RANCANG BANGUN *SEARCH ENGINE* TERJEMAHAN TAFSIR AYAT-AYAT AL-QURAN PADA DOKUMEN TEKS BERBAHASA INDONESIA MENGGUNAKAN METODE *DICE SIMILARITY*

Evana Ainaul Novita[1], Fatchurrochman, M. Kom[2], Zainal Abidin, M.Kom[3]

Jurusan Teknik Informatika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang
Jl. Gajayana 50 Malang 64144, Indonesia

E-mail: eanevangelist @gmail.com[1], fatchur70@gmail.com[2], zainal@ti.uin-malang.ac.id[3]

ABSTRAK

Al-Quran merupakan sumber hukum utama ajaran Islam dan pedoman hidup bagi manusia. Ayat-ayat Al-Quran menjelaskan berbagai hal secara terperinci, namun terdapat pula ayat yang menjelaskan sesuatu secara global. Ayat Al-Quran yang mengandung pengertian global memerlukan penjelasan dari berbagai kajian sehingga diperlukannya tafsir untuk menafsirkan ayat-ayat tersebut. Tafsir-tafsir al-Quran yang berkembang saat ini seperti tafsir Al-Misbah, tafsir ibnu katsir, tafsir Jalalain dan lain sebagainya. Tafsir-tafsir tersebut menjelaskan kandungan al-Quran yang tersirat. Hal inilah yang memotivasi untuk merancang dan membuat suatu aplikasi yang dapat memudahkan seorang dalam menemukan padanan tafsir terjemahan ayat al-Quran yang sesuai dengan teks/dokumen tertentu. Pada penelitian ini digunakan terjemahan tafsir Jalalain berbahasa Indonesia karena tafsir Jalalain penafsirannya komunikatif dan tidak berbeda jauh dengan terjemahan ayat di dalam al-Quran. Pada pencarian padanan tafsir ayat al-Quran tersebut diukur kemiripan dari dokumen dengan tafsir. Dokumen berupa teks berbahasa Indonesia. Dokumen tersebut sebagai inputan yang akan diproses dengan terjemahan tafsir al-Quran berbahasa Indonesia. Ukuran kemiripan berdasarkan frekuensi kata yang sama yang ada di dalam dokumen inputan dan tafsir al-Quran. Pada penelitian ini peniliti menggunakan metode *Dice Similarity* dalam pemrosesan pengukuran kemiripan dokumen.

Kata kunci: *Text Mining*, *Search Engine*, *Dice Similarity*, Tafsir Jalalain, *TF-IDF*

1. PENDAHULUAN

Al-Quran perlu untuk diperha-tikan dan dipelajari, memperhatikan ayat--ayat al-Quran dapat mengantar-kan manusia kepada keyakinan dan kebenaran Illahi, dan juga untuk me-nemukan alternatif-altenatif baru me-lalui pengintegrasian ayat-ayat terse-but dengan perkembangan situasi ma-syarakat tanpa mengorbankan prin—sip-prisip pokok ajarannya atau meng-abaikan perincian-perincian yang ti-dak termasuk dalam wewenang ijtihad (Quraish Shihab, 2007).

Ayat-ayat Al-Quran menjelaskan berbagai hal secara terperinci, namun ter-dapat pula ayat yang menjelaskan sesuatu se-cara global. Ayat Al-Quran yang me-ngan-dung pengertian global memerlukan penjelasan dari berbagai kajian sehingga diperlukannya tafsir untuk menafsirkan ayat-ayat tersebut. Perujukan suatu per-masalahan dibutuhkan merujuk terhadap ayat-ayat al-Quran dengan penjelasan yang tepat. Satu ayat al-Quran memung-kinkan banyak mengandung penafsiran.

Seiring dengan perkembangan ja-man, berkembang pula bermacam-macam tafsir al-Quran seperti, tafsir al-Misbah, tafsir Ibnu Katsir, tafsir Jalalain dan lain sebagainya. Perkembangan teknologi mem-pengaruhi pula perkembangan tafsir-tafsir tersebut dalam bentuk digital seperti *ebook* dan beberapa tafsir dibuat seperti *search engine* seperti aplikasi al-Quran digital, contoh Tafsir Jalalain Digital (Pe-sa-ntren Per-sa-tuan Islam 91 Kota Tasik-ma-laya).

Namun dalam tafsir digital tersebut masih berupa inputan kata atau beberapa kata. Aplikasi tersebut masih belum men-dukung jika inputan berupa teks doku-men.

