Implementasi Koreksi Jawaban Soal Essai Menggunakan Metode *Vector Space* Model

Ipin Sugiyarto[1], Hanafi Eko Darono[2]

***Abstract*— Correction answers to the essay still use manual way to judge based personal weight determined by each teacher. Correction value calculation result of the answers to the essay can use the quick way by computational methods. Used computational methods based text mining to test the system using the information retriveral vector space model. The result of the test method vector space model obtained by election the second-highest rank of valid answer content to answer valid student and teachers. Answer approach contained in rank-2 closest rank-1 with the result of the closest distance between the has a similarity value for student 0.45 and 0.10 teacher similarity value.**

Intisari—Koreksi jawaban soal essai selama ini masih menggunakan cara manual dengan menilai berdasarkan bobot personal yang ditentukan oleh setiap pengajar. Perhitungan nilai hasil koreksi jawaban soal essai dapat menggunakan cara cepat dengan metode komputasi. Metode komputasi yang digunakan berbasis text mining dengan menguji information retriveral system menggunakan metode *Vector Space Model*. Hasil dari uji coba menggunakan metode *Vector Space Model* didapatkan berdasarkan pemilihan rangking kedua tertinggi dari konten jawaban valid siswa terhadap jawaban valid pengajar. Jawaban mendekati terdapat di rank-2 terdekat dari rank-1 dengan hasil distance terdekat antara keduanya yang memiliki nilai *similarity* untuk siswa 0.45 dan 1.10 nilai similarity pengaja

Kata Kunci — Koreksi, vector space model, rank, *similarity*

I. PENDAHULUAN

Koreksi jawaban soal essai saat ini masih menggunakan cara manual berdasarkan pemberian bobot nilai pada setiap soal yang sudah ditentukan oleh pengajar. Penilaian cara tersebut masih kurang efektif secara waktu karena pengajar harus membaca satu persatu jawaban siswa secara detail dan sering sekali didapatkan jawaban yang melebihi konten teks yang sudah *valid* dari jawaban pengajar. Metode komputasi dapat digunakan untuk memecahkan permasalahan tersebut.

Text mining adalah ilmu multi-displin *text* yang mengambil informasi melibatkan, analisis teks, ekstrasi informasi, *clustering*, *categorization*, visualisasi, teknologi database, *machine learning* dan data *mining* [1]. Kasus yang akan dibahas mengenai penemuan kembali informasi berdasarkan kata kunci atau *query* yang relevan sebagai kesamaan jawaban essai antara siswa dengan pengajar.

Dalam hal ini apakah jawaban soal essai siswa sesuai dengan konten teks jawaban soal essai pengajar dan seberapa besar jarak kedekatan antara keduanya untuk dapat menentukan kecocokan dan pemberian keputusan benar atau tidak benar jawaban essai tersebut. Data yang digunakan menggunakan jawaban soal essai yang sudah ditentukan oleh pengajar dalam bentuk teks berbahasa Indonesia dan data teks jawaban soal essai dari masing-masing murid sebanyak tiga jawaban soal essai yang berbeda.

Berdasarkan identifikasi permasalahan tersebut dapat di garis bawahi bagaimana membuat aplikasi dengan algoritma yang efektif untuk mengkoreksi jawaban soal essai berupa teks sehingga memudahkan pengajar dalam menentukan nilai hasil koreksi jawaban soal essai dengan cepat dan efisiensi waktu. Dokumen yang digunakan dalam penelitian ini berupa dokumen teks berbahasa Indonesia. Penelitian ini menggunakan metode *Vector Space Model* sebagai solusi penyelesaian permasalahan.

Manfaat yang diharapkan dari penelitian ini dapat digunakan sebagai alat bantu untuk penilaian jawaban soal essai secara cepat. Tujuan dari penelitian untuk mengembangkan metode penilaian cepat dengan *Vector Space Model Method* yang menjadi pengendali keputusan berdasarkan tingkat similarity untuk menentukan jawaban soal essai yang benar.

