IMPLEMENTASI METODE *VECTOR SPACE MODEL* (VSM) UNTUK REKOMENDASI NILAI TERHADAP JAWABAN *ESSAY*

Harry Septianto

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : harryseptianto666@gmail.com

ABSTRAK

Setiap proses pembelajaran membutuhkan suatu evaluasi berupa ujian.Ujian esai merupakan evaluasi pembelajaran dalam bentuk soal esai yang mempunyai jawaban lebih bervariasi dibandingkan soal pilihan ganda. Variasi jawaban tersebut memberikan kesulitan terhadap guru dalam menilai jawaban esai. Pada penelitian ini metode yang digunakan untuk pencocokan kata adalah metode *Vector Space Model* (VSM).
Kata kunci : *Vector Space Model*, Ujian Esai,Rekomendasi Nilai.

1. PENDAHULUAN

Setiap proses pembelajaran membutuhkan suatu evaluasi berupa ujian. Ujian dapat dilakukan dalam tiga jenis, yang pertama yaitu ujian pilihan ganda, ujian isian singkat dan ujian esai.Ujian esai merupakan evaluasi pembelajaran dalam bentuk soal esai yang mempunyai jawaban lebih bervariasi dibandingkan soal pilihan ganda. Variasi jawaban tersebut memberikan kesulitan terhadap guru dalam menilai jawaban esai.

Sudah banyak penelitian tentang koreksi esai otomatis, salah satunya adalah penelitian yang dilakukan oleh Sahriar Hamzah, M. Sarosa dan Purnomo Budi Santoso yang menggunakan algoritma Rabin-Karb. Tingkat keakurasian dari algoritma Rabin-Krab adalah 90,31%. Selain menggunakan algoritma Rabin-Karb, algoritma pencocokan string yang lain adalah algoritma *winnowing* dengan tingkat keakurasian untuk algoritma winnowing adalah 75-80%. Pada penelitian ini untuk mencocokkan string menggunakan metode *Vector Space Model* (VSM).

Maka dari itu penelitian ini diharapkan mendapat suatu hasil penelitian berupa hasil nilai akurat dari metode VSM.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan oleh penulis di atas, maka dapat dirumuskan suatu masalah yaitu bagaimana cara mencocokkan kata dan merekomendasikan nilai terhadap jawaban esai yang telah dimasukan siswa di dalam media pembelajaran.

1.2 Maksud Dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas akhir ini adalah mengimplementasikan metode *Vector Space Model* (VSM) untuk pencocokan kata dan rekomendasi nilai terhadap jawaban esai.

Sedangkan tujuan yang akan dicapai dalam penelitian ini adalah sebagai berikut :

1. Untuk melihat keakurasian dari metode VSM dalam mencocokkan kata.
2. Untuk melihat seberapa akurat sistem dalam memberikan rekomendasi nilai terhadap jawaban siswa yang telah dengan kunci jawaban.

1.3 Batasan Masalah

Terdapat beberapa batasan permasalahan yang dapat dirumuskan agar pembahasan masalah dapat lebih terarah dan terperinci, dengan maksud untuk mempermudah identifikasi dan pemahaman terhadap aplikasi. Adapun batasan masalah dalam implementasi metode VSM ini adalah.

1. Bahasa yang dapat dibaca oleh sistem harus dengan bahasa Indonesia yang baik dan benar.
2. Data yang akan digunakan diperoleh dari Sekolah Menengah Atas (SMA) Negeri 13 Palembang. Data berupa kumpulan soal dan jawaban yang digunakan oleh guru di SMA Negeri 13 Palembang.
3. Studi kasus mata pelajaran Ekonomi kelas X (Sepuluh). Karena di dalam mata pelajaran tersebut mengandung banyak teori dibandingkan mata pelajaran lain.
4. Menggunakan algoritma Nazief dan Adriani dalam melakukan proses *stemming* dan *stopword*.
5. Menggunakan metode *Vector Space Model* (VSM) dalam mencocokan kata, sedangkan untuk metode pembobotan kata menggunakan *Term Frequency* (TF).
6. Menggunakan persentase nilai jawaban dalam memberikan rekomendasi nilai.
7. Menggunakan pemograman berorientasi objek.

8. Untuk memodelkan perangkat lunak menggunakan *Unified Modeling Language* (UML).
9. Sistem yang akan dibangun berbasis *website*.

1.4 Metodologi Penelitian

Metologi penelitian yang digunakan oleh penulis dalam menulis laporan tugas akhir ini adalah metodologi deskriptif, yaitu metode pembahasan masalah yang digunakan untuk menggambarkan objek untuk diteliti, dengan cara mencari, mengumpulkan, dan menganalisis data yang diperoleh.