Faktor tersebut membuat peneliti termotivasi untuk merancang dan mem-buat sebuah sestem yang dapat mem-permudah seseorang dalam me-nemukan padanan tafsir ayat al-Quran yang sesuai dengan suatu dokumen teks tertentu. Pencarian padanan tafsir yang sesuai dengan dokumen teks bahasa Indonesia yang akan diinputkan, peneliti meng-gunakan metode *dice similarity* dalam proses menemukan padanan yang sesuai.

1. Metode Similaritas *Dice Similarity*

Dice's Cooficient atau yang biasa dikenal sebagai *Dice similarity* adalah ukuran kemiripan yang ditemukan oleh Lee Raymond Dice (dice, 1945).

Dalam mengukur kemiripan sebuah dokumen dengan dokumen lain maka digunakan rumus *dice*

similarity pada persamaan 1:

$$Similarity\ (d_j, q) = \frac{2 \times \sum_{i=1}^{t}(W_{ij} \cdot W_{iq})}{\sum_{i=1}^{t} W_{ij}^2 + \sum_{i=1}^{t} W_{iq}^2} \quad (1)$$

Dari perhitungan tersebut akan diperoleh nilai 0d" xd"1, dimana jika nilai tersebut semakin mendekati 1 maka tingkat kemiripan dari apa yang dilambangkan dengan vektor dokumen A dan dokumen B makin besar (Husni, IR-dan klasifikas).

2. Pembobotan Menggunakan *TF-IDF*

Metode *tf*-idf (Robertson, 2005) merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan se-buah kata di dalam sebuah dokumen ter-tentu dengan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan secara umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi ke-seluruhan dokumen yang mengandung kata tersebut yang terdapat pada kum-pulan dokumen (*database*).

Rumus untuk *tf*-idf pada persamaan 2:

$$W_{ij} = tf\ x\ idf$$

$$W_{ij} = tf_{ij} x\ log\frac{N}{n} \quad (2)$$

Keterangan:

W_{ij} = bobot kata yang terdapat pada dokumen

tf_{ij} = jumlah kemunculan kata/term t_j di dalam dokumen

N = Jumlah semua dokumen yang ada dalam database

n= Jumlah dokumen yang mengandung kata t_j

3. Representasi *Vector Space Model (VSM)*

Pada *VSM*, setiap dokumen di dalam database dan query pengguna dire-presentasikan oleh suatu vektor multi-di-mensi (Polettini, Nicola : 2004; Cios, Krzysztof: 2007). Dimensi sesuai dengan jum-lah term dalam dokumen yang terlibat.

Banyaknya dimensi dari ruang vek-tor akan ditentukan oleh jumlah kata sig-nifikan yang tedapat dalam dokumen ataupun *query*. Diasumsikan terdapat *t* buah term yang berbeda, yang disebut *vo-cabulary* atau *term index*, pada koleksi dokumen. Kumpulan term tersebut akan dibentuk menjadi sebuah ruang vector dengan dimensi = *t* = *vocabulary*.

Misalkan terdapat sejumlah *n* kata yang berbeda sebagai kamus kata (*vo-ca-bu-lary*) atau indeks kata (*terms index*). Kata-kata ini akan membentuk ruang vektor yang memiliki dimensi sebesar *n*. Setiap kata *i* dalam dokumen atau *query* diberikan bobot sebesar *wi*. Baik do-ku-men maupun *query* direpre-sen-tasikan se—ba-gai vektor berdi-mensi *n*. Sebagai contoh terdapat 3 buah kata (T_1, T_2 dan T_3), 2 buah dokumen (D1 dan D2) serta sebuah *query* Q. Masing-masing nilai pada persamaan 3.

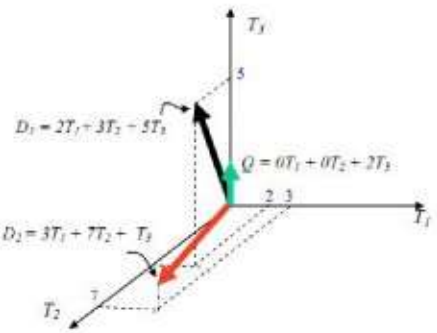
Koleksi dan *n* hasil dokumen dapat direpresentasikan dalam *Vector Space Model* dengan sebuah matriks term-dokumen sebagai berikut:

$$D_1 = 2T_1+3T_2+5T_3$$

$$D_2 = 3T_1+7T_2+0T_3$$

$$Q = 0T_1+0T_2+2T_3 \quad (3)$$

Tiap dokumen akan direpresen-tasikan sebagai sebuah vector akan berva-riasi sesuai dengan bobot dari masing-ma-sing term yang menjadi elemen vector. Se—tiap term *I*, dalam do-ku-men *j*, diberi bobot *Wij*. Repre-sentasi dokumen dan term dalam bentuk vector bedimensi *t* digambarkan seperti pada **Gambar 2** dan dengan representasi pada persamaan 4.