II. LANDASAN TEORI DAN TINJAUAN PUSTAKA

A. *Text Mining*

Text mining merupakan hasil pengembangan dari data *mining* dengan tujuan untuk mencari dan menemukan pola yang menarik dari sekumpulan data teks dengan jumlah besar [2]. Berikut ini langkah-langkah yang dapat dilakukan dengan *menggunakan text mining* sebagai berikut:

1. *Text Processing*
   Tahap awal *text processing* dengan cara *to Lower Case*, merubah semua karakter huruf Kapital menjadi huruf kecil dan dilakukan *tokenizing* yaitu proses pemisahan deskripsi yang awal berupa kalimat-kalimat menjadi kata-kata dan menghilangkan *delimiter-delimiter* seperti tanda titik(.), koma(,), spasi dan karakter angka yang ada dalam kata tersebut [3].
2. *Feature Selection*
   Proses *feature selection* merupakan tahap penghilangan *stopwoard (stopwoard removal)* dan *steamming* terhadap kata yang memiliki imbuhan [4][5]. *Stopwoard* adalah kosakata yang tidak termasuk ciri (kata unik)

[1]*Sekolah Tinggi Manajemen dan Ilmu Komputer, Program Studi Ilmu Komputer e-mail: ipin.sugiyarto@gmail.com*

[2]*Universitas Bina Sarana Informatika, Fakultas Teknologi Informasi, Program Studi Sistem Informasi, hanafi.haf @bsi.ac.id*

dari suatu dokumen [6]. Misalnya “di”, “oleh”, “pada”, “sebuah”, “karena” dan lain sebagainya

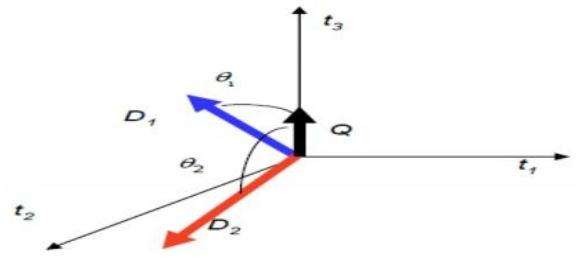
Steamming merupakan proses pemetaan dan pemisahan berbagai bentuk *(variants)* dari suatu kata menjadi bentuk kata dasarnya (stem) [7]. Tujuan dari proses steaming adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun konfiks yang terdapat pada setiap kata.

B. *Infromation Retriveral*

Information retriveral merupakan terapan ilmu komputer yang mempelajari tentang pengambilan sebuah informasi berdasarkan isi dan konteks dari dokumen-dokumen. Proses *information restriveral* dapat dideskripsikan seperti sebuah proses untuk mencari dokumen yang relevan dari sekumpulan dokumen dengan cara pencarian menggunakan kata kunci atau *query* yang ditentukan oleh *user* dalam mencari dokumen yang diinginkan. Salton menjelaskan bahwa sistem temu kembali informasi merupakan penyambung antara user dengan sumber informasi yang tersedia di sekumpulan database pencarian diantaranya seperti, mempresentasikan sekumpulan ide dalam sebuah dokumen menggunakan sekumpulan konsep, terdapat beberapa pengguna yang memerlukan ide tetapi tidak dapat mengidentifikasikan dan menemukannya dengan baik.

C. *Vector Space* Model

Vector Space Model merupakan metode yang digunakan untuk mencari tingkat kedekatan atau *smilarity term* dengan cara memberikan nilai bobot pada *term* yang telah ditentukan.