1.4.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian adalah Studi Pustaka. Studi Pustaka yang dilakukan adalah dengan mempelajari berbagai literatur, seperti buku-buku, artikel-artikel, *e-book*, *website*, jurnal, dan sumber-sumber yang berkaitan dengan metode VSM yang akan dibangun, meliputi kecerdasan buatan, desain, *tools* dan juga pemodelan dengan UML yang dapat membantu menyelesaikan implementasi metode VSM ini.

1.4.2 Metode Pembangunan Perangkat Lunak

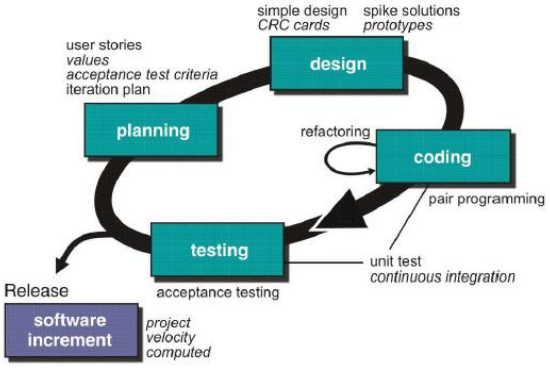
Metode yang digunakan untuk pembangunan perangkat lunak dalam penelitian ini menggunakan *Agile Model*. Model ini adalah model yang memberikan pendekatan-pendekatan sistematis dan berurutan bagi pengembang piranti lunak menurut Roger S. Pressman [5] adalah :

a. *Planning*
Tahap perancangan yang dilakukan adalah dengan pemodelan menggunakan metode pemrograman berorientasi objek dan menerapkan metode VSM pada sistem jawaban esai untuk pencocokkan kata dan rekomendasi nilai.

b. *Design*
Tahap *design* merupakan tahap perancangan dari pembangunan sistem jawaban esai yang akan dibuat untuk identifikasi dan mengatur *class – class* di konsep *object oriented.*

c. *Coding*
Setelah tahap perancangan sistem selanjutnya dilakukan konversi rancangan sistem ke dalam kode – kode bahasa pemograman yang digunakan yaitu *php.*

d. *Testing*
Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dibuat telah sesuai dengan desainnya dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan.



Gambar 1. Model *Agile* [5]

2. ISI PENELITIAN

2.1 *Vector Space Model* (VSM)

Vector space model (VSM) adalah representasi kumpulan dokumen sebagai vektor dalam sebuah ruang vektor. VSM merupakan teknik dasar dalam perolehan informasi yang dapat digunakan untuk penilaian relevansi dokumen terhadap kata kunci pencarian (*query*) pada mesin pencari, klasifikasi dokumen, dan pengelompokan dokumen [3]. Dalam *Vector Space Model*, koleksi dokumen direpresentasikan sebagai sebuah matrik *term-document* (matrik *term-frequency*). Setiap sel dalam matrik bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokmen yang ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir dalam dokumen [4].

D1 : Saya mahasiswa Ilmu Komputer
D2 : Saya menimba ilmu di Fakultas Ilmu Komputer
D3 : Mahasiswa Fakultas Ilmu Komputer banyak

	D1	D2	D3
Banyak	0	0	1
Di	0	1	0
Fakultas	0	1	1
Ilmu	1	2	1
Komputer	1	1	1
Mahasiswa	1	0	1
Menimba	0	1	0
Saya	1	1	0

Gambar 2. Contoh Dokumen Dan Matriks Kata-Dokumen

Melalui *vector space model* dan TF *weighting* maka akan didapatkan representasi nilai numerik dokummen sehingga kemudian dapat dihitung kedekatan antar dokumen. Semakin dekat dua vektor di dalam suatu VSM, maka semakin mirip dua dokumen yang diwakili vektor tersebut. Terdapat empat fungsi untuk mengukur kemiripan (*similarity measure*) yang dapat digunakan untuk model ini :

1. *Cosine distance* / *cosine similarity*
2. *Inner similarity*
3. *Dice similarity*
4. *Jaccard similarity*

Salah satu ukuran kemiripan teks yang popular adalah *cosine similarity*. Ukuran ini menghitung nilai cosinus sudut antara dua vektor. Jika terdapat dua vektor dokumen d dan *query* q, serta t *term* diekstrak dari koleksi dokumen maka nilai cosinus antara d dan q didefinisikan sebagai berikut :

$$sim(Di, Q) = \frac{\sum Di \cdot Q}{\sqrt{\sum D^2} \cdot \sqrt{\sum Q^2}} \quad (1)$$