Gambar 1: **Contoh model ruang vektor dengan dua dokumen D_1 dan D_2**

$$d_j = (W_{1j}, W_{2j}, \ldots, W_{tj}) \quad (4)$$

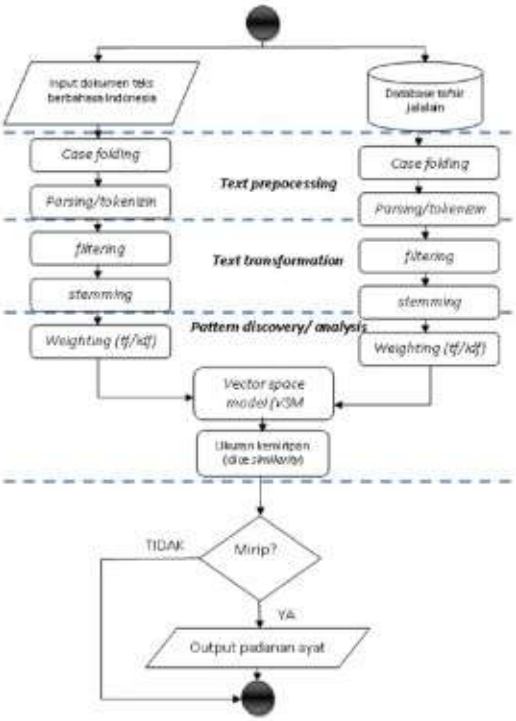
$$\begin{array}{c|cccc} & T_1 & T_2 & \cdots & T_t \\ D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{tn} \end{array}$$

Gambar 2: **Representasi Vektor dokumen dan query**

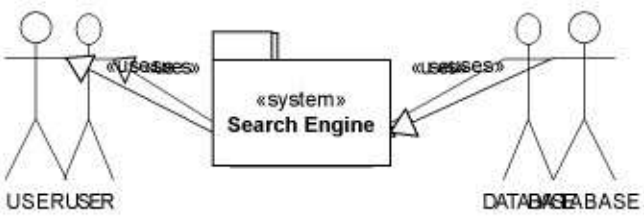
Sebuah entri dalam matriks term-dokumen berkorespondensi dengan bobot sebuah term dalam dokumen tertentu. Nilai 0 berarti keberadaan tem tersebut tidak signifikan dalam dokumen atau term tesebut tidak terdapat dalam dokumen.

Pembobotan term untuk VSM dapat dilakukan dengan konsep *Term Frequency* (*TF*) dan *Inverse Document frequency (IDF).*

4. Rancangan Sistem



Gambar 3: **Tahap-tahap penelitian**



Gambar 4: **Diagram *Usecase* Aplikasi *Search Engine* Tafsir Jalalain**

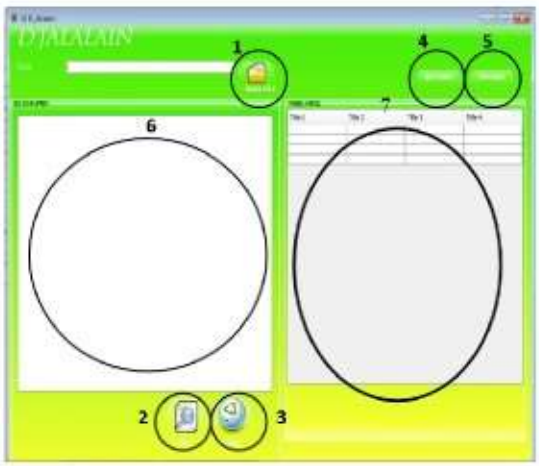
Tahap-tahap yang dilakukan dalam penelitian dapat dilihat pada **Gambar 3**. Pada penelitian ini, untuk proses *stem-ming* menggunakan *stemming Nazif Andriani*, dan pembobotan kata meng-gunakan pembobotan tf-idf. Sedangkan untuk ukuran kemiripan menggunakan metode *dice similarity.*

Dalam pembuatan *Search Engine* Tafsir Jalalain ini, melibatkan dua aktor yaitu *user* dan database, *user* bisa me-masukkan dukumen teks sebagai in-putan sistem. Database digunakan untuk me-nyimpan data tafsir jalalain sebagai acuan untuk menemukan padanan ayat. Ju-ga data pemrosesan kata dan perhi-tungan frekuensi kata.