Gbr 1 Ilustrasi *Vector Space Model*

Keterangan:
Ti = Kata di dalam database
Di = Dokumen
Q = Kata kunci

Cara menggunakan perhitungan persamaan pada *vector space model* adalah dengan menghitung nilai *cosines* sudut dari dua *vector*, yaitu *vector* kata kunci dan *vector* tiap dokumen ke-i.

$$Cosine\,\theta_{D_i} = Sim(Q,D_i) \qquad (1)$$

Keterangan:
Q = *Query* (kata kunci)
Di = Dokumen ke-i

$$Sim(Q,D_i) = \frac{\sum_j w_{i,j}\, w_{q,j}}{\sqrt{\sum_j w_{i,j}^2}\sqrt{\sum_i w_{q,j}^2}} \qquad (2)$$

Keterangan:
Di = Dokumen ke-i
Q = *Query* (Kata kunci)
J = Kata di seluruh dokumen

$$Cosine\,\theta_{D_i} = \frac{Q \cdot D_i}{|Q| * |D_i|} \qquad (3)$$

Keterangan:
Di = Dokumen ke-i
Q = *Query* (Kata kunci)
|Q| = *Vector* Q
|Di| = *Vector* Di

III. PEMBAHASAN

Berikut ini hasil proses teks *minning* pada konten teks jawaban soal essai. Menggunakan teks jawaban soal essai sebanyak 18 kata subjek, predikat dan objek yang terdapat pada masing-masing jawaban soal dari tiap responden yaitu murid dan pengajar serta satu kata kunci atau query berupa kata “norma”.

TABEL I DATA TRAINING

Query	D1 (Guru)	D2 (Murid _Janu)	D3 (Murid_ Raihan)	D4 (Murid_ Haikal)
Norma	Seperangkat aturan atau kaidah yang digunakan untuk mengatur tingkah laku masyarakat yang berisi perintah dan larangan	Adab yang mengatur perilaku seseorang	Norma itu aturan	adab

TABEL II PROSES TOKENISASI

Term	Doc ID
Seperangkat	1
Aturan	1
Kaidah	1
Digunakan	1
Mengatur	1
Tigkah	1
Laku	1
Masyarakat	1
Berisi	1
Perintah	1
Larangan	1
Adab	2

Term	Doc ID
Perilaku	2
Sesorang	2
Norma	3
Aturan	3
Adab	4

TABEL III PROSES INDEXER

Term	Doc ID
Adab	2
Adab	4
Aturan	1
Aturan	3
Berisi	1
Digunakan	1
Kaidah	1
Laku	1
Larangan	1
Masyarakat	1
Mengatur	1
Mengatur	2
Norma	3
Perilaku	2
Perintah	1
Seperangkat	1
Sesorang	2
Tigkah	1

TABEL IV PROSES FILTERING DAN STREAMING

Term	Doc ID Freq.	Plot
Adab	2	2,4
Atur	2	1,3
Isi	1	1
Guna	1	1
Kaidah	1	1
Laku	1	1
Larang	1	1
Laku	1	1
Masyarakat	1	1
Norma	1	3
Laku	1	2
Perintah	1	1
Perangkat	1	1
Sesorang	1	2
Tigkah	1	1

Tahap tokenisasi mengelompokan kata serta indentifikasi alamat dokumen. Proses indexer diurutkan secara abjad, kemudian proses filtering dan steaming yaitu menghilangkan kata berimbuhan awalan dan akhiran serta mengelompokan dalam satu kata yang unik dan pemberian jumlah banyaknya kemunculan atau frekuensi dokumen dan pemetaan plot letak asal dokumen.