2.2 *Term Frequency-Inverse Document Frequency* (TF-IDF) *Weighting*

Metode pembobotan yang paling sederhana terhadap suatu *term* (*term weighting*) adalah dengan menggunakan frekuensi kemunculan *term* (kata) / *term frequency* (TF) yang bersangkutan pada suatu dokumen. *Inverse Document Frequency* (IDF) adalah logaritma dari rasio jumlah keseluruhan dokumen yang diproses dengan jumlah dokumen yang memiliki *term* bersangkutan. Lalu Salton bereksperimen untuk mengkombinasikan kedua metode pembobotan tersebut, dengan mempertimbangkan frekuensi inter-dokumen dan frekuensi intra-dokumen dari suatu *term*. Dengan menggunakan frekuensi *term* pada suatu dokumen dan distribusinya pada keseluruhan dokumen, yakni kemunculan pada dokumen-dokumen lain (IDF). Salton menarik kesimpulan melalui eksperimennya bahwa *term-term* dengan total frekuensin menengah, lebih berguna dalam *retrieval* jika dibandingkan dengan *term-term* yang total frekuensinya terlalu tinggi atau terlalu rendah. Konsep intra-dokumen dan inter-dokumen ini kemudian dikenal sebagai metode TF-IDF.

Rumus yang digunakan untuk menyatakan bobot (*w*) masing-masing dokumen terhadap kata kunci adalah :

$$Wd,t = tf\ d,t \times \text{IDF}t \quad (2)$$

Dimana :

d = dokumen ke-d

t = kata ke-t dari kata kunci

Wd,t = bobot dokumen ke-d terhadap kata ke-t

2.3 Algoritma Stemming Nazief dan Adriani

Algoritma *stemming* Nazief dan Adriani (1996) dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuan menjadi awalan (*prefix*), sisipan (infix), akhiran (*suffix*) dan gabungan awalan-akhiran (*confixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.

Aturan *morfologi* Bahasa Indonesia mengelompokkan imbuhan ke dalam beberapa kategori sebagai berikut :

1) *Inflection suffixes* yakni kelompok akhiran yang tidak merubah bentuk kata dasar. Sebagai contoh, kata “duduk” yang diberikan akhiran “-lah” akan menjadi “duduklah”.
   Kelompok ini dibagi menjadi dua :
   a. *Particle* (P) atau partikel, yakni termasuk di dalamnya “-lah”, “-kah”, “-tah”, dan “-pun”.
   b. *Possessive Pronoun* (PP) atau kata ganti kepunyaan, termasuk di dalamnya adalah “-ku”, “-mu”, dan “-nya”.
2) *Derivation Suffixes* (DS) yakni kumpulan akhiran asli Bahasa Indonesia yang secara langsung ditambahkan pada kata dasar yaitu akhiran “-i”, “-kan”, dan “-an”.
3) *Derivation Prefixes* (DP) yakni kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Termasuk di dalamnya adalah :
   a. Awalan yang dapat ber*morfologi* (“me-”, ”be-”, “pe-”, dan “te-”)
   b. Awalan yang tidak ber*morfologi* (“di-”, “ke-”, dan “se-”)

Aturan untuk pemenggalan kata awalan pada algoritma *stemmer* Nazief dan Adiani dapat dilihat pada Tabel dibawah ini.

Tabel 1 Aturan Pemenggalan Awalan *Stemmer* Nazief Dan Adriani [7]

Aturan	Format Kata	Pemenggalan
1	berV…	ber-V…| be-rV…
2	berCAP…	ber-CAP… dimana C!=’r’ & P!=’er’
3	berCAerV…	ber-CaerV… dimana C!=’r’
4	belajar	bel-ajar

Aturan	Format Kata	Pemenggalan
5	beC1erC2…	be-C1erC2.. dimana C1!={'r'|'1'}
6	terV…	ter-V… | te-rV…
7	terCerV…	ter-CerV… diaman C!='r'
8	terCP…	ter-CP... dimana C!='r' dan P!='er'
9	teC1erC2...	te-C1erC2... dimana C1!='r'
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe{r|l}...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q}...	meng-{g|h|q}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV…
19	mempV...	mem-pV... dimana V!='e'
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
23	perCAP…	per-CAP... dimana C!='r' dan P!='er'
24	perCAerV...	per-CAerV... dimana C!='r'
25	pem{b|f|V}...	pem-{b|f|V}...
26	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
27	pen{c|d|j|z}...	pen-{c|d|j|z}...
28	penV...	pe-nV... | pe-tV...
29	peng{g|h|q}	peng-{g|h|q}...
30	pengV...	peng-V... | peng-kV...
31	penyV...	peny-sV…
32	pelV...	pe-lV... kecuali 'pelajar' yang menghasilkan 'ajar'
33	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
34	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!='e'

Keterangan simbol huruf :

C : huruf konsonan

V : huruf vokal

A : huruf vokal atau konsonan

P : partikel atau fragmen dari suatu kata, misalnya "er"

2.4 *Morphological Analysis*

Morphological Analysis adalah proses dimana setiap kata yang berdiri sendiri (individual word) dianalisis kembali ke komponen pembentuk dan token nonword seperti tanda baca dan sebagainya dipisahkan dari kata tersebut. Hasil akhir dari proses ini adalah proses Parsing. Parsing adalah proses mengkonversikan daftar kata yang berbentuk kalimat ke dalam bentuk struktur yang mendefinisikan unit yang diwakili oleh sebuah daftar [6]. Pada tabel dibawah ini dapat dilihat beberapa karakter (token nonword)yang harus dipisahkan dari kata.