5. Implementasi

Pada aplikasi *search engine* ter-dapat menu utama yang akan muncul ke-ti-ka program pertama kali dijalankan. *Me*—-nu utama dalam aplikasi *search en-gine* tafsir ayat-ayat al-Quran ditunjukkan oleh **Gambar 5**.

Pada menu utama terdapat tombol buka file pada komponen 1 untuk men-cari dokumen dan menampilkan isi dari do-kumen tersebut pada teks area pada komponen 6 untuk menampilkan isi do-ku-men. Pada bagian bawah teks area dokumen terdapat tombol cari pada komponen 2 dan refresh pada komponen 3. Tombol cari digunakan untuk mem-pro-ses isi dokumen yang hasilnya akan di-tampilkan pada table pada komponen 7. Tombol refresh pada komponen 3 di-gu-nakan untuk mengulang jika user ingin memproses dokumen lainnya.



Gambar 5 :**Tampilan Menu Utama *Search Engine* Tafsir al-Quran**

6. Hasil Uji Coba

Dokumen yang digunakan dalam uji coba adalah dokumen teks Bahasa Indo-nesia yang diambil dari kumpulan kultum setahun jilid 1 dan 2 karya

Fuad Abdul Azis Asy-Syalhub tahun 2008. Pada kumpulan khutbah yang digunakan akan diambil sampel sebanyak 30 doku-men. Pengujian ini dilakukan dengan cara menginputkan 30 dokumen ke aplikasi *search engine* dan melihat hasil dari pa-danan tafsir yang cocok. Kemudian hasil pa-da-nan tafsir tersebut dicocokkan dengan ayat yang berada pada dokumen.

Tabel 1: **Tabel hasil uji coba *recall*, *precision*, *dan accuracy* 30 dokumen**

No	Dokumen	*Recall*	*Precision*	*Accuracy*
1	Dokumen1	66.67	13.33	99.78
2	Dokumen2	80	26.67	99.8
3	Dokumen3	66.67	13.3	99.78
4	Dokumen4	100	13.33	99,79
5	Dokumen5	66.67	13.33	99,78
6	Dokumen6	100	13.33	99,79
7	Dokumen7	100	13.33	99,79
8	Dokumen8	50	6,67	99,76
9	Dokumen9	100	13.33	99,79
10	Dokumen10	100	6,67	99,78
11	Dokumen11	100	6,67	99,78
12	Dokumen12	66.67	13.33	99,78
13	Dokumen13	100	20	99,81
14	Dokumen14	0	0	99,69
15	Dokumen15	50	6.67	99,76
16	Dokumen16	0	0	99,69
17	Dokumen17	66.67	13,33	99,78
18	Dokumen18	100	13,33	99,79
19	Dokumen19	50	13.33	99,76
20	Dokumen20	0	0	99,74
21	Dokumen21	66.67	13.33	99,78
22	Dokumen22	0	0	99,74
23	Dokumen23	100	6,67	99,78
24	Dokumen24	20	6.67	99,71
25	Dokumen25	100	20	99,81
26	Dokumen26	12.5	6,67	99,66
27	Dokumen27	50	33.33	99,76
28	Dokumen28	33,33	13,33	99,73
29	Dokumen29	0	0	99,74
30	Dokumen30	0	0	99,73
	Rata-rata	**58.19**	**10.63**	**99.76**

Pengukuran dilakukan dengan melakukan perhitungan *presicion*, *recall*, dan *accuracy* pada hasil jawaban pada masing-masing pertanyaan. *Precision* ada-lah bagian yang dari dokumen di-temukan yang relevan dan *recall* adalah bagian dari dokumen relevan yang dite-mukan. *Accuracy* merupakan tingkat ke-aku-ratan sistem dalam menemukan jawaban.

Perhitungan 30 dokumen pada **tabel 1** dihitung rata-rata nilai *recall*, *pre-ci-sion*, dan *accuracy*. Rata-rata nilai *recall*, *precision* dan *accuracy* yang telah dilakukan untuk dokumen yang diperoleh adalah sebesar *recall* 58, 19%, precision 10, 63%, dan *accuracy* 99, 76%. Nilai *recall* ini sesuai dengan persamaan 4.1 adalah dokumen atau ayat yang *relevan* dan *retrieve* dibagi dengan dokumen yang *relevan* dan *retrieve* ditambah dengan dokumen yang *relevan* akan tetapi *nonretrieve*. Nilai yang dihasilkan rata-rata masih tinggi karena tingginya dokumen yang ditemukan relevan. Dari 30 dokumen yang telah diujicoba, ada 80% atau 24 dokumen yang ditemukan padanan ayatnya dan sebanyak 20% atau 6 dokumen tidak ditemukan padanan ayatnya.

