, rata-rata lama pencarian sebesar 34.4 detik.

4. KESIMPULAN

Dari hasil penelitian diatas dapat disimpulkan bahwa:

1. Pencarian dokumen diurutkan berdasarkan nilai *similaritas* hasil perhitungan dengan nilai > 0 dan diurutkan dari nilai terbesar ke nilai tertinggi. Pencarian akan dilakukan sampai tahap akhir yaitu mencari nilai *similaritas* dokumen, jika terdapat kata pada kata kunci didalam dokumen. Sebaliknya jika tidak terdapat kata tersebut maka, proses pencarian terhenti sampai proses *preprocessing* saja.
2. Berdasarkan hasil perhitungan pada tabel 8, dapat disimpulkan bahwa D1, D3 dan D5 memiliki nilai *similaritas* yang sama dan tertinggi dengan kata kunci, sedangkan D2 dan D4 memiliki nilai

similaritas terendah dari kata kunci. Hal itu dikarenakan D1, D3 dan D5 memiliki lebih banyak kata yang *relevan* atau mirip dengan kata kunci dibandingkan D2 dan D4.

3. Berdasarkan uji coba yang dilakukan terhadap 5 kata kunci dengan 50 data set, diperoleh rata-rata nilai *precision* sebesar 72% dan *recall* sebesar 100%.
4. Berdasarkan percobaan yang dilakukan terhadap 50 data, proses pencarian akan membutuhkan waktu selama 34.4 detik dengan menggunakan 5 kata kunci. Semakin banyak data yang digunakan sebagai data set akan mempengaruhi waktu pencarian dikarenakan semakin banyak data yang diproses.

ROBINSON, L., 2015. Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI. Jurnal UNIKOM. Bandung.

WAITELONIS, J., EXELER, and C., SACK., 2015. Linked Data Enabled Generalized Vector Space Model To Improve Document Retrieval. Proceedings of NLP & DBpedia. Germany.

WONG, S.K.M., ZIARKO, W., and PATRICK, C.N.W., 1985. Generalized Vector Space Model In Information Retrieval, <http://www.cs.odu.edu/~jbollen/IR04/readings/p18-wong.pdf>, akses tanggal 27 Desember 2013.

5. DAFTAR PUSTAKA

- ABDUL, M. A., 2015. Penerapan Algoritma TF-IDF Untuk Pencarian Karya Ilmiah. Skripsi. Semarang.
- AMIN, F., 2015. Sistem Temu Kembali Informasi dengan Metode Vector Space Model. Jurnal Sistem Informasi Bisnis. Vol 2, No. 1. Semarang.
- BAEZA, Y.R., RIBEIRO, N.B., 1999. Modern Information Retrieval.
- DWIYANTORO, 2017. Sistem Temu Kembali Informasi Dengan Keyword (Recall dan Precision pada Judul dan Subjek di OPAC Perpustakaan Universitas Gadjah Mada). Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan. Vol. 5, No. 2, Page164–174. Yogyakarta.
- HENDRA, B., CHATAHALEA, P.N., 2015. Aplikasi Information Retrieval (IR) CATA Dengan Metode Generalized Vector Space Model. Jurnal Informatika. Vol 4, No. 1. Bandung
- LEMAN, D., ANDESA, K., 2015. Implementasi Vector Space Model Untuk Meningkatkan Kualitas Pada Sistem Pencarian Buku Perpustakaan. Seminar Nasional Informatika. Vol 1, No. 1. Medan.
- PARDEDE, J., 2014. Implementasi Multithreading Untuk Meningkatkan Kinerja Information Retrieval Dengan Metode GVSM. Jurnal Sistem Komputer. Vol. 4, No. 1. Semarang.
- PRABOWO, D.A., FHADLI, M., NAJIB, M.A., FAUZI H.A., CHOLISSODIN I., 2016. TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization. Jurnal Teknologi Informasi dan Ilmu Komputer. Vol 3, No. 3. Page 208-215. Malang.