Tabel 2 Karakter (Token *Nonwrod*)

Karakter			
!	~	+	/
@	&	+	\
#	*	{	"
$	(	}	'
%	)	[	:
^	-	]	:
`	_	|	.
,	<	>	?
White space (tab, spasi, enter)			

2.5 *Stopwords Removal*

Stopword removal adalah sebuah proses untuk menghilangkan kata yang 'tidak relevan' pada hasil parsing sebuah dokumen teks dengan cara membandingkannya dengan *stoplist*. *Stoplist* berisi sekumpulan kata yang 'tidak relevan', namun sering sekali muncul dalam sebuah dokumen. Pada tabel dibawah ini merupakan daftar *stoplist* yang digunakan didalam sistem.

Tabel 3 Daftar *Stoplist* [7]

Stoplist				
'yang'	'untuk'	'ini'	'telah'	'begitu'
'pada'	'ke'	'karena'	'dari'	'maka'
'menurut'	'namun'	'kepada'	'di'	'lagi'
'antara'	'dia'	'oleh'	'serta'	'tentang'
'ia'	'dua'	'saat'	'bagi'	'demi'
'seperti'	'tidak'	'harus'	'sekitar'	'dimana'
'jika'	'dan'	'sementara'	'kami'	'kemana'
'sehingga'	'kembali'	'setelah'	'belum'	'sampai'
'sebagai'	'ada'	'mereka'	'anda'	'sedangkan'
'masih'	'juga'	'sudah'	'itulah'	'selagi'
'hal'	'akan'	'saya'	'daripada'	'sementara'
'ketika'	'denga	'terhada	'yakni'	'sebelum

Stoplist				
	n’	p’		’
‘adalah’	‘kita’	‘secara’	‘yaitu’	‘tetapi’
‘itu’	‘hanya’	‘agar’	‘kenapa’	‘apakah’
‘dalam’	‘atau’	‘lain’	‘mengapa’	‘supaya’
‘bisa’	‘bahwa’	‘anda’	‘begitu’	‘dll’

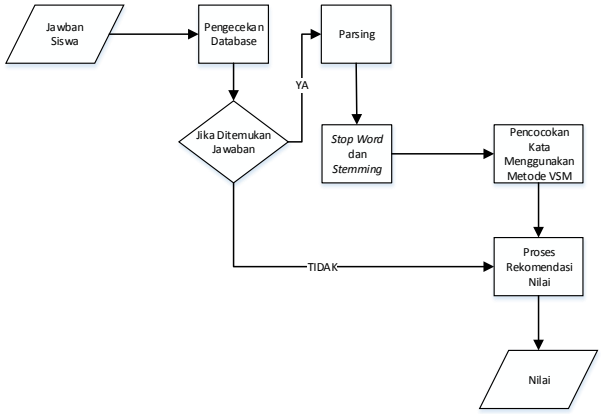
2.6 *Stemming & Lemmatization*

Stemming merupakan sebuah proses yang bertujuan untuk mereduksi jumlah variasi dalam representasi dari sebuah kata. Resiko dari proses stemming adalah hilangnya informasi dari kata yang di-stem. Hal ini menghasilkan menurunnya akurasi atau presisi. Sedangkan, keuntungannya adalah, proses stemming dapat meningkatkan kemampuan untuk melakukan recall.

Tujuan dari stemming sebearnya adalah meningkatkan performance dan mengurangi penggunaan resource dari system dengan mengurangi jumlah unique word yang harus diakomodasikan oleh sistem. Jadi, secara umum algoritma stemming mengerjakan transformasi dari sebuah kata menjadi sebuah standar representasi morfologi (yang dikenal sebagai stem).

Lemmatization adalah sebuah proses untuk menemukan bentuk dasar dari sebuah kata. Ada sebuah teori yang menjelaskan bahwa lemmatization adalah proses yang bertujuan untuk melakukan normalisasi pada teks atau kata berdasarkan bentuk dasar yang merupakan bentuk lemma-nya. Normalisasi disini dalam artian mendefinisikan dan menghapus sebuah prefix serta suffiks dari sebuah kata. Lemma adalah bentuk dasar dari sebuah kata yang memiliki arti tertentu berdasarkan pada kamus.