Tahap berikutnya adalah menghitung bobot dokumen seperti tabel V :

TABEL V PROSES FILTERING DAN STREAMING

No	Token	Tf					Df	
		Q	D1	D2	D3	D4		D/df
1	norma	1	0	0	1	0	1	4
2	perangkat	0	1	0	0	0	1	4
3	Atur	0	1	1	1	0	3	1
4	Tingkah	0	1	0	0	0	1	4
5	Perilaku	0	1	1	0	0	2	2
6	Masyarakat	0	1	0	0	0	1	4
7	Isi	0	1	0	0	0	1	4
8	Perintah	0	1	0	0	0	1	4
9	Larang	0	1	0	0	0	1	4
10	Adab	0	0	0	0	1	2	2
11	Seorang	0	0	0	0	0	1	4
12	Laku	0	1	0	0	0	1	4
13	Guna	0	1	0	0	0	1	4
14	Kaidah	0	1	0	0	0	1	4

Perhitungan bobot dokumen menggunakan persamaan rumus:

$$\text{D/df} = \frac{\text{Jumlah keseluruhan dokumen}}{\text{Jumlah dokumen yang dicari}} \quad (4)$$

Setelah perhitungan bobot dokumen didapatkan, langkah berikutnya mencari nilai tf-idf weighting yaitu perhitungan bobot hasil perkalian antara dokumen dengan hasil log dari D/df dengan hasil sebagai berikut:

TABEL VI PERHITUNGAN TF-IDF WEIGHTING

IDF	W				
log(D/df)	Q	D1	D2	D3	D4
0.602	0.602	0	0	0.602	0
0.602	0	0.602	0	0	0
0.125	0	0.125	0.125	0.125	0
0.602	0	0.602	0	0	0
0.301	0	0.301	0.301	0	0
0.602	0	0.602	0	0	0
0.602	0	0.602	0	0	0
0.602	0	0.602	0	0	0
0.602	0	0.602	0	0	0
0.301	0	0	0.301	0	0.301
0.602	0	0	0.602	0	0
0.602	0	0.602	0	0	0
0.602	0	0.602	0	0	0
0.602	0	0.602	0	0	0

Tahap berikut setelah tf-idf didapatkan maka menghitung jarak dokumen dan *query* dari hasil akar kuadrat masing-masing *query* (Q) dan data (Di), berikut ini hasil perhitungan:

TABEL VII HITUNGAN JARAK DOKUMEN DAN QUERY

Token	W				
	Q^2	D1^2	D2^2	D3^2	D4^2
Norma	0.362	0	0	0.362	0
Perangkat	0	0.362	0	0	0
Atur	0	0.016	0.016	0.016	0
Tingkah	0	0.362	0	0	0
Perilaku	0	0.091	0.091	0	0
Masyarakat	0	0.362	0	0	0
Isi	0	0.362	0	0	0

Perintah	0	0.362	0	0	0
Larang	0	0.362	0	0	0
Adab	0	0	0	0	0.091
Seorang	0	0	0.362	0	0
Laku	0	0.362	0	0	0
Guna	0	0.362	0	0	0
Kaidah	0	0.362	0	0	0
	SORT Q		SORT Di		
	0.602	1.835	0.748	0.615	0.301

Kemudian langkah berikutnya menghitung nilai dot hasil dari perkalian antara nilai kuadrat dokumen ke-i dengan nilai kuadrat dari query. Seperti hasil berikut ini:

TABEL VIII PERHITUNGAN DOT

Menghitung dot			
Q*D1	Q*D2	Q*D3	Q*D4
0	0	0.131	0
0.131	0	0	0
0.006	0.006	0.006	0
0.131	0	0	0
0.033	0.033	0	0
0.131	0	0	0
0.131	0	0	0
0.131	0	0	0
0.131	0	0	0
0	0.033	0	0.033
0	0.131	0	0
0.131	0	0	0
0.131	0	0	0
0.131	0	0	0
Sum (Q*Di)			
1.220	0.203	0.137	0.033

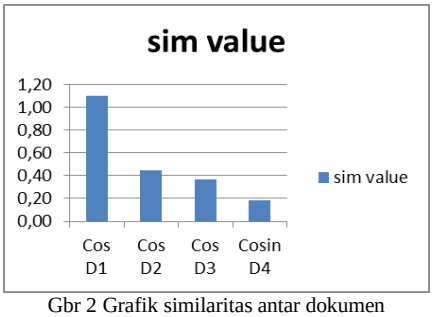
Tahap berikutnya adalah menghitung similaritas antar dokumen dengan rumus persamaan sebagai berikut:

$$Sim = \frac{\sum (Q*Di)}{\sqrt{Q\text{^}2} * \sqrt{D\text{^}2}} \qquad (5)$$

Perhitungan similaritas menghasilkan nilai yang dapat digunakan untuk menentukan ranking antar dokumen sebagai penentu hasil kedekatan dokumen yang memiliki tingkat kemiripan tertinggi.