7. KESIMPULAN

Dari penelitian dan uji coba aplikasi ini, dapat disimpulkan bahwa dari 30 dokumen yang telah diuji coba, ada 80% atau 24 dokumen yang ditemukan padanan ayatnya dan sebanyak 20% atau 6 dokumen tidak ditemukan padanan ayatnya.

Pengujian berdasarkan kesamaan kata pada dokumen dan tafsir sangat mempengaruhi tafsir ayat yag ditemukan. Pengujian berdasarkan pembobotan kata sehingga kesamaan kata yang muncul pada dokumen dan tafsir sangat mempengaruhi nilai similaritas dan tafsir ayat yang ditemukan. Referensi ayat tidak hanya diperhitungkan dengan kesamaan kata dengan dokumen tetapi juga diperhitungkan kesamaan makna-nya. Sehingga tafsir ayat berdsarakan referensi 20 % tidak ditemukan.

8. SARAN

Ada beberapa hal yang perlu dikembangkan dari penelitian ini, diantaranya:

a. Dibutuhkan optimasi waktu sehingga aplikasi dapat berjalan lebih cepat dalam melakukan pencarian meskipun input berupa dokumen yang panjang.

b. Pembatasan jumlah tafsir ayat yang ditemukan sistem sangat mempengaruhi tafsir ayat yang ditemukan sehingga diperlukan uji coba dengan pengujian tafsir ayat dengan jumlah batasan tafsir ayat yang ditemukan oleh sistem lebih besar.

c. Perlu dilakukan uji coba dan analisis dengan menggunakan metode lain sehingga dapat disimpulkan metode mana yang lebih baik dalam menemu-kan tafsir ayat dan keaku-ratannya.

d. Pada penelitian ini, pencarian ter-hadap teks tidak memperhatikan kedudukan kata dalam kalimat atau teks tersebut. Untuk penelitian se-lan-jutnya dapat dikembangkan dengan memperhatikan susunan dan makna kata kalimat dalam pencarian.

9. REFERENSI

Aisan J, Williams H.E dan Tahaghogi, S.M.M. 2005. *Stemming Indonesia*, Melbourne: RMIT University

Asy-Syalhub, Fuad Abdu lAzis. 2008. *Kumpulan Kultum Setahun Jilid I dan II*. Jakarta: DarulFalah

Dice, Lee R. 1945. *Measures of the amount of Ecologic association between species.*

Even.Yahir dan Zohar. 2002. *Introduction to Text Mining. Auto-mated Learning Group National Center For Supercomputing Aplica-tions*. University of Illionis.

Garcia. 2006. Cosine Similarity and Term Weigh Tutorial. http://www.miisli-ta.com/information-retrieval-tuto-rial-/cosine-similarity-tuto-rial.html. Di-akses pada 25 Juli 2011

http://www.qurandatabase.org/. Diakses pada tanggal 10 September 2011

Polettini, Nicola. 2004. *The Vector Space Model in Information Retrieval – Term Weighting Problem*

Santosa, R.Gunawan. 2004. *Statistik*. Yogyakarta : Penerbit Andi

Shihab, Quraish. 2007. *Membumikan Al-Quran*. Bandung: Penerbit Mizan

Tala., Fadillah Z. 2003. *A Study of Stemming Efects on Information Retrieval in Bahasa Indonesia*. Institute for Logic, Language and Computation Universite itvan Ams-ter-dam The Netherlands. www.-illc.-uva.nl/publications/ResearchReport/Mol200302.text.pdf. Diakses tang-gal 8 Juni 2011.

Tan, Pan Ning etc. 2009. *Introduction to Data Mining*. http:www-users.-cs.-umn.edu/-kumar/dmbook/index.php. Diakses pada 06 April 2012

Trunojoyo, Husni. Buku Ajar. *Information Retrieval dan Klasifikasi*. http://husni.trunojoyo.-ac.id/wp-content/uploads/2010/03/-Husni--IR-dan-Klasifikasi.pdf. Di-ak-ses pada 9 Agustus 2011.