2.7 Proses Utama Sistem



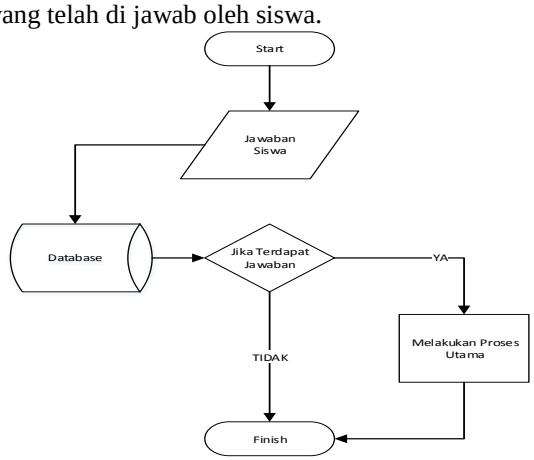
Gambar 3 Alur Proses Utama Sistem

Penjelasan dari gambar 2.2 adalah sebagai berikut :

1. Pengecekan *Database*
   Merupakan langkah dimana sistem melakukan pengecekan kedalam database, pertanyaan mana saja yang telah di jawab oleh siswa.
2. *Parsing*
   Merupakan proses mencari kata-kata unik dari jawaban yang telah dimasukkan oleh siswa.
3. *Stopword* dan *Stemming*
   Merupakan proses pencarian kata-kata sambung, seperti : dengan, atau, yang dsb, dan mengembalikan kata ke dalam kata dasarnya.
4. Pencocokan kata menggunakan metode VSM
   Merupakan proses pencocokan kata masukan dari siswa dan kunci jawaban yang terdapat di dalam *database*.
5. Rekomendasi Nilai
   Merupakan proses memberikan rekomendasi nilai sesuai dengan kecocokan antara jawaban siswa dengan kunci jawaban yang terdapat di dalam *database*.

2.7.1 Pengecekan *Database*

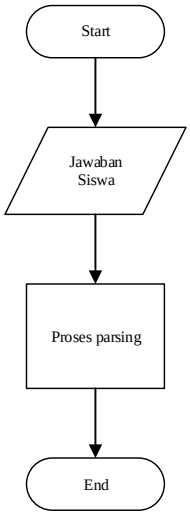
Merupakan langkah dimana sistem melakukan pengecekan kedalam database, pertanyaan mana saja yang telah di jawab oleh siswa.



Gambar 4 *Flowchart* Proses Pengecekan *Database*

2.7.2 *Parsing*

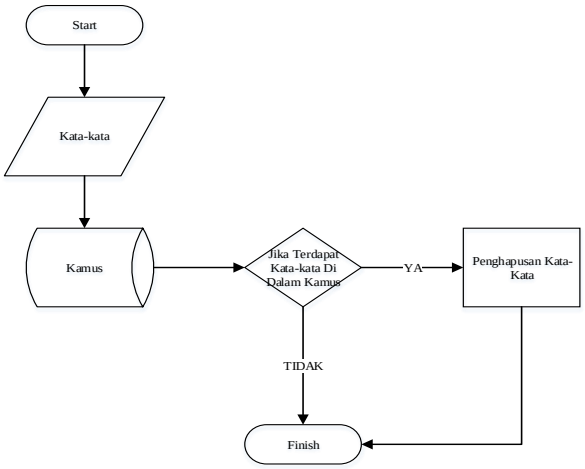
Merupakan proses mencari kata-kata unik dari jawaban yang telah dimasukkan oleh siswa.



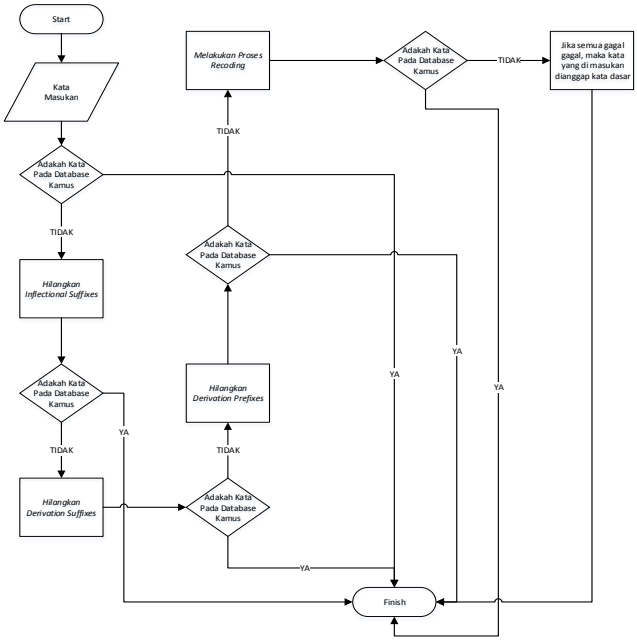
Gambar 5 *Flowchart* Proses *Parsing* Jawaban Siswa

2.7.3 Stopword dan *Stemming*

Merupakan proses pencarian kata-kata sambung, seperti : dengan, atau, yang dsb, dan mengembalikan kata ke dalam kata dasarnya.