TABEL IX PERHITUNGAN SIMILARITY

	Consine D1	Consine D2	Consine D3	Consine D4
	1.10	0.45	0.37	0.18
Rank	1	2	3	4



Gbr 2 Grafik similaritas antar dokumen

IV. KESIMPULAN

Hasil implementasi koreksi jawaban soal essai menggunakan *metode vector space model* menghasilkan nilai similaritas antara dokumen D1 dengan nilai 1.10, dokumen D2 dengan nilai 0.45, dokumen D3 dengan nilai 0.37 dan dokumen D4 dengan nilai 0.18. Dari hasil perhitungan similarity tersebut dapat disimpulkan bahwa hasil koreksi jawaban essai yang paling tepat di antara D2, D3 dan D4 dengan dokumen ke-1 (D1) pada query norma adalah dokumen ke-2 (D2) dengan tingkat kemiripan yang tertinggi mendekati nilai *smilarity* dokumen ke-1.

REFERENSI

[1] Feldman, R. & Dagan, L. (1995) Knowledge discovery in textual database (KDT). Inproceedings of the first International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, Agustus 20-21, AAAI Press, 122-117.

[2] Klasifikasi Konten Berita Dengan Metode Text Mining. Jurnal Dunia Teknologi Informasi Vol. 1, No.1, (2012), 14-19.

[3] Lin, S. 2008. A Document Classification and retriveral system for R&D in semiconductor industry-A hybrid approach. Expert system 18, 2:4753-4764.

[4] Berry, M. W. & Kogan, J. 2010. Text Mining application and theory. WILEY: United Kingdom.

[5] Draut, E. Fang, F. Sistla, P. Yu, S & Meng, W. 2009. Stop word and related problems in web interface integration. http://www.vldb.org/pvldb/2/vldb09-384.pdf. Diakses pada tanggal 22 Jan 2018.

[6] Tala, Fadilla, Z. 2003. A Study of Stemming Effect on Information Retriveral in Bahasa Indonesia. Institute for Logic, language and computation Universitate van Amsterdam the netherland. http://www.illc.uva.nl/Research/Reports/MoL-2003-02.text.pdf. Diakses pada tanggal 23 Januari 2018.

[7] Weiss, S. M. Indurkhya, N. Zang, T. Dhamerau, F. J. 2005. Text Mining: Predictif method of Analyzing Unstructured Information. Springer: New York.

[8] Salton, G. 1989. Automatic Text Processing, The Transformation, Analysis and Retriveral of Information by Computer, Addison – Westly Publishing Company, Inc. All right reserved

Ipin Sugiyarto – Lahir di Jakarta, 8 Maret 1988. Lulus S1 Teknik Informatika STMIK Nusa Mandiri, saat ini sedang melanjutkan kuliah S2 Ilmu Komputer di STMIK Nusa Mandiri. Menjadi dosen pengajar di beberapa perguruan tinggi. Salah satunya STMIK Antar Bangsa.

Hanafi Eko Darono lahir di Jakarta, 31 Agustus 1982. Pendidikan Terakhir Strata Dua (S2) STMIK Nusa Mandiri Pasca Sarjana. Bekerja sebagai tenaga pengajar di Universitas Bina Sarana Informatika dan sebagai Head Department Stock Controller di PT. Trans Retail Indonesia (Carrefour).