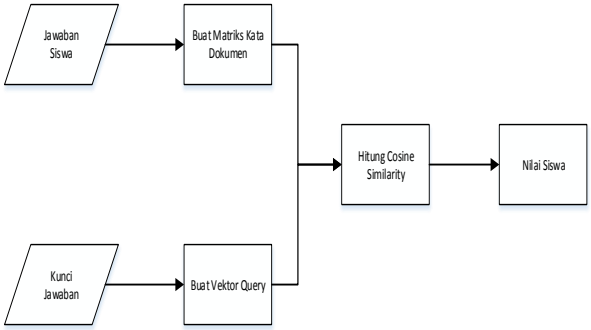
Gambar 6 *Flowchart* Penghapusan Kata-Kata (*Stopword*)



Gambar 7 *Flowchart* Algoritma Nazief Dan Adriani [7]

2.7.4 Pencocokan Kata

Metode yang digunakan dalam mencocokkan kata adalah metode *Vector Space Model* (VSM). Alur dari metode VSM dapat dilihat pada gambar dibawah ini.



Gambar 8 Alur Proses Utama VSM

Untuk menghitung jumlah kata yang cocok digunakan *cosine similarity*. Rumus untuk menghitungnya adalah sebagai berikut :

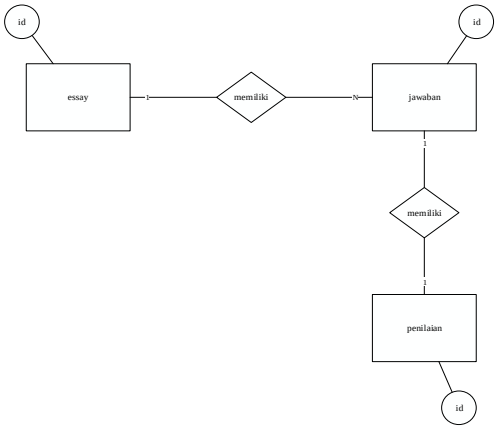
$$sim\,(Di, Q) = \frac{\sum Di * Q}{\sqrt{\sum D\text{^}2} * \sqrt{\sum Q\text{^}2}}$$

2.7.5 Rekomendasi Nilai

Merupakan proses memberikan rekomendasi nilai sesuai dengan kecocokan antara jawaban siswa dengan kunci jawaban yang terdapat di dalam *database*. Cara menghitungnya adalah sebagai berikut :

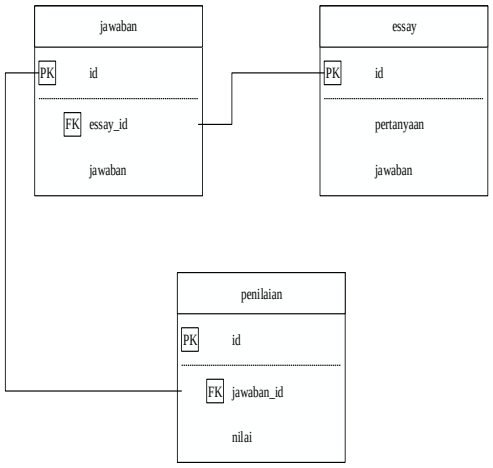
$$total\ nilai = \frac{Jumlah\ Nilai\ Benar}{Jumlah\ soal}$$

2.8 ERD



Gambar 9 ERD

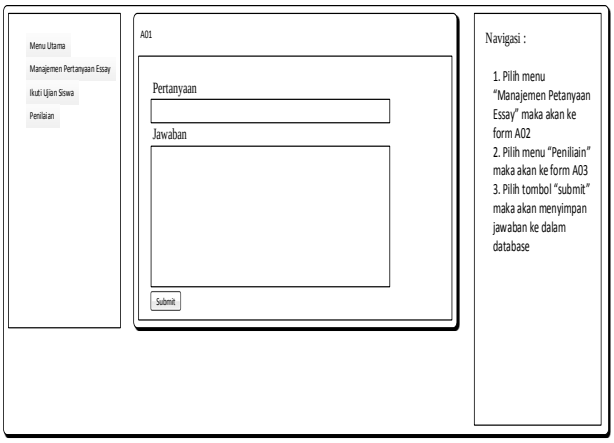
2.9 Skema Relasi



Gambar 10 Skema Relasi

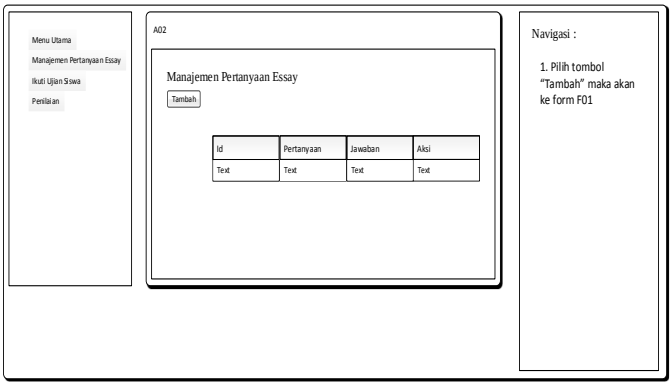
2.10 Perancangan Antarmuka

1. Perancangan Antarmuka Tampilan Utama



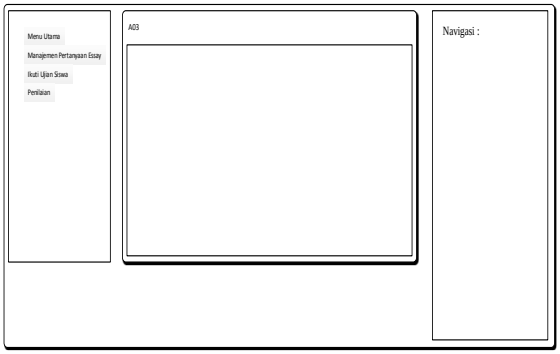
Gambar 11 Perancangan Antarmuka Tampilan Utama

2. Perancangan Antarmuka Tampilan Manajemen Pertanyaan Essay



Gambar 12 Perancangan Antarmuka Tampilan Manajemen Pertanyaan Essay

3. Perancangan Antarmuka Tampilan Penilaian



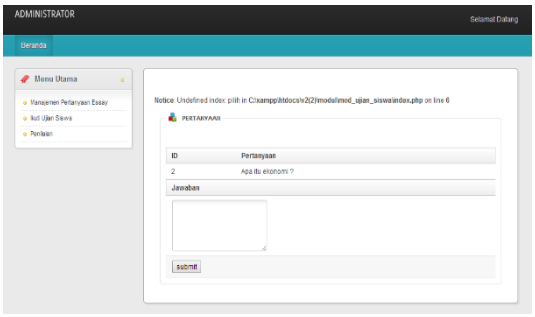
Gambar 13 Perancangan Antarmuka Tampilan Penilaian

3. IMPLEMENTASI DAN HASIL PENGUJIAN

3.1 Implementasi Antarmuka

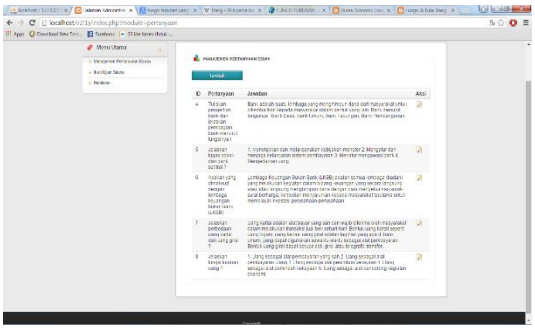
Dari perancangan antarmuka yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan. Implementasi antarmuka sistem meliputi :

1. Implementasi Antarmuka Tampilan Utama



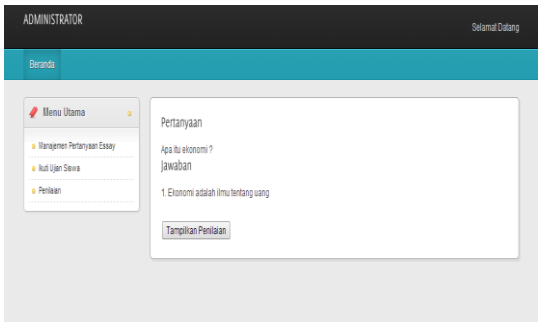
Gambar 14 Implementasi Antarmuka Tampilan Utama

2. Implementasi Antarmuka Tampilan Manajemen Pertanyaan Essay



Gambar 15 Implementasu Antarmuka Tampilan Manajemen Pertanyaan Essay

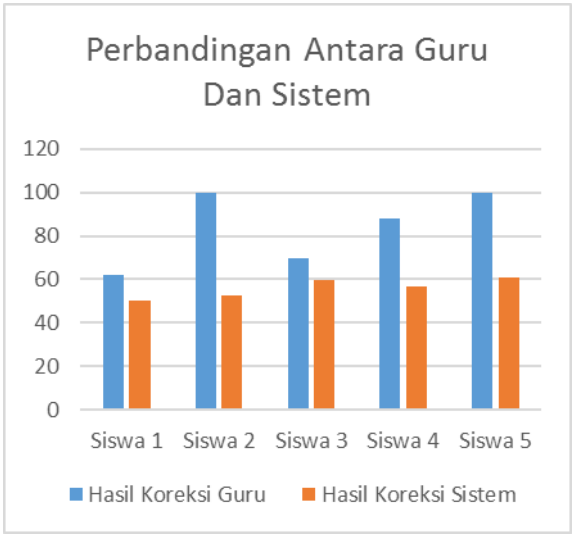
3. Implementasi Antarmuka Tampilan Penilaian



Gambar 16 Implementasi Antarmuka Tampilan Penilaian

3.2. Hasil Pengujian

Pengujian akurasi dimulai dengan pengkoreksian secara manual, yaitu guru langsung mengkoreksi jawaban yang telah di jawab oleh siswa. Kemudian untuk tahap selanjutnya menggunakan metode VSM untuk pencocokan kata dan sistem dalam memberikan rekomendasi nilai. Setelah dilakukan kedua proses tersebut akan didapatkan perbandingan hasil keakuratan antara koreksi yang dilakukan oleh guru dan yang dilakukan oleh sistem. Dalam hal ini diambil data contoh jawaban dari lima orang siswa. Didapatkanlah Hasil yang dapat dilihat pada gambar dibawah ini :



Gambar 17 Perbandingan Hasil Koreksi Guru Dan Sistem

4. PENUTUP

4.1. Kesimpulan

Berdasarkan hasil pengujian dapat ditarik kesimpulan sebagai berikut :

1. Metode VSM dapat mencocokkan kata antara kunci jawaban dan jawaban yang telah dimasukkan oleh siswa.
2. Didapatkan rata-rata nilai yang di rekomendasikan oleh sistem adalah 56,07% dan rata-rata nilai yang direkomendasikan oleh guru adalah 84%, dan perbedaan antara nilai yang diberikan oleh guru dan sistem adalah 27,93%.
3. Waktu yang dibutuhkan oleh sistem dalam mencocokkan kata dan memberikan rekomendasi nilai sangat lama, dikarenakan semakin banyak siswa yang memasukkan jawabannya, semakin banyak pula waktu yang dibutuhkan oleh sistem dalam mencocokkan kata dan memberikan rekomendasi nilai. Rata-rata waktu yang dibutuhkan sistem dalam mencocokkan kata dan memberikan rekomendasi nilai untuk contoh diatas adalah 17 detik.

4.1 Saran

Berikut adalah saran yang dapat dilakukan untuk pengembangan dari penelitian yang telah dilakukan :

1. Untuk meningkatkan keakurasian sistem dalam memberikan rekomendasi nilai lebih baik menggunakan metode *Natural Language Processing* (NLP) karena NLP menilai tidak hanya menilai berdasarkan kesamaan kata saja, tapi berdasarkan susunan kata (*grammar*) dari jawaban yang telah dimasukkan oleh siswa.
2. Untuk penelitian lebih lanjut disarankan menggunakan penggabungan metode yang

ada dengan beberapa metode yang lain untuk mendapatkan hasil yang lebih baik.

DAFTAR PUSTAKA

[1] S. Hamza, M. Sarosa and P. B. Santoso, "Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rapid Karp," *Jurnal EECCIS*, vol. 7, 2013.

[2] S. Astutik, A. D. Cahyani and M. K. Sophan, "Sistem Penilaian Otomatis Dengan Menggunakan Algoritma Winnowing," *Jurnal Informatika*, vol. 12, pp. 47 - 52, 2014.

[3] H. Septiantri, "Perbandingan Metode Latent Semantic Analysis Dan Vector Space Model Untuk Sistem Penilaian Jawaban Esai Otomatis Bahasa Indonesia," 2009.

[4] Darmawan, Heru Adi; Wurijanto, Tutut; Masturi, Akh;, "Rancang Bangun Aplikasi Search Engine Tafsir Al-Qur'an Menggunakan Teknik Text Mining Dengan Algoritma VSM (Vector Space Model)".

[5] R. S. Pressman and B. R. Maxim, Software Engineering, A Practitioner's Approach Eighth Edition, New York: McGraw-Hill Education, 2015.

[6] W. Budiharto and D. Suhartono, Artificial Intelligence : Konsep dan Penerapannya, Jakarta: Andi, 2014.

[7] Tahitoe, Andita Dwiyoga, "Implementasi Modifikasi Enchanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming," *Jurnal Informatika*, 2010.

[8] S. Dikli, "An Overview Of Automated Scoring Of Essay," *The Journal of Technology, Learning,and Assessment*, Vols. 5, number 1, 2006.

[9] R. A. S. and M. S. , Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek, Bandung: Informatika, 2013.

[10] Fathansyah, Basis Data : Edisi Revisi, Bandung: Informatika, 2012.